# Vom Consumer zum Maker

Handbuch über Single-Board Platinen
für Lehrkräfte, SchülerInnen und Interessierte

CC BY NC SA

► Lisa Kuka

# Acknowledgment

## 2019

Die Informationen in diesem Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht ausgeschlossen werden. Die Autorin kann für fehlerhafte Angaben und deren Folgen werder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler ist die Autorin dankbar.
Logos von den Hardware und Software Herstellern sind von den jeweiligen Websites entnommen.

folgende Bilder wurden von **https://www.elegoo.com/download/** entnommen:
Bauteile: IR Sensor, Temperatur und Feuchtigkeitssensor, Stepper Motor, DC Motor, Druckknöpfe, Water Level Sensor, Piezo Lautsprecher

folgende Bilder wurden von der **deutschen und englischen Wikipedia** entnommen:
Bauteile: Lichtsensoren, LEDs, Transistoren, Kondensatoren, Widerstände, Lichtspektrum

folgende Bilder wurden von **https://www.adafruit.com** entnommen:
Bauteile: flora, gemma
Projekt: Beweg-Dich-Armband, Ampli-Tie

folgende Bilder wurden von **https://www.instructables.com** entnommen:
Bauteile: flora, gemma,
Projekt: Projekt Sonic Scarf, Weather Station

Schaltsymbole und Vektorgrafiken von Schaltmodellen wurden aus der Software
**fritzing** und **circuito.io** entnommen

Grafiken wurden von **freepik.com** entnommen
Grafiken der Schulfächer basieren auf: School vector, ibrandify und macrovector created by freepik

folgende Platinen und Bauteile wurden von **Sparkfun https://www.sparkfun.com** entnommen (sie stehen unter CC Lizenz):
Arduino, LilyPad, Raspberry Pi 3, Makey Makey, LilyPad USB Breakout

Bilder und Grafiken von **Lisa Kuka**:
Multimeter, Grafik Ohm'sches Gesetz und Wassertanks (angelehnt an Sparkfun Grafiken), Bilder zu Blackout Poetry, "Stein, Schere, Papier" und DIY-Surgery, Screenshots Arduino IDE
Gaming Console: Bilder der Controller

Bilder von **Bianca Lindorfer**:
Gaming Console: Work-in-Progress Bilder

Titelseite: Verwendet Grafik-Elemente pikisuperstar / Freepik &
Photo by cottonbro from Pexels

# Inhaltsverzeichnis


1

**Einleitung ..... 1**
Projektübersicht ..... 2
Creative Commons ..... 3
Single-Board Platinen ..... 4
Software ..... 6
Making ..... 8

2

**Elektronik 101 ..... 9**
Elektronische Bauteile ..... 12

3

**Programmier-Basics ..... 14**

4

**Arduino ..... 17**
Arduino IDE ..... 18
Arduino Beispiele ..... 20
Schere, Stein, Papier ..... 26

5

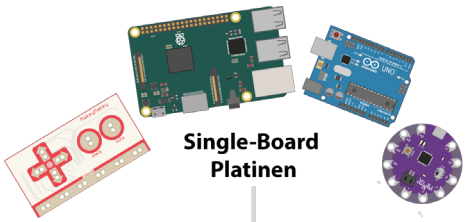
**LilyPad ..... 29**
Tipps für Wearables ..... 30
eFashion ..... 30
Ampli-Tie ..... 31
Sonic Sound Scarf ..... 32
Beweg-Dich-Armband ..... 34

6

**Makey Makey ..... 35**
Do-It-Yourself-Surgery ..... 36
Blackout Poetry ..... 37
eInstruments ..... 38
Sprechende Gemälde ..... 39
Scratch ..... 40

7

**Raspberry Pi ..... 41**
Raspberry Pi Projekte ..... 43
Alexify My Pi ..... 44
Game Station ..... 46



Single-Board Platinen

Handbuch

01 12-Lehr-Lern-Prinzipien

02 COOL-Konzept

COoperative Open Learning
COmputer-supported Open Learning
COOL

03 Critical Creativity

Sage es mir,
und ich werde es vergessen.
Zeige es mir,
und ich werde es vielleicht behalten.
Lass es mich tun,
und ich werde es können.

Hilf mir, es **selbst zu tun.**
Der Lehrer muss passiv werden,
damit das Kind **aktiv werden** kann.

Wir müssen uns daran erinnern,
dass Intelligenz nicht ausreicht.
**Intelligenz plus Charakter** -
das ist das Ziel wahrer Bildung.
Die vollständige Ausbildung
gibt nicht nur Konzentrationsfähigkeit,
sondern auch **würdige Ziele,**
**auf die man sich konzentrieren kann.**

Man kann einen
Menschen **nichts lehren;**
man kann ihm nur helfen,
es in sich **selbst zu finden.**

Der Sinn und Zweck,
einem Kind etwas beizubringen, ist,
es dazu zu befähigen,
**ohne seinen Lehrer auszukommen.**

Der **Analphabet** der Zukunft
wird nicht die Person sein,
die nicht lesen kann.
Es wird die Person sein,
**die nicht lernen kann, zu lernen.**

KONFUZIUS
(551 - 479 V.U.Z.)

GALILEO GALILEI
(1564 - 1642)

MARIA MONTESSORI
(1870 - 1952)

ELBERT HUBBARD
(1856 - 1915)

MARTIN LUTHER KING JR
(1929 - 1968)

ALVIN TOFFLER
(1928 - 2016)

# Einleitung

Als Lehrkraft stellt man sich zwangsläufig die Frage, wieviel von dem was man den Schülern und Schülerinnen mit auf den Weg gibt, tatsächlich hängen bleibt – also was in ein paar Jahren noch vorhanden sein wird. Das sogenannte Binge Learning – also das sture auswendig Lernen von Prüfungsstoff um ihn dann bei der Prüfung wieder “auszukotzen” und anschließendem Vergessen – ist wohl für alle Beteiligten, reine Zeitverschwendung. Zeit, die Schüler und Schülerinnen sowie Lehrkräfte sinnvoller nutzen hätten können, wie auch die Lehrkräfte. Nichts ist frustrierender als der Gedanke unnötig Zeit verbracht zu haben.

Dieses Handbuch versucht, Ideen rund um Single-Board Platinen zu geben um einen fächerübergreifenden Unterricht zu ermöglichen. Der Ruf nach digitalen Kompetenzen wird immer lauter, die Frage ist jedoch, wie diese am besten vermittelt werden können.
Die Informatik ist seit jeher eine interdisziplinäre Disziplin gewesen, sie bietet viele Überschneidungen mit anderen Fächern. Ihr volles Potential kann also erst in fächerübergreifender Projektarbeit entfaltet werden. Hierbei ist zu beachten, dass das Rad nicht neu erfunden werden muss. Unzählige Communitys bieten online bereits eine Vielzahl an Projektideen, sie müssen von Lehrkräften lediglich aufgegriffen werden und mit didaktischen Überlegungen verknüpft werden. Das bietet auch den Vorteil, dass Schüler und Schülerinnen einen kritischen, produktiven Umgang mit dem Internet lernen. Das Internet bietet also nicht nur lustige Katzenvideos und verleitet zu kriminellen Aktivitäten, nein – ganz im Gegenteil – es bietet Tutorial und eine immense Menge an Informationen, die nur darauf warten, entdeckt und weiterentwickelt zu werden.

Das Handbuch baut auf Erkenntnissen der Neurodidaktik auf, die wichtigsten sind in den 12-Lehr-Lern-Prinzipien von Caine zusammengefasst. Diesen Prinzipien hat sich auch das COOL-Konzept verschrieben. Unterricht sollte kooperativ, offen und computer-unterstützt sein, dadurch wird er einfach COOL. Das Schüler und Schülerinnen Freude am Lernen haben sollten, geht nicht auf eine Kuschel- oder Hippiepädagogik zurück, so wie von Kritiker*innen gerne behauptet, sondern auf einen Erfahrungsschatz von Pädagog*innen, der heute durch die Neurodidaktik untermauert werden kann. Freude und positive Emotionen, aber auch sinnhafte Praxiserfahrungen fördern das Lernen. Außerdem ist unser Gehirn sozial. Diese Erkenntnisse schreien förmlich nach offenen interdisziplinären Projekten.

Weiters stellt das Handbuch keine genaue Schritt-für-Schritt Anleitung dar. Dem Forschungsgeist soll nicht im Weg gestanden werden. Die Projekte sind grob beschrieben und sollen inspirieren. Experimentieren und auch Fehler zu machen  ist erwünscht.

Viel Spaß

Dipl.-Ing. Lisa Kuka

► 12-Lehr-Lern-Prinzipien nach Arnold und Caine

# Projektübersicht

| Projekt | involvierte Fächer | Plattform | Schwierigkeitsgrad |
|---|---|---|---|
| Übungsbeispiele | Physik, Geografie | Arduino | ★ |
| Schere, Stein, Papier | Biologie | Arduino | ★ ★ ★ ★ |
| Weather Station | Geografie | Arduino | ★ |
| Ampli-Tie | Physik, Texitles Werken | LilyPad | ★ ★ ★ |
| Sonic Sound Scarf | Textiles Werken, Musikerziehung | LilyPad | ★ ★ ★ ★ |
| Beweg-Dich-Armband | Informatik (Ergonomie), Biologie | LilyPad | ★ ★ |
| DIY-Surgery | Biologie, Technisches Werken | Makey Makey | ★ ★ |
| Blackout Poetry | Deutsch, Fremdsprachen | Makey Makey | ★ |
| eInstruments | Musikerziehung, Technisches Werken | Makey Makey | ★ |
| Sprechende Gemälde | Kunst, Bildnerische Erziehung, Geschichte, Biologie, Geografie, ... | Makey Makey | ★ |
| Alexify My Pi | Informatik | Raspberry Pi | ★ ★ ★ ★ |
| Game Konsole | Technisches Werken | Raspberry Pi, Makey Makey | ★ ★ ★ ★ ★ |

Bildnerische Erziehung

Biologie

Deutsch

Fremdsprachen

Geschichte

Geografie

Informatik

Musik

Physik

Textiles Werken

Technisches Werken

# Creative Commons

Das Internet macht es einfach, kreative Werke von anderen zu teilen und darauf aufzubauen – allerdings stellen Gesetze oftmals eine Hürde dar. Immer wenn man ein Foto macht, ein Lied aufnimmt oder einen Artikel veröffentlicht, dann steht dieses Werk laut Urheberrechtsschutz automatisch unter "All right reserved" . Das bedeutet, dass andere es nicht wiederverwenden oder abwandeln (remixen) dürfen ohne vorher eine Erlaubnis einzuholen. Was ist aber, wenn man als Urheber*in möchte, dass es benutzt wird?

## Von "All rights reserved" zu "Some rights reserved"

Im digitalen Zeitalter sollte jeder und jede über Creative-Commons-Lizenzen Bescheid wissen. Creative Commons hilft, legal Wissen und Kreativität zu teilen, um eine Welt zu schaffen, in der besserer Zugang zu Informationen und Innovationen gewährleistet wird. Creative-Commons-Lizenzen wollen das volle Potential des Internets ausnützen, um im Informationszeitalter neue Entwicklungen und Produktivität zu steigern.

Eine CC-Lizenz erleichtern es, selber zu entscheiden, welche Rechte man behalten möchte und was man anderen erlaubt – und das ganz einfach und unkompliziert. Diese Lizenzen ermöglichen es, freie, einfach zu benützende und urheberrechtlich geschützte Lizenzen zu erzeugen, um der Öffentlichkeit eine Möglichkeit zu bieten, kreative Arbeiten zu teilen und zu benutzen. Werke, die mit einer Creative-Commons-Lizenz ausgestattet sind, dürfen also vervielfältigt, verbreitet, verändert und in neue Werke integriert werden.

Es gibt eine Vielzahl an unterschiedlichen Creative-Commons-Lizenzen. Sie teilen jedoch alle gewisse wichtige Eigenschaften. Beispielsweise bieten sie die Möglichkeit, das eigene Werk urheberrechtlich zu schützen und trotzdem anderen die Möglichkeit zu geben, dieses zu benutzen, weiterzuentwickeln oder anderweitig zu nutzen - für nicht-kommerzielle Zwecke. Die Lizenz stellt außerdem sicher, dass dem Lizenzgebenden die gebührende Anerkennung zuteil wird. Weiters sind sie weltweit einsetzbar und erlöschen mit dem gesetzlichen Urheberrecht. Jede CC-Lizenz kann zusätzlich mit Erlaubnissen ausgestattet werden, die die genauen Nutzungsbedingungen festlegen.

Das eigene Copyright wird dadurch also von "All rights reserved" zu "Some rights reserved abgewandelt.

**BY**

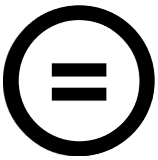
**Namensnennung**

**Autor*In** muss **immer genannt** werden

**NC**

**Nicht kommerziell**

Kann verwendet werden, aber **ohne Geld damit zu verdienen**



**ND**

**Keine Bearbeitung**

**Keine Veränderungen** erlaubt

**SA**

**Weitergabe unter gleichen Bedingungen**

Darf nur verwendet werden, wenn das neue Werk **unter derselben Lizenz** geführt wird

Fragen zur Lizenzwahl laut CC Website:

- Soll es anderen möglich sein, meine Werke kommerziell zu nutzen oder nicht?
- Soll es anderen möglich sein, meine Werke abzuwandeln und zu verändern?
- Weitergabe unter denselben Bedingungen: wenn es erlaubt ist, mein Werk abzuwandeln, kann auch darüber entschieden werden, ob das neue Werk unter derselben CC-Lizenz zur Verfügung gestellt werden soll.

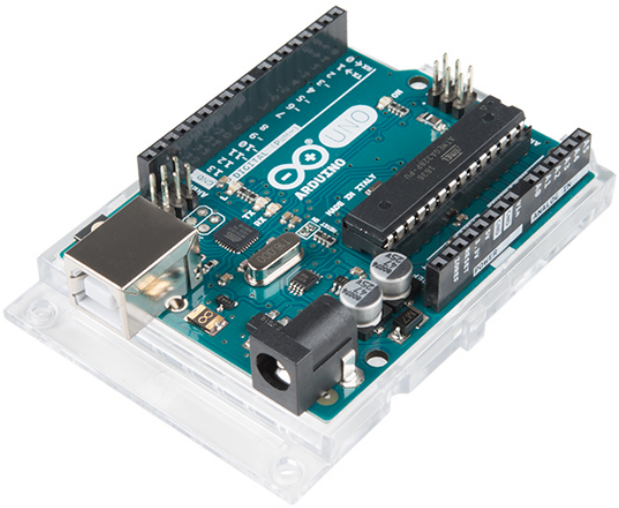
Mehr Informationen unter https://creativecommons.org/

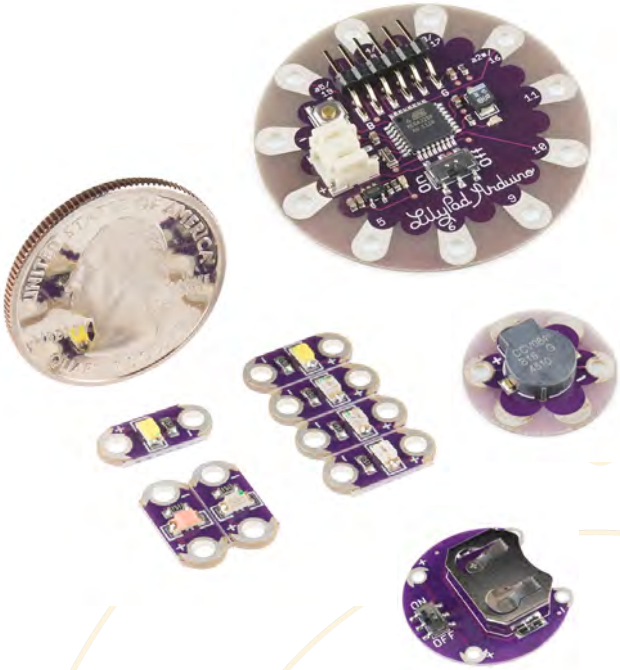
# Single-Board Platinen

## Arduino

- https://www.arduino.cc
- https://www.arduino-tutorial.de

Der Arduino ist ein open source Mikrocontroller Board. Er stellt digitale und analoge Ein- und Ausgänge zur Verfügung, über die sich Sensoren und Motoren ansteuern lassen. Für die Programmierung gibt es eine eigene Arduino IDE, über die sich per Knopfdruck der Sketch (das Programm) auf den Arduino hochladen lässt. Es gibt eine große Community und eine Vielzahl an Startersets und Literatur.

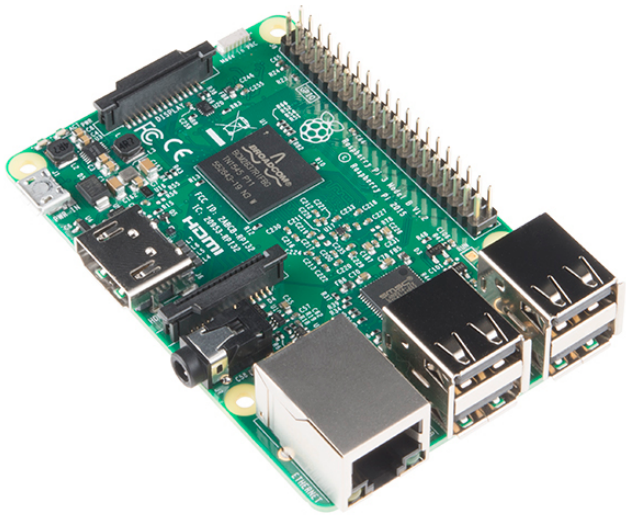




## LilyPad

Das LilyPad ist im Grunde genommen nichts anderes als ein Arduino, nur in einer anderen Form. Es wurde konzipiert, um auf Kleidung und Stoffen aufgenäht zu werden. Die analogen und digitalen Pins sind im Kreis angeordnet und lassen sich mit leitfähigen Garn verbinden. Außerdem gibt es eine Vielzahl an elektronischen Teilen, die oftmals schon Vorwiderstände, etc integriert haben und sich somit leichter vernähen lassen. Das Thema Wearables wird hier groß geschrieben.

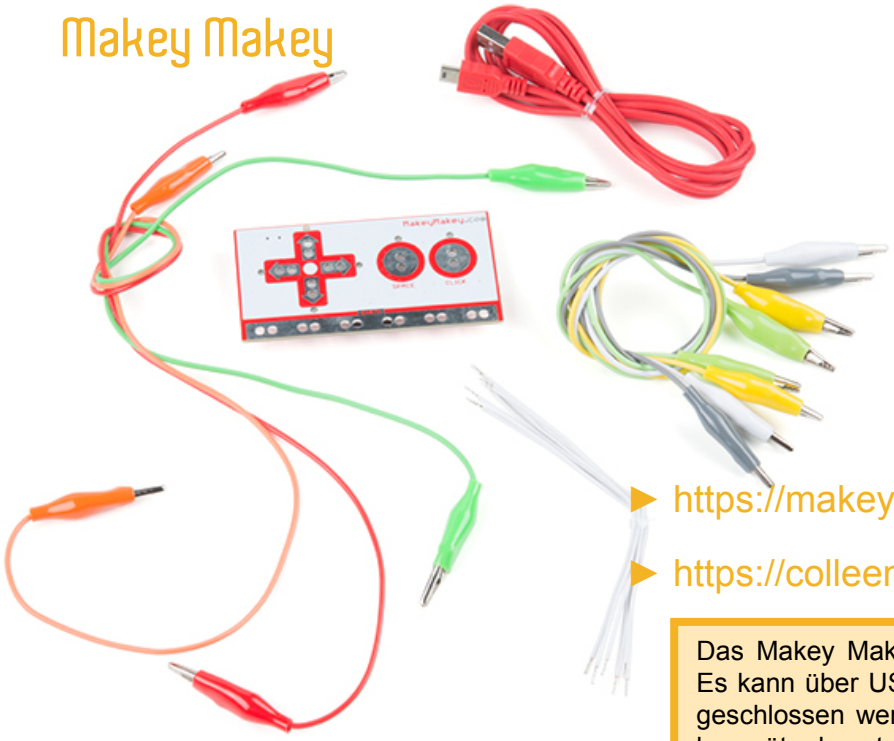
## Raspberry Pi



Der Raspberry Pi ist ein Einplatinencomputer. Er bietet nicht nur einige Pins über die – wie beim Arduino – Sensoren und Motoren angesteuert werden können, sondern auch SD-Karten Slot, USB-Ports und einen HDMI-Port. Er benötigt außerdem ein Betriebssystem. Somit ist der "Raspi" ein vollwertiger Computer. Auch er bietet eine große Community, viele verschiedene Startersets und Literatur.

- https:// www.raspberrypi.org
- https:// tutorials-raspberrypi.de

## Makey Makey



- https://makeymakey.com
- https://colleengraves.org/tag/makeymakey/

Das Makey Makey ist eine Leiterplatine. Es kann über USB an den Computer angeschlossen werden und wird als Eingabegerät erkannt. So können leitfähige Gegenstände – wie beispielsweise Bananen – zu einem Controller umgebaut werden. Die Handhabung ist sehr einfach und so können selbst Kinder damit problemlos umgehen.

- https://www.instructables.com
- https://learn.sparkfun.com/tutorials
- https://www.adafruit.com

# Software

zum Programmieren

## Scratch

Scratch ist eine einfache, grafische Programmierungsumgebung, die vom MIT entwickelt wurde, um Kindern das Programmieren näher zu bringen. Es können sehr einfach Spiele entwickelt werden, indem Blöcke (die für einfache Befehle stehen) per drag&drop miteinander verbunden und verschachtelt werden. Auch das Aufnehmen von Sprache ist ein Leichtes und eignet sich dadurch in Kombination mit dem Makey Makey dafür Sounds abzuspielen.
Das Betriebssystem Raspbian (Raspberry Pi) liefert eine etwas abgespeckte Version von Scratch mit.

► https://scratch.mit.edu/download

## Arduino IDE

Um den Arduino zu programmieren, ist eine Entwicklungsumgebung (Integrated Development Environment) nötig. Sie ist gratis und ermöglicht den leichten Einstieg in die Programmierung. Man lernt auch, dass man das Rad nicht immer neu erfinden muss und dass man bereits fertigen Code einbetten darf, um in zu verwenden, aber auch weiterzuentwickeln. Eine Vielzahl an Bibliotheken ermöglicht eine einfache Programmierung von Motoren, etc.

► https://www.arduino.cc/en/Main/Software

## MIT App Inventor

Der MIT App Inventor ist eine browserbasierte APP Entwicklungsumgebung, mit der es möglich ist, einfache Android Apps zu erstellen. Vor allem, wenn man seinen Arduino mit einem Bluetooth Modul vom Smartphone aus steuern möchte, muss man dazu in der Regel eine App programmieren. Der MIT App Inventor baut dabei – wie Scratch – auf einer modularen Programmierung mit eigenen Blöcken auf, die einfach zusammengestöpselt werden können.

► http://ai2.appinventor.mit.edu

## zum Erstellen von Schaltunplänen

### Fritzing

Fritzing ist eine open source Software, die das Erstellen von Schaltplänen erlaubt. Es steht eine große Bibliothek mit Bauteilen zur Verfügung. Dabei gibt es nicht nur Standardbauteile wie LEDs und Widerstände, sondern auch Platinen und Spezialbauteile von bekannten Herstellern wie Arduino und Sparkfun. Die bildhaften Darstellungen werden von der Software auch in Schaltpläne mit genormten Schaltzeichen umgewandelt. Mit Fritzing lassen sich die eigenen Projekte leicht dokumentieren und auch für andere zugänglich machen.

► http://fritzing.org/ download

### Circuito.io

Circuito ist eine browserbasierte Software, dies es – wie Fritzing – ermöglicht Schaltpläne zu erstellen. Allerdings stellt sie zusätzlich Beispielcode zur Verfügung und erstellt eine Shopping Liste. Der Clou bei Circuito besteht außerdem darin, dass die Software mitdenkt, das heißt füge ich beispielsweise eine LED hinzu, findet die Software eigenständig den passenden Widerstand und nimmt die Verbindungen selbstständig vor. Das kann für Anfängerinnen und Anfänger hilfreich sein.

► https://www.circuito.io/

App – kurz für Applikation und beschreibt im Allgemeinen ein Programm auf einem Tablet oder Smartphone

browserbasiert – es ist keine Installation auf dem Computer nötig

Entwicklungsumgebung oder IDE – (Integrated Development Environment,) Software mit der man programmieren kann

open source – Software oder Hardware, die frei zugänglich ist. Das bedeutet der Code oder Schaltplan ist frei einsehbar und das Weiterverarbeiten ist nicht nur erlaubt, sondern erwünscht

# Making

Seit vielen Jahren kann nun ein Aufblühen der Maker Szene beobachtet werden, vor allem angetrieben durch technologische und gesellschaftliche Entwicklungen wie das Web 2.0, erschwingliche Mikrocontroller und Einplatinencomputer, die Open Source Idee, sowie den Unbehagen passiver Konsument zu sein. Makers designen, programmieren und konstruieren. Dabei steht der Wissensaustausch und das gemeinsame Teilen von Ressourcen im Mittelpunkt. Diese Communitys haben durch das Internet Aufschwung erfahren. Das Tüfteln wird durch Zurschaustellung ergänzt. Nicht nur im Internet, auch auf Maker Faires und Ausstellungen kann sich Inspiration und Rat geholt werden (Katterfeldt, Zeising & Lund, 2013, 123f).

Auch die Lerntheorie des Konnektivismus, begründet von George Siemens, postuliert eine neue Sichtweise auf das Lernen. In Zukunft verliert das reine Auswendiglernen von Fakten und Handlungsweisen (das Was und Wie) an Bedeutung und das Auffinden von Wissen (Wo) nimmt zu. Gründe dafür liegen in der ständig wachsenden Informationsflut und dem immer schnelleren Verfall von Wissen durch technologische Entwicklungen. Wissen wird als Netzwerk mit Knoten und Bindungspunkten verstanden. Das bedeutet auch, dass die Wichtigkeit an informellem, vernetztem Wissen durch elektronisch gestütztes Lernen zunimmt. Dadurch entstehen "Communities of Practice", die sich gemeinsam Wissen erschließen und auch wieder weitergeben. Metakompetenzen, wie Unwichtiges von Wichtigen zu unterscheiden und vertrauensvolle Quellen zu überprüfen gewinnen an Bedeutung (Siemens, 2005, S. 3-10; iMoox, 2019, 10:35). Diese Metakompetenzen fallen in die schon erwähnten digitalen Grundkompetenzen, welche ver- mittelt werden sollen. Mit Hilfe von Making kann das auf eine sinnstiftende und kreative Weise erfolgen. Daraus kann geschlossen werden, dass das passive Konsumieren der Vergangenheit angehört und das aktive Machen auch im Schulwesen gefördert werden sollte.

**Making** ist daher ein **demokratisierendes und emanzipierendes Mittel**.

Seit dem Ende der 2000er Jahre begannen Forscher*innen und Pädagog*innen **FabLabs** für Unterrichtszwecke einzusetzen. Beispielsweise startete die renommierte Stanford Universität das FabLab@ School-Projekt bei dem es FabLabs in sogenannten K-12 Schulen auf der ganzen Welt initiierte. 2011 startete Maker Media ein MakerSpace Projekt mit einem DARPA (Defense Advanced Research Projects Agency, an US military research agency) Sponsoring. In den folgenden Jahren schlossen sich zahlreiche Museen, Schulen, Community Centers und Büchereien diesem Trend an (Blikstein, 2013, S. 206)

Auch in Österreich entstehen immer mehr Makerspaces. In größeren Städten jedes Bundeslandes wächst deren Anzahl stetig (Kneuer). Außerdem entstehen auch immer mehr akademische Makerspaces. 2019 eröffnete eine Hightech-Werkstatt der TU Graz und gilt somit als größter akademischer Makerspace in Österreich (Labor für Innovation, 2019). Hierin kann auch gesehen werden, dass die Wirtschaft ein Potential erkannt hat, und versucht sich, diese Räume zu eigen zu machen. Dies steht allerdings in Kontrast zur ursprünglichen Idee. Obwohl auch manche HTLs ihre Schule öffnen und so den Zugang zu den Werkzeugen und Maschinen erleichtern (Donnerbauer, 2019), gibt es immer noch zu wenig Initiativen um an Schulen kleine Makerspaces einzurichten. Die Ausstattung mit Single-Board Platinen, Lötkolben, elektronischen Bauteilen und Bastelmaterial würde dafür schon ausreichen.


Blikstein, P. (2013). Digital Fabrication and 'Making' in Education. In J. Walter-Herrmann & C. Büching (Hrsg.), *Fablab: Of machines, makers and inventors* (S. 203-222). Bielefeld: transcript Verlag.

Dittert, N. & Krannich, D. (2013). Digital Fabrication in Educational Contexts: Ideas for a constructionist workshop setting. In J. Walter-Herrmann & C. Büching (Hrsg.), *Fablab: Of machines, makers and inventors* (S. 173-180). Bielefeld: transcript Verlag.

iMoox.at. (2019, März). *[lul1] Von Lerntheorien zum didaktischen Design.* Zugriff am 15. August 2019 auf https://www.youtube.com/watch?v=7Em7bq7wLVg&feature=youtu.be

Donnerbauer, S. (2019, Januar). Zugriff am 15. August 2019 auf https://www.noen.at/hollabrunn/ neuer-maker-space-htl-hollabrunn-baut-offene-werkstatt-fuer-alle-hollabrunn-htl -hollabrunn-133867044#

Katterfeldt, E.-S., Zeising, A. & Lund, M. (2013). Notes on Maker Culture. In J. Walter-Herrmann & C. Büching (Hrsg.), *Fablab: Of machines, makers and inventors (S. 123-126).* Bielefeld: transcript Verlag.

Kneuer, M. (o. J.). *Makerspaces in Österreich.* Zugriff am 15. August 2019 auf http://www.makerszene .at/makerspaces-in-oeterreich/

*Labor für Innovation: Hightech-Werkstatt eröffnet an TU Graz.* (2019, April). Zugriff auf https://science.apa.at/dossier/Labor_fuer_Innovation_Hightech-Werkstatt_eroeffnet _an_TU_Graz/SCI_20190425_SCI83015586848148774

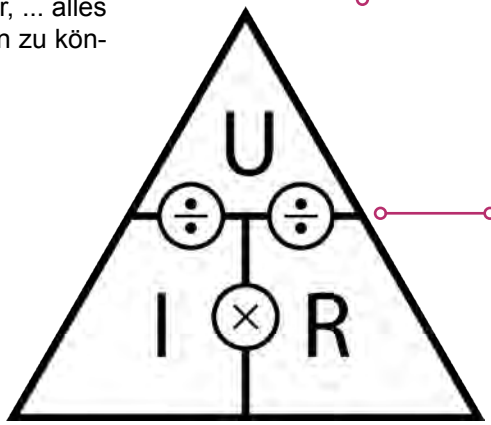
# Elektronik 101

Elektrizität ist die Bewegung von Elektronen. Elektronen erzeugen Ladung, die wir für uns arbeiten lassen können. Eine Glühbirne, das Smartphone, der Fernseher, ... alles zapft Strom ab, um arbeiten zu können.

## Spannung
(voltage)

Formelbuchstabe: U
Einheitszeichen: V Volt

das elektrische Potential zwischen zwei Punkten in einem Schaltkreis




### Ohm's Law

zeigt den Zusammenhang der drei Größen und wie sie sich errechnen lassen

U = I x R
R = U / I
I = U / R

## Strom
(current)

Formelbuchstabe: I
Einheitszeichen: A Ampere

die Elektronen, die sich durch den Leiter bewegen

## Widerstand
(resistance)

Formelbuchstabe: R
Einheitszeichen: Ω Ohm

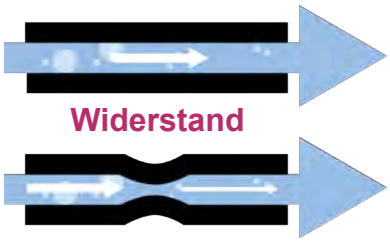
ein Mittel, dass den Stromfluss reduziert

Dadurch, dass Elektrizität vom Menschen nicht gesehen werden kann, wird das Prinzip der Elektrizität oftmals mit einem Wassertank mit angeschlossenem Schlauch verglichen.
Dabei entspricht

die Wassermenge = Ladung
der Wasserfluss = Strom
der Wasserdruck = Spannung
der Schlauchdurchmesser = Widerstand

Der Druck am unteren Ende des Tanks entspricht der Spannung. Das Wasser entspricht der Ladung. Je mehr Wasser im Tank ist, desto höher ist die Ladung und der Druck, also die Spannung am unteren Ende des Tanks.
Der Tank gleicht dabei einem Akku bzw. einer Batterie. Wenn wir quasi den Wasserschlauch aufdrehen, dann geht Wasser verloren, das heißt die Spannung wird weniger. Auch bei einer Taschenlampe wird die Leuchtkraft immer schwächer, sobald die Batterien leer werden. Weniger Druck bedeutet auch, dass weniger Wasser fließt. Der Wasserfluss kann mit dem Strom verglichen werden. Je höher der Druck, desto höher der Wasserfluss und umgekehrt.
Wenn man nun den Schlauch an einer Stelle zusammendrückt, dann fließt weniger Wasser hindurch. Diese Funktion des Regelns des Stromflusses übernimmt der Widerstand.

**Spannung**



**Widerstand**

## das Multimeter

Die Messleitungen müssen in die beiden Buchsen gesteckt werden.

- das **rote Kabel** in die rote Buchse, die mit **VΩmA (Widerstands- und Spannungsmessung) oder 10A (Strommessung)**
- das **schwarze Kabel** in die schwarze Buchse, die **COM** heißt. COM steht dabei für "commons" und bedeutet soviel wie "gemeinsam"

Starte mit der niedrigsten Einheit, sollte es eine 1 anzeigen, heißt das der Wert ist größer als die Anzeige darstellen kann,
dann muss einfach auf den nächsten Anzeigebereich gestellt werden.

## Widerstand messen

1. das rote Messkabel an **VΩmA** anschließen
2. den Drehregler auf einen Wert von **Ω** stellen
3. das rote und schwarze Kabel links und rechts vom Widerstand anlegen

## Strom messen

Damit der Strom gemessen werden kann, muss das Multimeter Teil des Stromkreises werden. Daher muss es in Reihe mit der zu prüfenden Komponente geschalten werden.

1. das rote Messkabel an **10A** anschließen
2. den Drehregler auf einen Wert von A stellen

## Spannung messen

### Gleichspannung vs. Wechselspannung

Die Elektronen wandern von einem Pol zum anderen. Die Spannung und Bewegungsrichtung der Elektronen ändert sich nicht, alles bleibt gleich

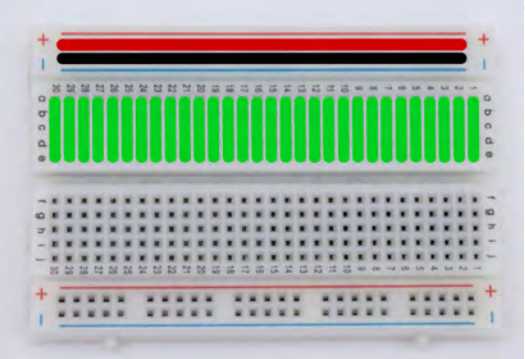
Abkürzung:
**DC (Direct Current)**

Bei Wechselspannung ändert sich die Flussrichtung der Elektronen ständig.

Abkürzung:
**AC (Alternating Current)**

1. das rote Messkabel an **VΩmA** anschließen
2. den Drehregler auf einen Wert von V stellen
3. das rote Kabel an den Pluspol halten und das schwarze an den Minuspol einer Batterie

## Steckbrett
### (breadboard)



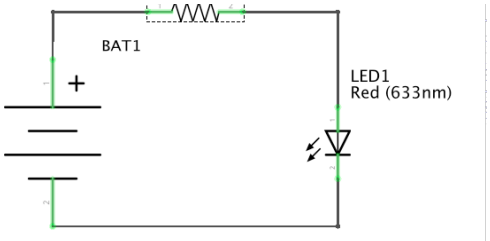
Das Steckbrett oder Steckplatine wird im Englischen als Breadboard bezeichnet. Es dient dazu, Schaltungen aufzubauen und zu testen. Die Bauteile können dabei jederzeit umgesteckt werden. Die Steckplätze sind untereinander verbunden und zwar so, wie auf der Abbildung gezeigt wird. Wenn das Brett waagrecht vor einem liegt, ist in der Mitte eine waagrecht verlaufende Vertiefung zu erkennen. Die Kontakte darüber und darunter sind senkrecht miteinander verbunden. Die oben und unten verlaufenden Leisten am Board besitzen ebenfalls eine Verbindung zueinander. Manchmal gibt es auch eine zweite parallel verlaufende Reihe. Dabei sind die Löcher nur mit den Nachbarn nebeneinander verbunden. Die beiden Reihen sind nicht miteinander verbunden. Normalerweise werden sie verwendet um das Board mit Strom zu versorgen und mit Plus und Minus einer Batterie oder des Arduinos zu verbinden.

## Schaltkreis
### (circuit)

Eine Batterie oder Steckdose hat eine bestimmte Anzahl an Volt – so wird die elektrische Spannung angegeben.
Damit wir Strom benutzen können, muss er im Stande sein, sich zu bewegen. Strom kann nur durch leitfähige Materialien fließen.
Strom fließt immer von einer Spannung zur anderen, wenn man also einen leitfähgien Pfad zwischen einer hohen und niedrigeren Spannungsquelle herstellt, wird der Strom durchfließen. Befindet sich entlang dieses Pfades etwas Nützliches wie beispielsweise eine LED, dann wird diese aufleuchten.
Jede Stromquelle hat zwei Seiten: + und –.
Zwei Bauteile können auf zwei Arten miteinander verbunden werden:

## Serienschaltung

Etwas in Serie oder in Reihe zu schalten bedeutet, die Bauteile werden nacheinander angereiht.




Eine Standard LED hat einen Durchmesser von 5 mm. Normalerweise beginnt sie bei 8 – 12 mA zu leuchten, erhöht man den Strom leuchtet sie heller. Im Datenblatt kann man nachlesen wie viel Strom die LED maximal verträgt – normalerweise 20 mA (wobei der Unterschied zu 15 mA nur minimal ist). Auch die Durchflussspannung $U_F$ ist im Datenblatt nachzulesen. Bei der Widerstandsbstimmung muss die Durchflussspannung berücksichtigt werden (dh vom Gesamtstrom abgezogen werden).

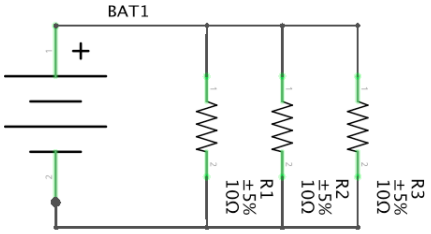
### Berechnung des Widerstandes für eine LED

Stromquelle: 9 V Batterie
gewünschter Betriebsstrom: 18 mA (laut Datenblatt)
Durchflussspannung: 1,8 V (laut Datenblatt)

R = U / I => U = 9 V – 1,8 V = 7,2 V
R = 7,2 / 0,018 = 400 Ohm

## Parallelschaltung

Bei der Parallelschaltung werden die Bauteile nebeneinander und unabhängig voneinander geschaltet.



# Elektronische Bauteile

Der **Transistor** ist das grundlegende Bauelement aller elektronischen Elemente. Er hat im Wesentlichen zwei Aufgaben: **Verstärken und Schalten!**

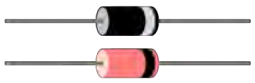
## Transistor

**Kondensatoren** haben die Fähigkeit, **Energie** in Form von Spannung für kurze Zeit zu **speichern**. Es gibt viele verschiedene Arten von Kondensatoren (Keramik, Aluminium, ...)

## Kondensator
(capacitor)

## BASICS

## Diode



Elektrischer Strom kann eine **Diode** nur in eine Richtung passieren

| Farbe | 1. Ring | 2. Ring | Multiplikator | Toleranz |
|---|---|---|---|---|
| farblos | - | - | - | 20 % |
| silber | - | - | - | 10 % |
| gold | - | | - | 5 % |
| schwarz | - | 0 | - | - |
| braun | 1 | 1 | 0 | 1 % |
| rot | 2 | 2 | 00 | 2 % |
| orange | 3 | 3 | 000 | - |
| gelb | 4 | 4 | 0.000 | - |
| grün | 5 | 5 | 00.000 | 0,5 % |
| blau | 6 | 6 | 000.000 | 0,25 % |
| violett | 7 | 7 | 0.000.000 | 0,1 % |
| grau | 8 | 8 | 00.000.000 | 0,05 % |
| weiß | 9 | 9 | 000.000.000 | - |

Beispiel: Braun – Orange – Braun => 1 – 3 – 0 = 130 Ohm
(silber oder gold sollte immer rechts stehen)

## Widerstand
(resistance)

Der **Widerstand** sorgt dafür, den **Strom** zu **reduzieren**, damit empfindlichere Bauteile nicht beschädigt werden. Es ist egal, wie der Widerstand eingebaut wird. Die farbigen Streifen sind ein Code, mit dem man errechnen kann, um welche Stärke es sich handelt.

## KONNEKTIVITÄT

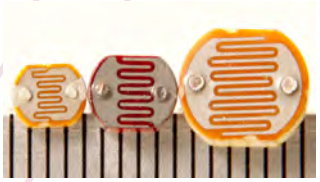
## Kroko Klemmen
(crocodile wire)

## Jumper
(jump wire)

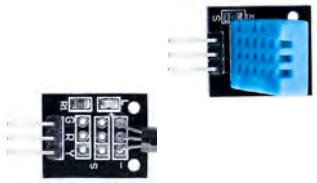
## Bluetooth Module
(low energy)

## IR
(infrared sensor)

## Fotowiderstand
(photoresistor)



## Temperatur und Luftfeuchtigkeit
(temperature and humidity)

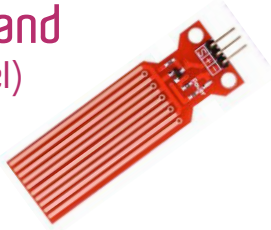


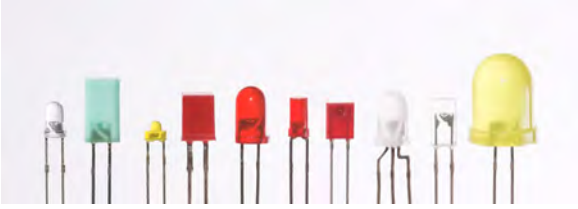
## INPUT
(Sensoren)

## Taster
(button)

## Ultraschall
(ultrasonic sensor)

## Wasserstand
(water level)





## LED – Light Emitting Diode

## Piezo-Lautsprecher

## OUTPUT
(zB Aktoren, LEDs)

## MOTOR

## Stepper-Motor

## Servo-Motor

## DC-Motor

# Programmier-Basics

https://www.arduino.cc/reference/en/

## Was ist eine Variabe?

Eine Variable ist eine Art Platzhalter, der verschiedene Werte annehmen kann.
Zunächst muss eine Variable **deklariert** werden. Das bedeutet man sagt dem Programm “Hey, es gibt hier eine Variable, die heißt ‘soundso’ und besitzt Zahlen (oder etwas anderes) als Werte”. Man schreibt dazu beispielsweise folgendes:

```
int x;
```

Nun hat man dem Programm gesagt, es gibt eine Variable mit Namen x, die Integer-Werte, also ganze Zahlen enthält, das heißt sie ist vom Typ Integer. Noch ist diese Variable leer, das heißt sie hat keinen Inhalt, aber das Programm hat Speicherplatz dafür bereit gelegt. Mit dem "=" Zeichen kann der Variable nun ein Inhalt zugewiesen werden: x = 2;

Es gibt eine handvoll primitiver Datentypen.

| Typ | Beschreibung | Wertebereich | Bits |
|---|---|---|---|
| byte | ganze Zahlen | -128 .. 127 | 8 |
| short | | -32768 .. 32767 | 16 |
| int | | -2147483648 .. 2147483647 | 32 |
| long | | -9223372036854775808 .. 9223372036854775807 | 64 |
| float | Gleitkommazahlen | einfache Genauigkeit | 32 |
| double | | doppelte Genauigkeit | 64 |
| boolean | kann 2 Zustände annehmen | true und false, 1 und 0 | 8 |
| char | Zeichen, Buchstaben, Unicode | 0 .. 65.536 | 16 |

## Was ist ein if-Statement (Bedingung)?

```
if ( Bedingung ) {
        Anweisung;
} else {
        Anweisung;
}
```

Mit der if-Bedingung können bestimmte Programmteile selektiv ausgeführt werden.

Die Bedingung ist ein Bool’scher Ausdruck. **Wenn** die Bedingung wahr ist, **dann** wird die Anweisung ausgeführt, **sonst** wird sie übergangen.

Die Ziele von if und else sind einzelne Anweisungen oder Anweisungsblöcke. Die else-Klausel ist optional. Hat die Bedingung des Ausdrucks den Wert true, wird der Anweisungsblock von if ausgeführt, andernfalls wird, falls vorhanden, der Anweisungsblock von else ausgeführt. Beide werden niemals gleichzeitig ausgeführt. Als verschachtelte if-Bedingung wird sie dann bezeichnet, wenn nach dem else gleich wieder ein if kommt.

| Operatoren | |
|---|---|
| < | kleiner als |
| <= | kleiner als oder gleich |
| > | größer als |
| >= | größer als oder gleich |
| == | gleich |
| != | ungleich |
| && | UND |
| \|\| | ODER |

## Was ist ein Loop (Schleife)?

Mit einer Schleife kann Programmcode wiederholt ausgeführt werden. Am häufigsten werden while-Schleifen und for-Schleifen verwendet.

```
for (Initialisierung; Bedingung; Iteration) {
    Anweisung;
}
```

```
for ( int x; x < 100; x++ ) {
    Serial.println(x);
}
```

In der allgemeinen Form wird bei der Initialisierung der Schleife eine Steuervariable auf ihren Anfangswert gesetzt.

Die Bedingung ist ein Boole'scher Ausdruck, der die Steuervariable testet. Liefert das Ergebnis des Tests den Wert true, wird die for-Schleife weiter durchlaufen. Hat das Ergebnis den Wert false, wird sie beendet.

Der Iterationsausdruck legt fest, wie die Kontrollvariable bei jedem Schleifendurchlauf verändert wird.

```
while (Bedingung) {
    Anweisung;
}
```

```
int x;
while ( x < 100 ) {
    Serial.println(x);
    x++;
}
```

Eine While-Schleife wird solange wiederholt ausgeführt bis die Bedingung nicht mehr wahr ist.

Wird also im Anweisungsblock keine Operation gesetzt, die die Bedingungsvariable verändert, handelt es sich um eine Endlosschleife.

**Inkrement**

count = count + 1;
entspricht
count++;

dasselbe funktioniert auch mit Minus
count -- zählt also hinunter

## Was ist eine Methode?

Methoden, sind wiederverwendbare Programmstücke und werden manchmal auch als "Unterroutinen" oder "Unterprogramm" bezeichnet.

Sie sind prinzipiell dazu da den **Code leserlicher** zu machen und **Redundanzen (Wiederholungen) zu vermeiden**. Ein Stück Code, dass immer wieder verwendet werden soll, ist am besten in einer Methode aufgehoben. Natürlich kann man oftverwendete Programmierzeilen immer wieder mit Copy & Paste einfügen, doch dass führt zu langem, schwer leserlichen Code und es ist auf Dauer mühsam immer wieder ein Codestück rauszusuchen, zu kopieren und einzufügen.

Allgemeine Form:

```
ret-Typ Name (Parameterliste) {
//Methodenrumpf
}
```

Der Name kann frei gewählt werden und wird prinzipiell klein geschrieben. Der ret-Typ (return Typ) ist der Rückgabewert. Und die Parameterliste ist eine Aufzählung an Variablen, die mit einem Komma getrennt werden. Sie wird außerdem immer außerhalb des Hauptprogramms geschrieben, bei Arduino heißt das, außerhalb der loop(){...} Methode.

Wenn man selber eine Methode schreibt, überlegt man sich also, was die Methode tun soll und was man dazu benötigt. Möchte ich beispielsweise ein Methode schreiben zum Addieren zweier Integer-Werte, dann erwarte ich mir von demjenigen, der die Methode aufruft, dass er mir zwei Integer-Werte gibt und keine Strings oder booleans. Diese Werte nennt man Argumente oder Parameter und werden in der Parameterliste näher definiert. Außerdem möchte ich, dass die Methode die Summe zurück gibt. Auch hier muss ich definieren von welchem Typ der Rückgabewert ist. Die Methode würde beispielsweise so aussehen:

```
int add2integers(int x, int y){
        int sum;
        sum = x + y;
        return sum;
}
```

In dieser Methode ist das erste int der **Rückgabewert**. Sobald ich hier einen Wert stehen habe, muss ich **return** und eine Variable von diesem Typ angeben. Sobald return aufgerufen wird, endet die Methode, alles was danach steht wird nicht mehr ausgeführt.
Der Name add2integers ist frei gewählt. Die Parameterliste deklariert zwei Variablen, die nur in dieser Methode bekannt sind. Sie werden anschließend addiert und der Variable sum zugewiesen.

Um die Methode im Hauptprogramm aufzurufen, wird der Name der Methode geschrieben und, falls vorhanden, Parameter angegeben.

```
int number1 = 10;
int number2 = 20;
int sum = add2integers(number1, number2);
```

Wenn eine Methode **keinen Rückgabewert** besitzt, so wird dafür der Begriff **void** als ret-Wert verwendet.

# Arduino



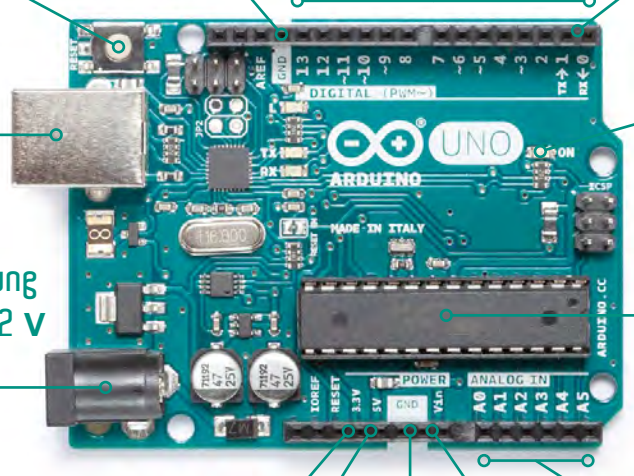
**Open-Source** eine frei benutzbare und bearbeitbare Software oder Hardware, deren Ziel das gemeinschaftliche, freie Arbeiten und Weiterentwicklen von Dingen ist

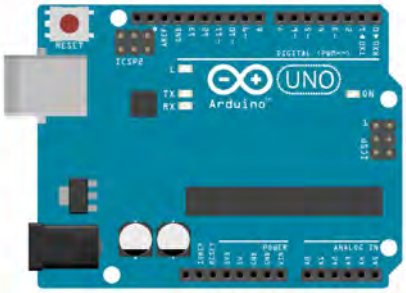
**Elektronik** Technologie, die sich die Bewegung von Elektronen durch leitfähiges Material zu Nutze macht

**Prototyping** eine Grundform (Entwurf), die als Standard oder Vorlage für andere Dinge gilt

**Plattform** eine Hardware-Architektur mit einer Entwicklungsumgebung, auf der Software laufen kann.

PIN 13 verfügt über einen Vorwiderstand, dh hier kann jederzeit eine LED angebracht werden, ohne Angst haben zu müssen, dass diese durchbrennt



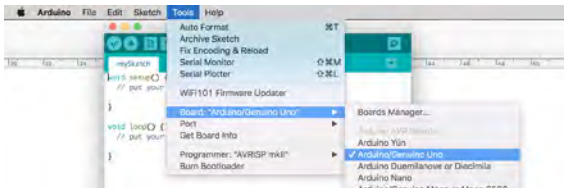



| DIGITAL | | us. | ANALOG |
|---|---|---|---|
| binär | | | Wertebereich |
| true | false | | |
| 1 | 0 | | 0 .. ∞ |
| on | off | | |
| HIGH | LOW | | |
| discrete & finite | | | continous & infinite |

# Arduino IDE

Um den Arduino zu programmieren, wird eine Software – die Entwicklungsumgebung IDE – benötigt.

Diese ist gratis downloadbar unter
https://www.arduino.cc/en/Main/Software



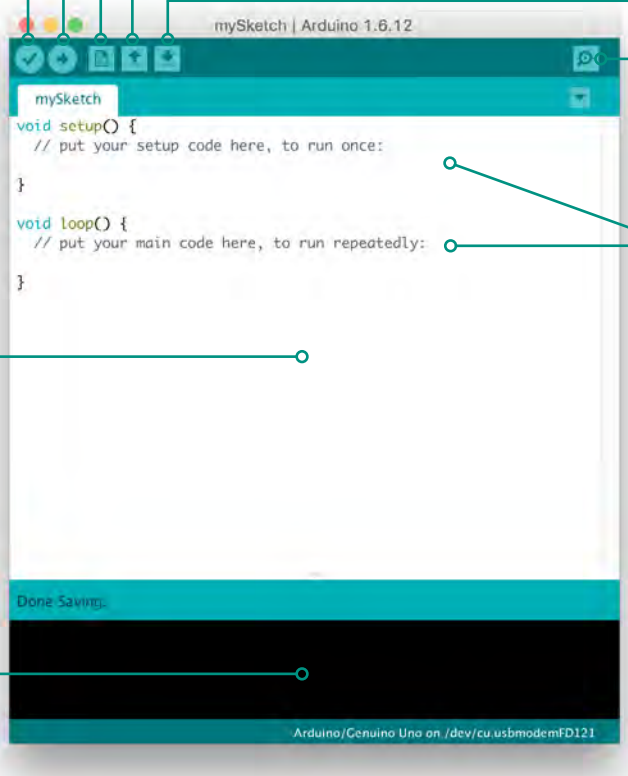

## Verbinde das Arduino Board mit der IDE

Tools > Board > Arduino

## Wähle den richtigen Port aus

Tools > Port > usbmodem (Arduino)

Öffne deinen ersten Sketch unter
**File > New**
Als Sketch werden die Programme auf dem Arduino bezeichnet.




Lädt den Sketch auf den Arduino

Verifiziert den Sketch

überprüft das Programm auf Fehler

Öffnet einen Sketch

Neuer Sketch

Speichert einen Sketch

Serial Monitor

die zwei vorgegebenen Funktionen setup() und loop()

wird 1x zum Programmstart ausgeführt

wird immer wieder wiederholt

Text Editor

Terminal

zeigt verschiedene Mitteilungen an
=> ein Fehler im Programmcode
=> erfolgreich durchgeführter Upload

## Serial Monitor

Der serielle Monitor muss in der setup() Funktion mit Serial.begin(9600) gestartet werden.
9600 gibt dabei die Frequenz an – bits per seconds (Baudrate)
Serial.print();
Serial.println();
geben dann Text (unter Anführungszeichen) oder Variablen aus.
Beides kann auch durch ein + verbunden werden:

**Serial.println("Text unter Anführungszeichen " + variablenName);**

## Libraries

Es gibt eine riesige Arduino Community, die auch eigene Libraries schreibt. Eine Library – ein Bibliothek – bietet schon jede Menge fertiger Funktionen und Programmteile. Sie erleichtern die Programmierung mit Arduino ungemein.
Eingebunden wird eine Lib über:

**Sketch > Include Library > Manage Libraries**

# Arduino Beispiele

## Bringe eine LED zum Leuchten!

Für das allererste Programm wird nur der Arduino und ein USB Kabel benötigt. Der Arduino wird angeschlossen und das Beispielprogramm wird geladen:

**File > Examples > 01.Basics > Blink**

**pinMode(pin, mode)** => ob ein Pin als Input oder Output fungieren soll

**digitalWrite(pin, value)** => wenn der Pin als OUTPUT konfiguriert wurde über pinMode(), dann schaltet der Wert HIGH den Pin auf 5V (oder 3.3V bei 3.3V Boards) oder der Wert LOW auf 0V (ground)

**delay()** bedeutet Verzögerung, der Wert 1000 ist in Millisekunden angegeben

Blink §

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

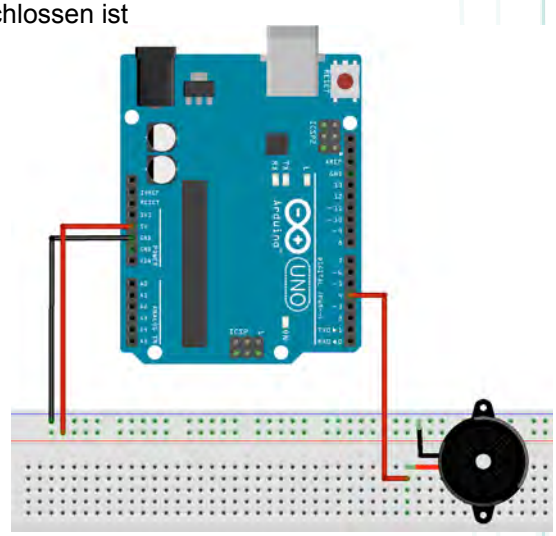
Es kann zusätzlich eine LED bei Pin 13 angeschlossen werden, diese kann direkt in den Arduino gesteckt werden, das lange Beinchen (Plus) wird dabei in Pin 13 gesteckt und das kurze Beinchen (Minus) in den Ground Pin daneben. Das ist nur möglich, weil **Pin 13 bereits über einen Vorstandswiderstand verfügt**. Wäre das nicht der Fall, würde das Lämpchen durchbrennen.

## Teste den Piezo-Lautsprecher!

Der Piezo-Lautsprecher wird auch **passive buzzer** genannt und bietet die Möglichkeit, verschiedene Töne, Melodien und Sirenensignale zu generieren.

Die Erzeugung des Tones basiert auf dem Befehl **tone(x, y)**
=> x-Wert gibt den Pin an, an dem der Lautsprecher angeschlossen ist
=> y-Wert gibt die Tonhöhe an

```
tone §
void setup() {
}

void loop()
  tone(4, 100); // Der Befehl "tone ( x , y )" gibt einen Ton ab
                // x steht dabei für den Eingangspin
  delay(1000);  // Dauer von 1 Sekunde
  noTone(4);    // Der Ton wird abgeschaltet
  delay(1000);  // Der Lautsprecher/Buzzer bleibt eine Sekunde aus
}
```

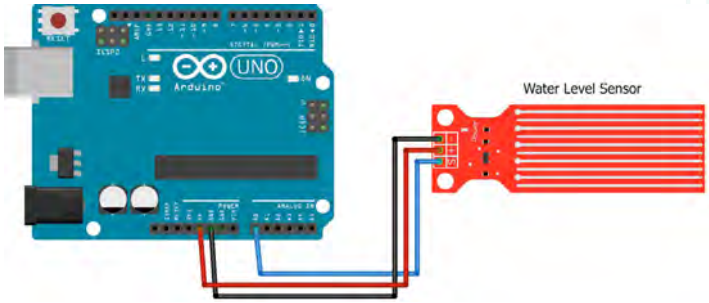


Rufe das folgendes Beispielprogramm auf:
**File > Examples > 02.Digital > toneMelody**

VORSICHT: adaptiere den Pin, den du benutzt

## Teste den Wasserstandmesser!

Verbinde den Water Level Sensor mit dem Arduino und lasse die gemessenen Werte im Serial Monitor ausgeben. Merke dir dabei, welche Zahl bei der Hälfte des Sensors ausgegeben wird.

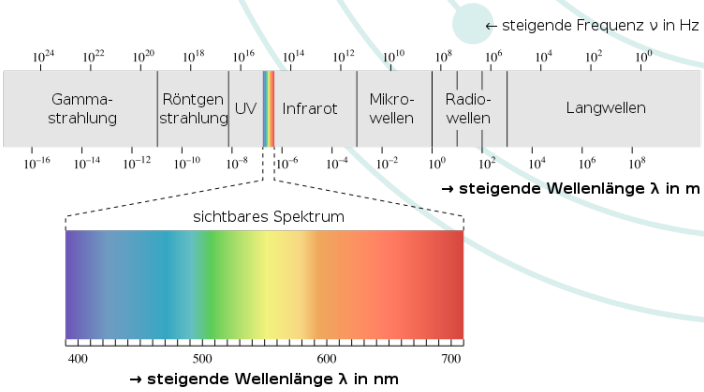



Erweitere nun die Schaltung um einen Buzzer und lass immer dann einen Ton ausgeben, wenn dieser Wert erreicht wird.

```
WaterLevel-Buzzer
#define waterLevel A5
int border;

void setup(){
  Serial.begin(9600);
}

void loop()
  Serial.print("Water level Sensor Value:");
  border = analogRead(waterLevel);
  Serial.println(analogRead(waterLevel));
  delay(100);

  if(border >= 250){ // wenn der Sensor einen bestimmten Wert übertrifft,
                     // dann soll ein Ton abgegeben werden
      tone(4, 100); // Der Befehl "tone ( x , y )" gibt einen Ton ab
                    // x steht dabei für den Eingangspin
      delay(1000);  // Dauer von 1 Sekunde
      noTone(4);    // Der Ton wird abgeschaltet
      delay(1000);  // Der Lautsprecher/Buzzer bleibt eine Sekunde aus
  }
}
```

## Teste die IR-Fernbedienung!

Infrarotstrahlung ist eine Form von Licht – allerdings können Menschen dieses nicht sehen, da es sich in ihrer Frequenz und Wellenlänge von dem für Menschen sichtbaren Licht unterscheidet.
Da es sich also um eine Form von Licht handelt, benötigt der Empfänger und der Transmitter eine direkte Sichtverbindung.




Anders als bei IR benötigt Bluetooth und WiFi keine direkte Sichtverbindung und das Signal kann durch Wände und andere Materialien hindurchgehen.

Lade dazu die Library
- IRremote

IR-Test | Arduino 1.6.12

IR-Test

```
/*
* IRremote: IRrecvDemo - demonstrates receiving IR codes with IRrecv
* An IR detector/demodulator must be connected to the input RECV_PIN.
* Version 0.1 July, 2009
* Copyright 2009 Ken Shirriff
* http://arcfn.com
*/

#include <IRremote.h>

int RECV_PIN = 9;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
Serial.begin(9600);
irrecv.enableIRIn(); // Start the receiver
}

void loop() {
  if (irrecv.decode(&results)) {
  Serial.println(results.value, HEX);
  irrecv.resume(); // Receive the next value
  }
}
```

30 Arduino/Genuino Uno on /dev/cu.usbmodem14121

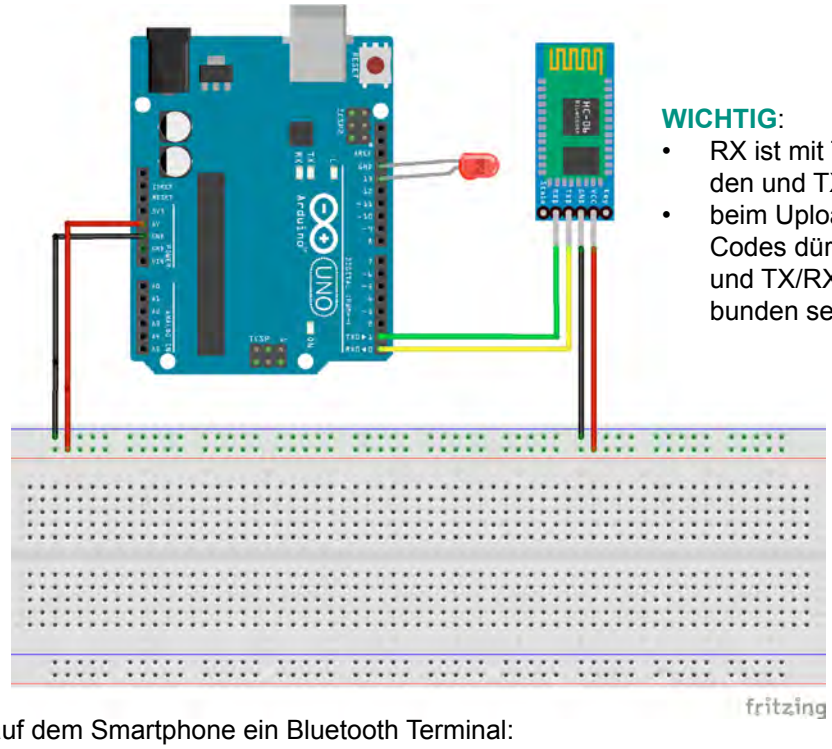
## Teste das Bluetooth Modul HC-08!

Das Bluetooth Modul HC-08 dient zur seriellen Kommunikation zwischen zwei bluetoothfähigen Geräten. Die Pins am Bluetooth Modul sind wie folgt belegt:

Bluetooth-Test | Arduino 1.6.1z

Bluetooth-Test

```
/*
 * Bluetooh Verbindung testen
 * Coder - Lisa Kuka
 * Website - codeandart.at
 */

char data; // Variable, die Daten vom Serial Port erhält
int led = 13; // Pin 13 == OnBoard LED
void setup() {
  pinMode(led, OUTPUT);  // Pin 13 als Output definieren
  Serial.begin(9600);       // Starte die serielle Kommunikation (Serial Communication) bei Port 9600 bps
}
void loop() {
  if(Serial.available()) {      // Wenn Daten über Bluetooth zur Verfügung stehen
    data = Serial.read();         // Lese die Daten aus und speichere sie in data
  }
  if(data == 'H') {           // wenn ein 'H' für HIGH gesendet wurde
    Serial.println("LED ist eingeschaltet!");
    digitalWrite(led, HIGH);  // wird die LED eingeschaltet
  } else if (data == 'L') {  // wird ein 'L' gesendet, dann schalte die LED aus (LOW)
    digitalWrite(led, LOW);
  }
  delay(200);                     //Warte 200 ms bis zum nächsten Auslesen
}
```




**WICHTIG**:
- RX ist mit TX verbunden und TX mit RX
- beim Uploaden des Codes dürfen RX/TX und TX/RX nicht verbunden sein

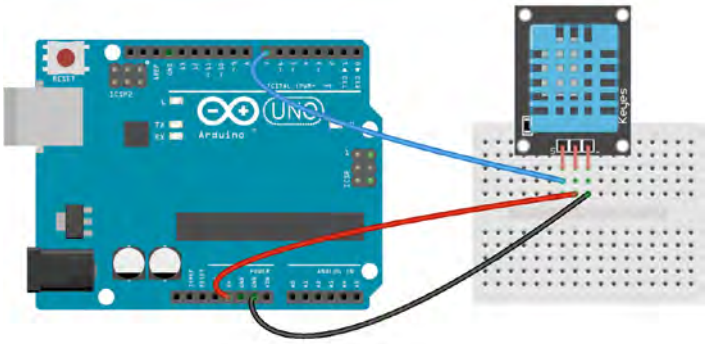
Du benötigst auf dem Smartphone ein Bluetooth Terminal:
- unter **Android** empfiehlt sich **DSD Tech Bluetooth**
- unter **iOS** empfiehlt sich **HM10 Bluetooth Serial Lite**

## Teste den Temperatur und Luftfeuchtigkeitssensor!

Verbinde den Sensor wie auf der Abbildung !

Lade dazu die beiden Libraries
- DHT sensor library
- Adafruit Unified Sensor (wird für DHT benötigt)

Erweitere die Schaltung mit dem Bluetooth Modul.

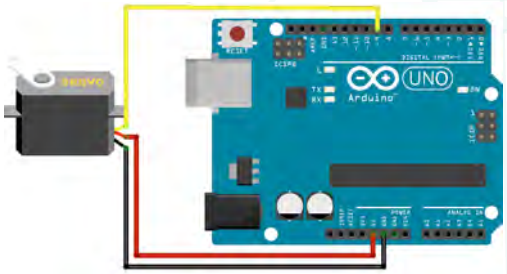


Humidity §

```
#include "DHT.h"

#define DHTPIN 4        // what digital pin we're connected to
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);

char data; // Variable, die Daten vom Serial Port erhält

void setup() {
  Serial.begin(9600);             // Starte die serielle Kommunikation (Serial Communication) bei Port 9600 bps
  dht.begin();
}
void loop() {
  if(Serial.available()) {        // Wenn Daten über Bluetooth zur Verfügung stehen
    data = Serial.read();         // Lese die Daten aus und speichere sie in data
  }
  if(data == 'H') {               // wenn ein 'H' für HIGH gesendet wurde
      // Reading temperature or humidity takes about 250 milliseconds!
      // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
      float h = dht.readHumidity();
      // Read temperature as Celsius (the default)
      float t = dht.readTemperature();
      // Read temperature as Fahrenheit (isFahrenheit = true)
      float f = dht.readTemperature(true);

      // Check if any reads failed and exit early (to try again).
      if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
      }
        Serial.println("Humidity: ");
        Serial.print(h);
        Serial.print(" %\t");
        Serial.println("Temperature: ");
        Serial.print(t);
        Serial.print(" *C ");
        Serial.print(f);
        Serial.print(" *F\t");

        data='L';

    }
  delay(200);                     //Warte 200 ms bis zum nächsten Auslesen
}
```

## Steuere einen Servo-Motor an!

Servo-Motoren können sich im Normalfall 180° drehen.
Erstelle eine Schaltung wie abgebildet.



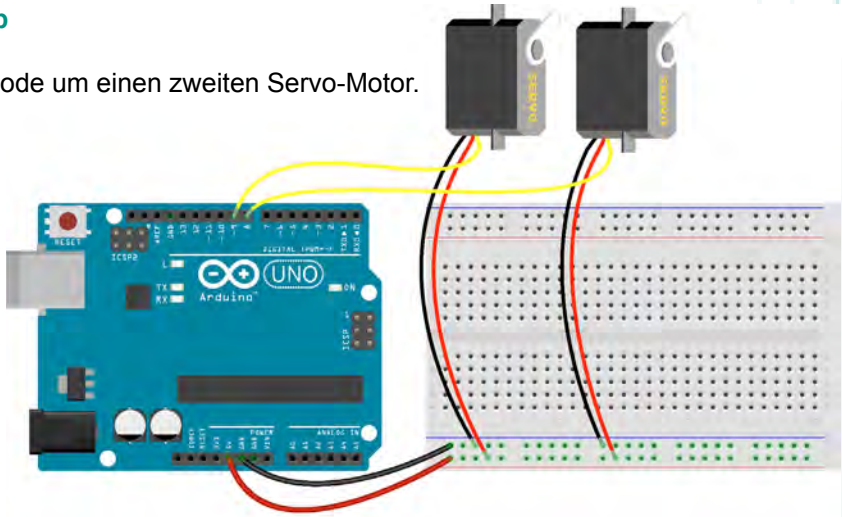
Der Servo-Motor besitzt drei Anschlüsse:

- Strom
- Erdung
- Daten

Lade das Beispielprogramm:

**File > Examples > Servo > Sweep**

Erweitere die Schaltung und den Code um einen zweiten Servo-Motor.



Sweep §

```
/* Sweep
 by BARRAGAN <http://barragan.com>
 This example code is in the public domain.

 modified 8 Nov 2013
 by Scott Fitzgerald
 http://www.arduino.cc/en/Tutorial/Sweep
 modified by Lisa Kuka
*/

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

Servo myservo2;  // create servo object to control a servo

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  myservo.write(0);

  myservo2.attach(8);  // attaches the servo on pin 9 to the servo object
  myservo2.write(0);
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    myservo2.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    myservo2.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

# RoboticHand SchereSteinPapier

## Schere, Stein, Papier

Das Spiel "Schere, Stein, Papier" ist bei Kindern sehr beliebt. Was aber, wenn man niemanden zum Spielen findet? Dann kann man sich mit einem Arduino und zwei Servo-Motoren eine Roboterhand bauen.

### Was wird benötigt?

- Arduino
- USB-Kabel
- 2 Servo-Motoren (oder 5 falls jeder Finger angesteuert werden soll
- Jumper Kabel (mind. 8)

- Leichtschaumplatte oder Karton
- Strohhalme (mindestens 8 mm Durchmesser)
- Holzkügelchen für jeden Finger
- Cutter und Schere
- Faden
- Nadel
- Heißklebepistole

1 Mache dich mit der Anatomie der menschlichen Hand vertraut.

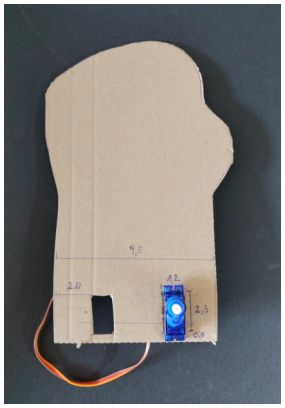
2 Schneide die Grundform der Hand aus Karton oder der Leichtschaumplatte aus.

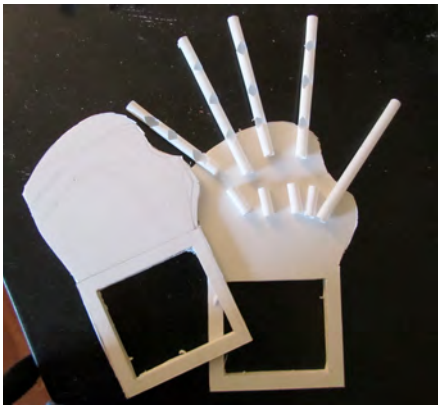
3 Schneide die Finger aus den Strohhalmen zurecht. Bedenke dabei, dass du sie auch an der Grundform anbringen musst, schneide sie also lang genug ab.

4 Schneide die Halme an den Stellen ein, wo sich die Finger abbiegen lassen.

5 Fädle den Faden durch das Holzkügelchen und dannach durch die Strohhalme.

6 Klebe mit Heißkleber die "Finger" an der "Hand" an.



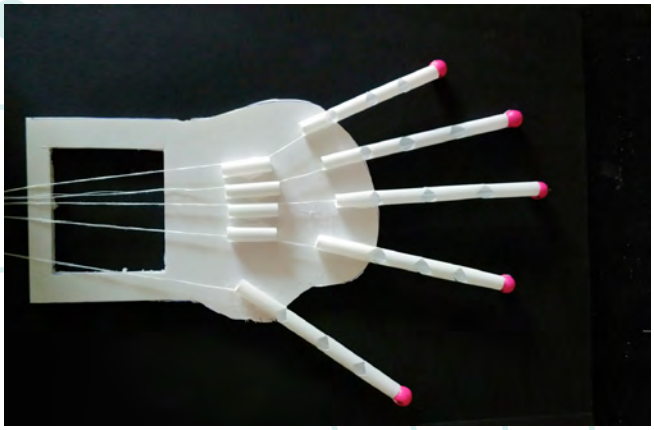


7 Mache dich mit den Servo-Motoren vertraut. Hänge dazu einen Servo-Motor an und lade das Beispiel

**File > Examples > Servo > Sweep**

8 Überlege dir nun, wie viele Zustände eine Hand bei dem Spiel "Schere, Stein, Papier" besitzt.

| | |
|---|---|
| Papier | alle Finger ausgestreckt |
| Schere | zwei Finger ausgestreckt, drei Finger abgewinkelt |
| Stein | alle Finger abgewinkelt |

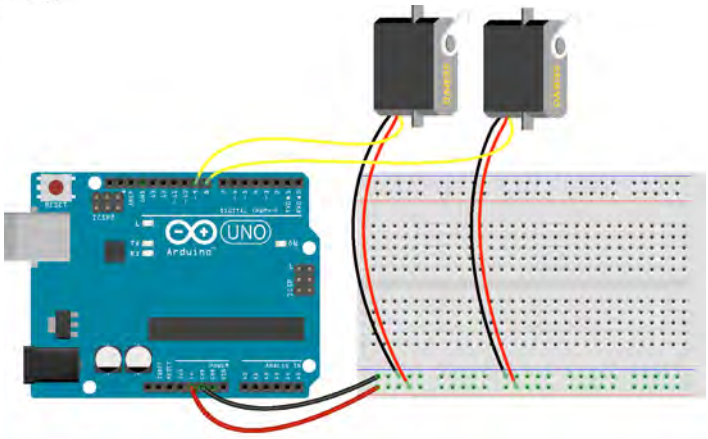


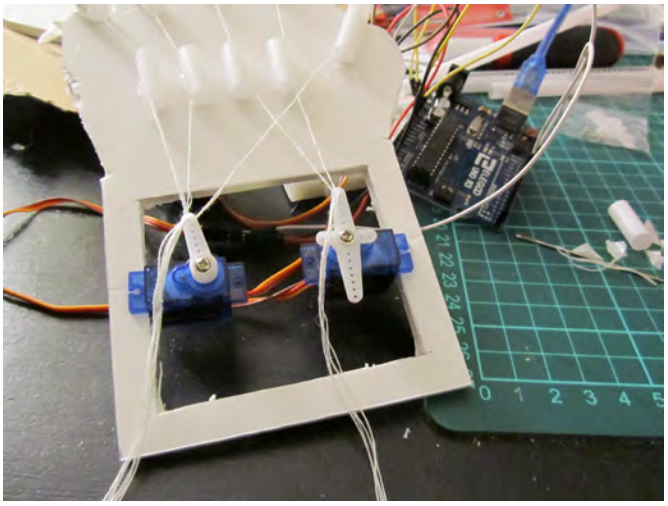
9 Es braucht daher nur zwei Servo Motoren – einen, der Daumen, Ring- und den kleinen Finger steuert und einen, der Zeige- und Mittelfinger steuert.

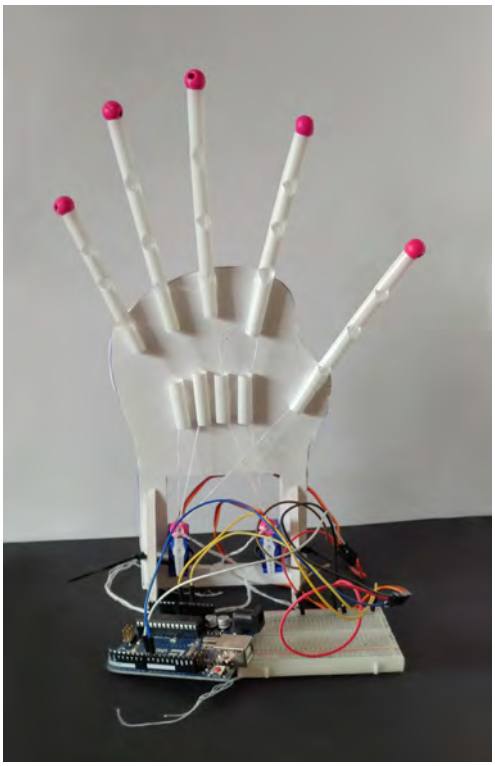
10 Bereite die beiden Servo Motoren vor und verkabele diese wie auf dem Schaltplan beschrieben.

RoboticHand §

```
/* Robotic Hand
 by Lisa Kuka <https://codeandart.at>
 This example code is in the public domain

*/
#include <Servo.h>

Servo thumbRingPinky;  // create servo object to control thumb, ring finger and pinky
Servo indexMiddle;  // create servo object to control index and middle finger

int handPosition = 0;


void setup() {
  Serial.begin(9600);
  thumbRingPinky.attach(8);  // attaches thumb, ring finger and pinky on pin 8 to the servo object
  indexMiddle.attach(9);     // attaches index and middle finger on pin 9 to the servo object
  paper();
}

void loop() {
  handPosition = random(1,4);
  switch(handPosition){
    case 1:
            paper();
            Serial.println(handPosition);
            break;
    case 2:
            stone();
            Serial.println(handPosition);
            break;
    case 3:
            scissors();
            Serial.println(handPosition);
            break;
    default: break;
  }

  delay(5000);
}

void paper(){
  thumbRingPinky.write(0);
  indexMiddle.write(0);
}

void stone(){
  thumbRingPinky.write(180);
  indexMiddle.write(180);
}

void scissors(){
  thumbRingPinky.write(180);
  indexMiddle.write(0);
}
```





**11** Schreibe deinen eigenen Code wie in der Abbildung angegeben.

**12** Lade den Code auf den Arduino.

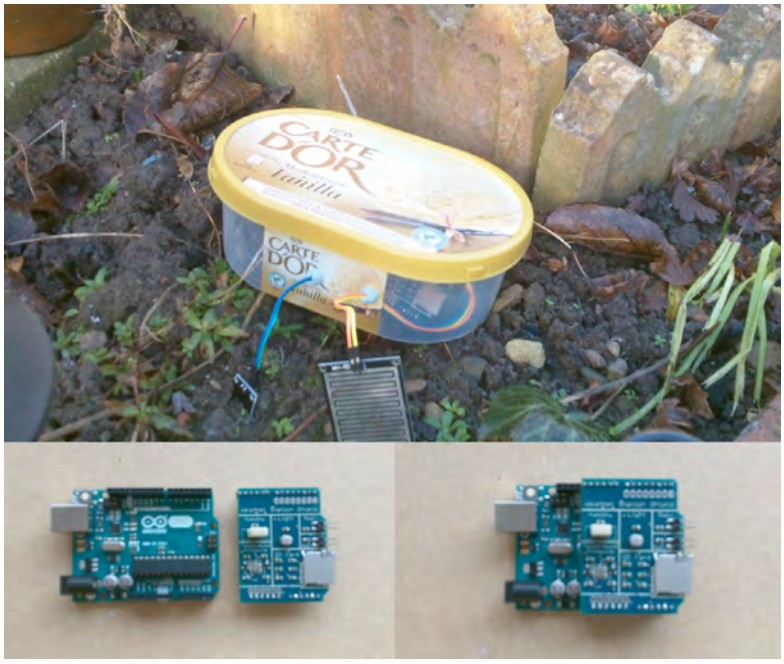
# Weather Station

►https://www.instructables.com/id/Arduino-Weather-Station-1/

Bau dir deine eigene Wetter Station. Entweder man benützt die Sensoren aus den Übungsbeispielen, oder wie in diesem Fall das Arduino Wetter Shield. Es beinhaltet Sensoren wie einen Lichtsensor, Feuchtigkeitssensor, Regenmelder, Bodenfeuchtigkeitssensor, Temperatur- und Drucksensor. Außerdem beinhaltet das Shield eine microSD-Karten Slot, so können die Daten direkt im CSV Format aufgenommen werden und mit einem Tabellenkalkulationsprogramm ausgelesen werden.

## Was wird benötigt?

- Arduino
- Wetter Shield
- Box/alte Eisverpackung
- PC/Mac
- Schere/Messer
- Ladegerät
- microSD Karte




1 Vorbereiten des Codes. Dazu werden folgende Libraries verwendet:

- Weather Station
- Sparkfun BMP180
- SDFat

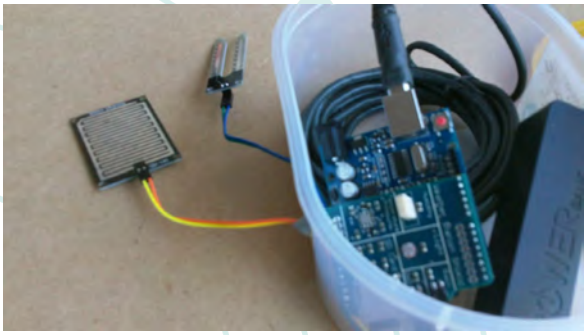
Laden den Example Codes:

**File > Examples > Weather Station > Datalogger**

2 Da der Arduino nicht wasserdicht ist, muss eine Box gebaut werden. Auf den Abbildungen wurde dazu eine alte Eisverpackung verwendet.

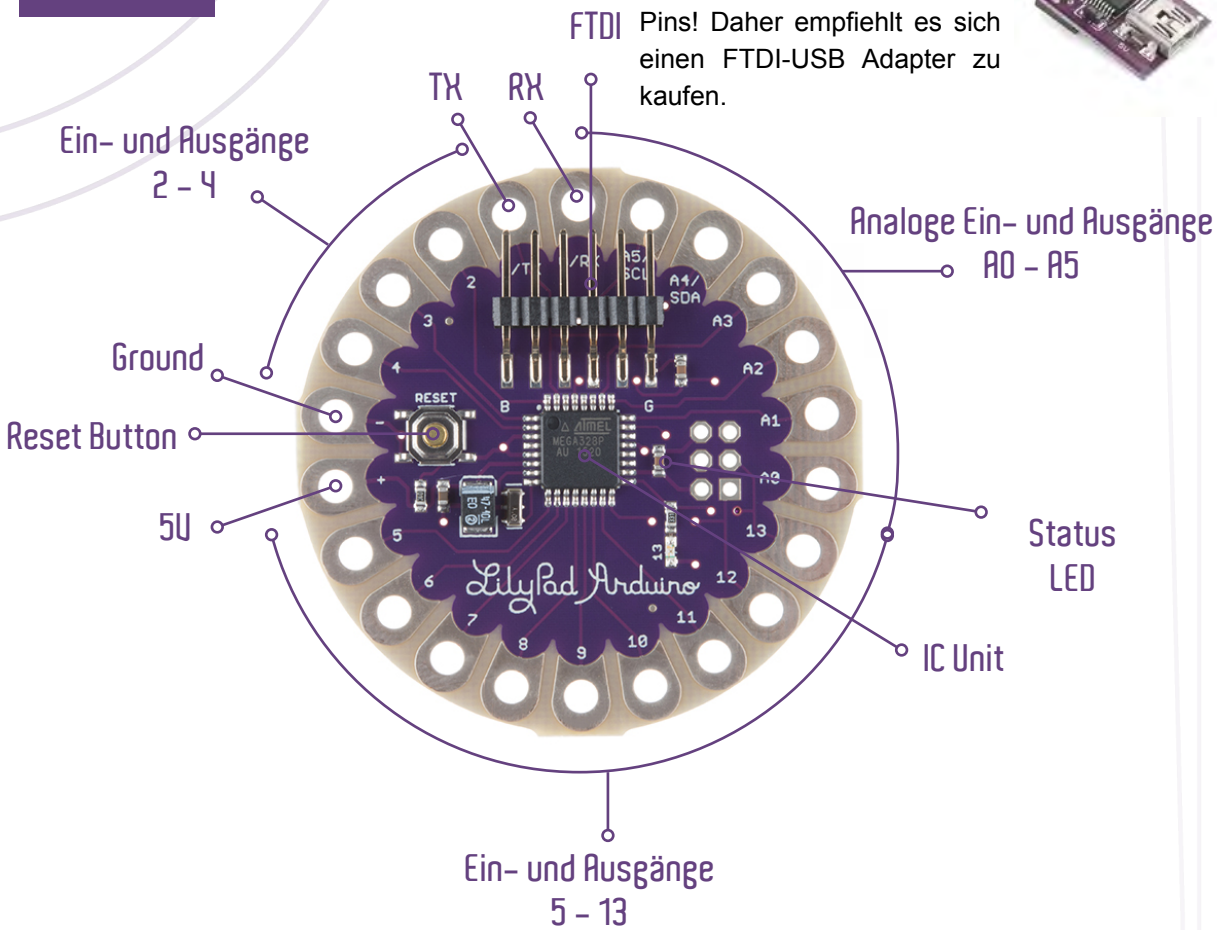
3 Nachdem man die Box ein paar Tage draußen gelassen hat, können die Daten der SD-Karte ausgelesen werden.

Überlege dir ruhig, wie das Projekt erweitert werden kann. Zum Beispiel könnte man ein Bluetooth Modul hinzufügen, um die Daten so kabellos zu übertragen.



# LilyPad

Das LilyPad verfügt über keinen USB Port, sondern FTDI Pins! Daher empfiehlt es sich einen FTDI-USB Adapter zu kaufen.




Im Grunde genommen ist das LilyPad nichts anderes als ein Arduino in einer kreisrunden Bauform. Die Löcher der Pins sind etwas größer und dafür ausgelegt, dass sie mit leitfähigem Garn angenäht werden können. Es wurde außerdem darauf geachtet, dass alle Komponenten sehr flach angeordnet sind, um beim Tragen nicht stören.

Der Hersteller gibt an, dass er nicht dafür haftet, wenn man es wäscht, allerdings gibt dieser schon an, es zu waschen. Einen Kompromiss bieten daher auch andere Varianten, wie zum Beispiel das LilyPad Arduino Simple Snap, das mit Druckknöpfen befestigt und auch wieder entfernt werden kann. Außerdem hat diese Variante bereits einen LiPo-Akku eingebaut.

Das LilyPad bietet außerdem noch eine Vielzahl an weiteren Bauteilen wie LEDs, Sensoren, Battery Clip, etc. Sie haben den Vorteil, dass sie sehr flach verarbeitet sind und meistens schon einen Vorwiderstand integriert haben. Man muss daher nicht mühsam berechnen, welche Bauteile man noch benötigt.

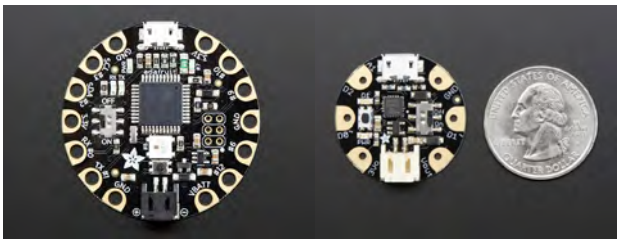
Achtung – es muss dieser Treiber installiert werden:

► http://www.ftdichip.com/Drivers/VCP.htm

## Alternativen:

Adafruit Flora
Arduino Gemma

beide sind ebenfalls kreisrund angeordnet und für den Entwurf von Wearables ausgerichtet, wobei das Gemma etwas kleiner und leistungsschwächer ist.



Flora Gemma

# Tipps für Wearables

Wearables, eFashion, Smart-Fashion, E-Textiles sind sehr ähnliche Begriffe, die mit tragbarer Elektronik in Verbindung gebracht werden.

Elektronische Schaltkreise müssen nicht immer gelötet werden, sie können genauso-gut mit leitfähigem Stoff oder Garn realisiert werden.

1 Textile Leiterbahnen aufbringen durch ...

- leitfähiges Garn aufnähen
- leitfähigen Stoff aufnähen
- Leiterbahn mit Spinnvlies aufbügeln
- Leiterbahn aufkleben

2 Leiterbahn isolieren mit Nagellack, Textilkleber oder durch Einnähen von Textil oder Garn

3 Mache das LilyPad abnehmbar, indem du Druckknöpfe anlötest. Da Druckknöpfe leiten, können die Ports immer noch verwendet werden.

# eFashion

Suche dir ein geeignetes Kleidungsstück – am besten einen Schal, Schuhe oder eine Handtasche.

Bedenke, dass nicht alle elektronischen Komponenten in der Waschmaschine gewaschen werden können. Die LilyPad LEDs können ohne Probleme gewaschen werden, allerdings sollte das LilyPad selbst abnehmbar sein.

Überlege dir, was du machen möchtest:

- Willst du einfach nur LEDs anbringen, sollen diese abwechselnd leuchten oder alle durchgehend?
- Sollen diese durch einen Sensor aktiviert werden?
- Soll es vibrieren?

# Ampli-Tie

by Becky Stern

►https://learn.adafruit.com/led-ampli-tie

Die Ampli-Tie ist eine Krawatte, die ausschlägt, wenn jemand mit dir spricht. Sie verhält sich also wie ein VU-Meter und ist dabei ein richtiger Hingucker!

## Was wird benötigt?

- Flora/LilyPad
- 16 Flora NeoPixels
- PC/Mac
- Mikrofon Amplifier Breakout
- 100mAh LiPoly Akku
- Kabel/Jumper
- 
- leitfähiges Garn
- Krawatte
- Stoff für Akku-Tasche
- Nadel oder Nähmaschine
- Schere

1 Die NeoPixels werden miteinander an die Erdung und an VBATT angeschlossen. Der Datenkanal wird an D6 angeschlossen, während das Mikrofon an 3.3 V, GND und D9 gelegt wird.

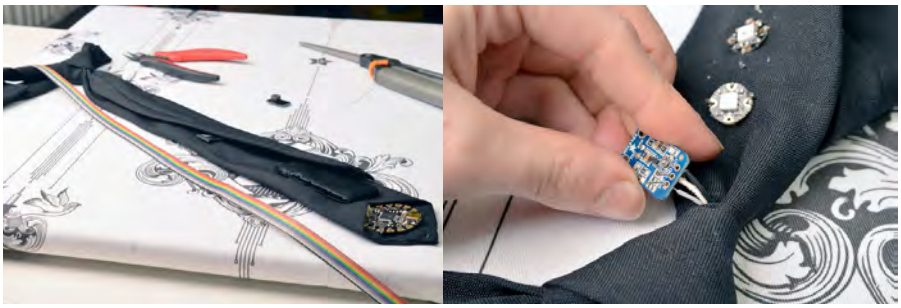
2 Für den Akku wird auf der Rückseite der Krawatte eine kleine Tasche aus Stoff gefertigt

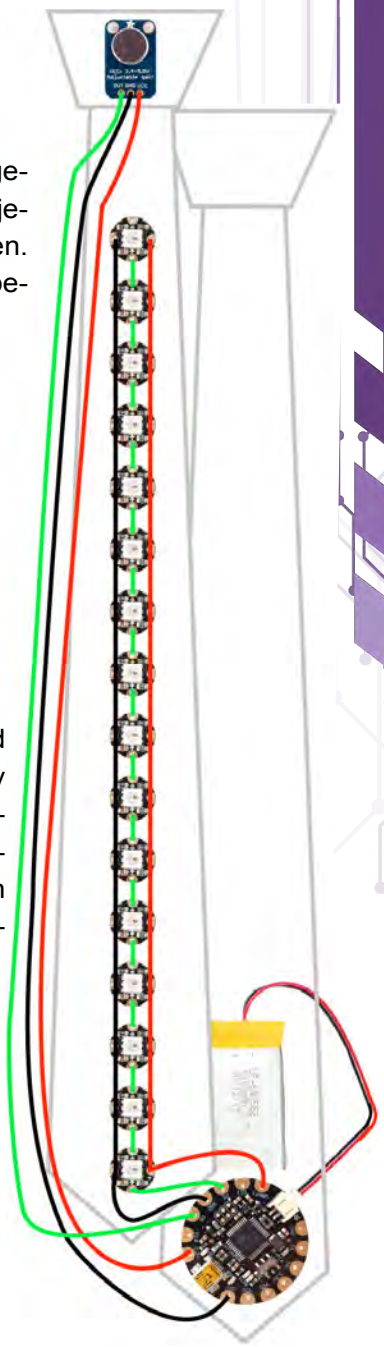
3 Danach werden die 16 NeoPixels aufgenäht. Achte dabei, dass die NeoPixels richtig ausgerichtet sind. Teste mit dem Multimeter nach jedem NeoPixel immer, ob sie noch funktionieren. Der Stromkanal und Erdung dürfen sich nicht berühren.

4 Das Mikrofon kann mit schwarzen Nagelack lackiert werden um zur Krawatte zu passen.

5 Für das Projekt wird die NeoPixel Library "Adafruit_Neoixel" benötigt. Diese wird üblicherweise über den Library Manager gefunden.

6 Der Projekt-Code ist verfügbar unter: https://learn.adafruit.com/led-ampli-tie/the-code





# Sonic Sound Scarf

by Stephanie Mars

►https://www.instructables.com/id/Make-your-own-sonic-sound-scarf/

Der Sonic Sound Scarf ist ein Schlauchschal, der Musik abspielen kann.

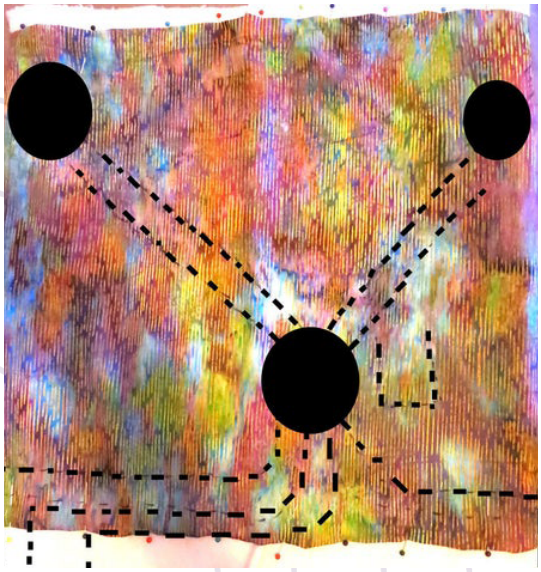
## Was wird benötigt?

- LilyPad MP3 Player
- MicroSD Karte
- ev SD Karten Adapter
- PC/Mac
- 2x Speaker
- leitfähiges Garn
- Li-Ion Akku
- Stoff
- Nadel oder Nähmaschine

1 Suche dir einen geeigneten Stoff für den Schal. Ein Schlauchschal, auch Snood genannt, ist sehr einfach selbst zu machen. Der Stoff wird später einmal in der Mitte gefaltet und zusammengenäht.

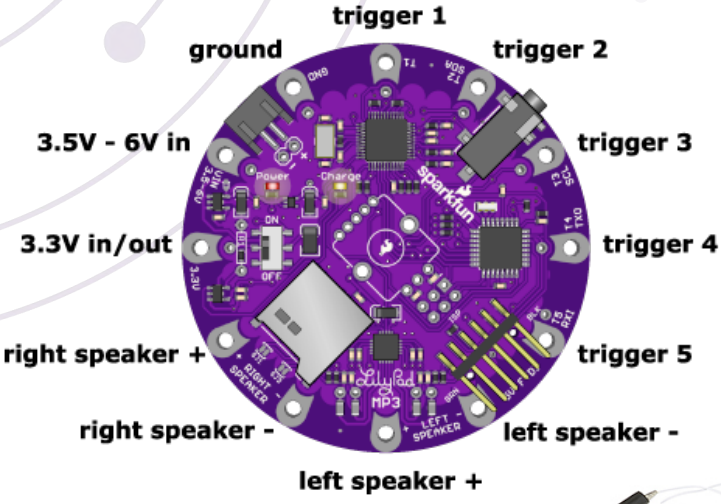
2 Schneide zwei Stücke aus dem Stoff aus – eines wird der innere und eines der äußere Teil des Schals sein. Der äußere Teil sollte dabei etwas größer sein als der innere. Die Größe kann selbst ausgesucht werden, empfohlen ist beispielsweise 100cm x 150cm, damit er zweimal umgewickelt werden kann.

3 Zeichne mit strichlierter Linie ein, wo dein leitfähiger Faden verlaufen soll und an welchen Stellen die Komponenten sitzen sollen. Die Lautsprecher sollten dabei an einer Stelle sein, die nahe am Ohr liegen.

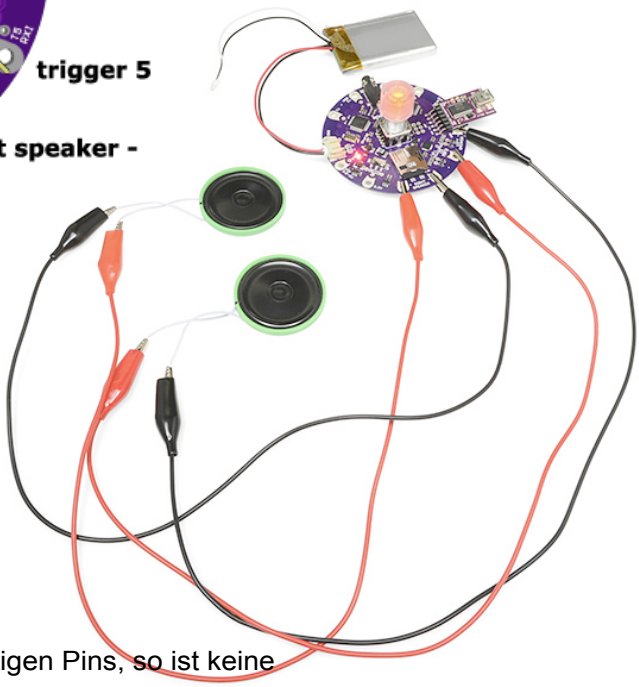


## Anleitung DIY Snood

►http://www.cottonandcurls.com/2012/11/diy-infinity-or-circle-scarf-tutorial-aka-the-snood/




CC Sparkfun



4 Nähe nun das Lilypad an und gestalte eventuell eine kleine Tasche für den Akku und die Lautsprecher.

5 Das LilyPad MP3 kommt mit allen nötigen Pins, so ist keine bis nur wenig Programmierung nötig. Außerdem hat es die Software "Trigger" vorinstalliert, die es erlaubt, bestimmte Dateien abzuspielen, wenn die Input Pins geerdet werden. Wem das nicht ausreicht, kann die "Player" Software runterladen und zusätzliche Funktionen einprogrammieren.

In diesem Beispiel wird eine von drei Dateien abgespielt, wenn der längerer diamantförmige Teil eine der drei dreieckigen Formen berührt.

# Beweg-Dich-Armband

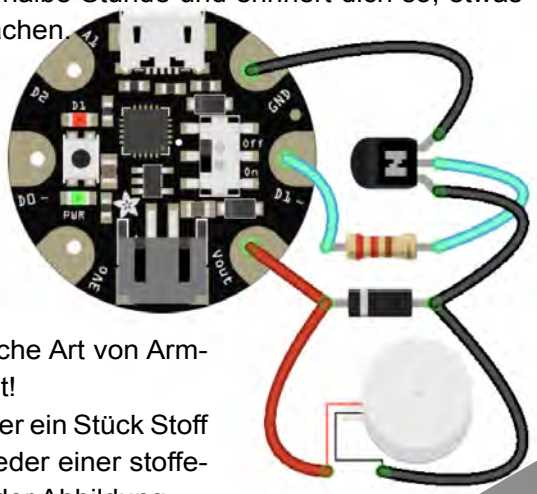
by Becky Stern

►https://learn.adafruit.com/buzzing-mindfulness-bracelet

Wer kennt das nicht – man sitzt den ganzen Tag in der Schule, im Büro oder vor dem Fernseher. Da kann man schon mal vergessen, aufzustehen. Das Beweg-Dich-Armband vibriert alle halbe Stunde und erinnert dich so, etwas Bewegung zu machen.

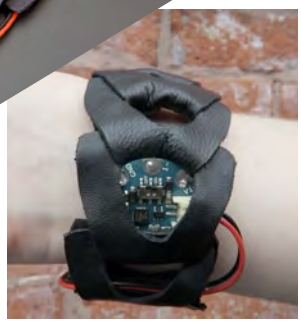
## Was wird benötigt?

- LilyPad oder Gemma
- Vibrierender Mini Motor
- PC/Mac
- 1N4001 Diode
- PN2222 NPN Transistor
- 200-1K Ohm Widerstand
- 100mAh LiPoly Akku
- Schrumpfschlauch
- ev LEDs/NeoPixels
- 
- leitfähiges Garn
- Stoff
- Nadel oder Nähmaschine
- Schere
- Druckknöpfe + Gerät

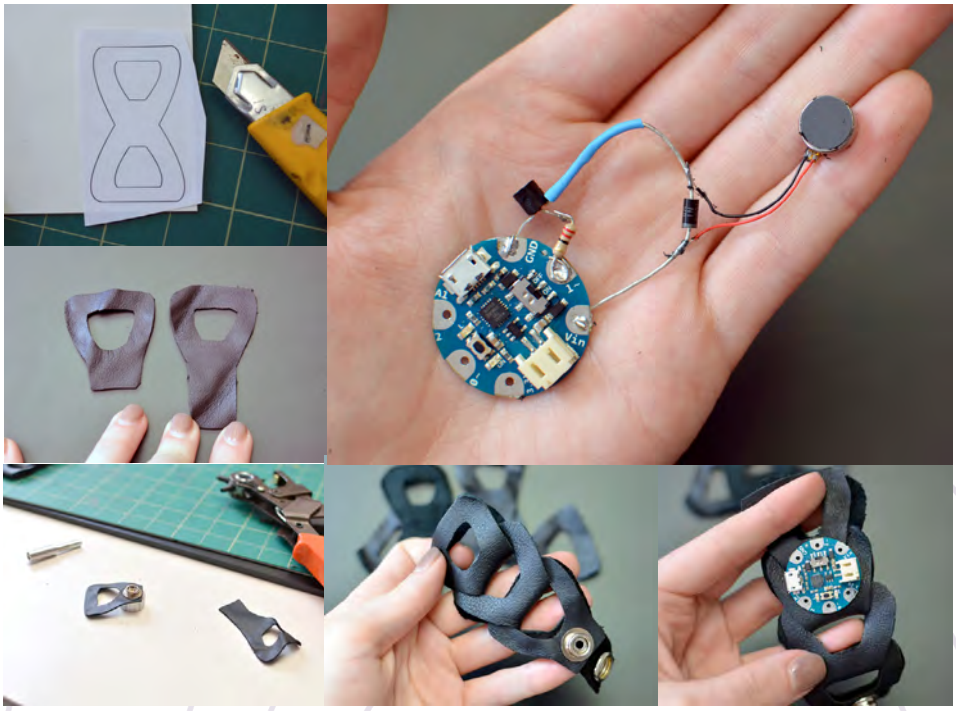


1 Überlege dir, welche Art von Armband du möchtest!
Schneide entweder ein Stück Stoff zu recht oder Glieder einer stoffenen Kette wie in der Abbildung.

2 Die Schaltung kann einfach eingenäht werden oder in die Glieder der Kette eingearbeitet. Die zweite Version ist etwas kniffeliger.



3 Der Projekt-Code ist verfügbar unter:
https://learn.adafruit.com/buzzing-mindfulness-bracelet/code



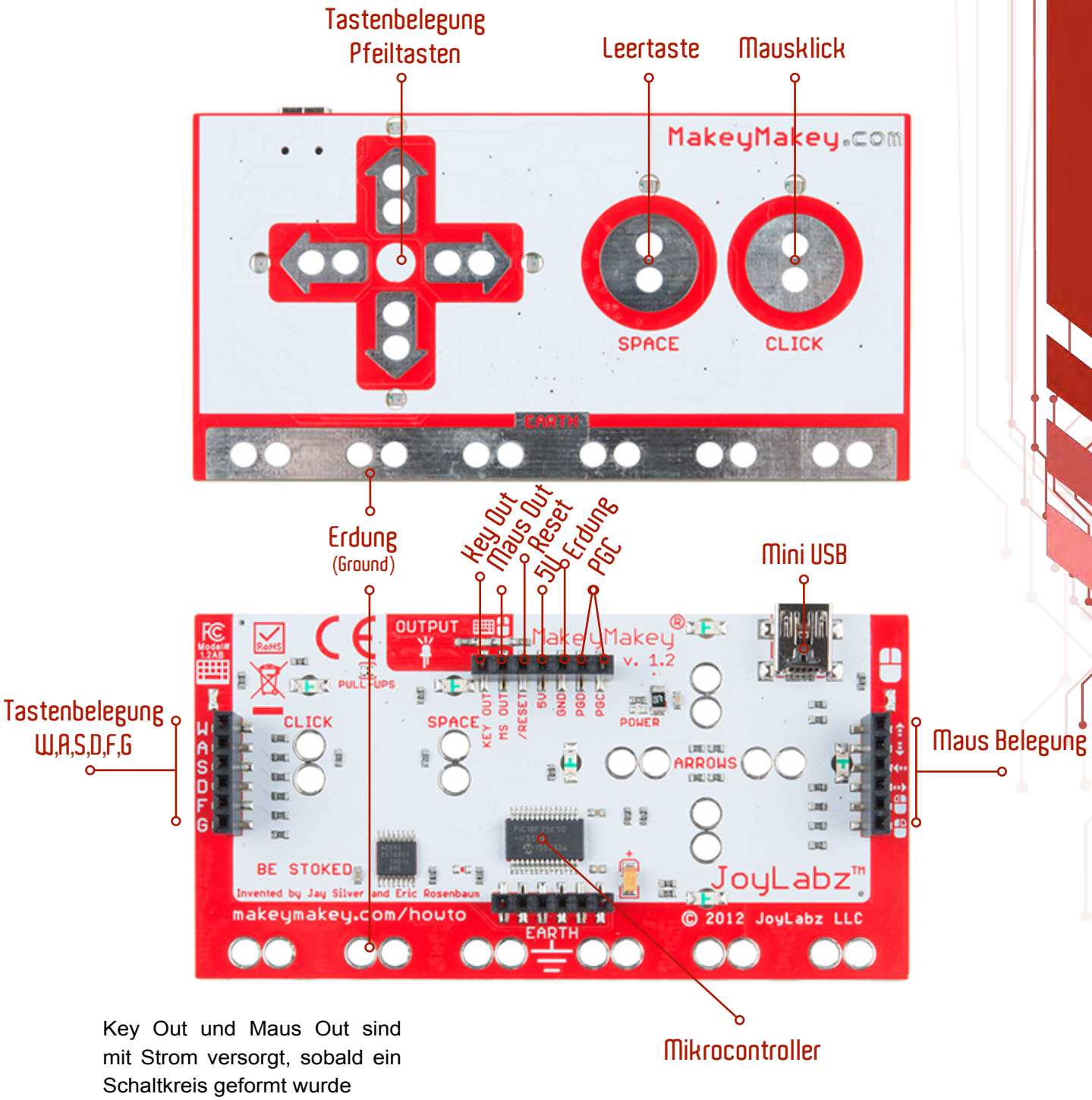
4 Auf dem Armband können nach Belieben NeoPixels angebracht werden.

# Makey Makey

Während Arduino und Raspberry Pi etwas mehr technisches Hintergrundwissen voraussetzen, ist die Handhabung eines Makey Makey etwas einfacher. Jeder, der weiß, wie man einen elektronischen Schaltkreis schließt, kann mit einem Makey Makey umgehen.

Das Makey Makey ersetzt im Wesentlichen ein Eingabegerät wie eine Tastatur oder Maus.

Obwohl das Makey Makey ziemlich selbsterklärend ist, hier ein kleiner Überblick.

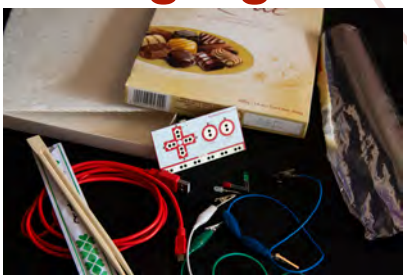



Key Out und Maus Out sind mit Strom versorgt, sobald ein Schaltkreis geformt wurde

#DIY-Surgery

# Do-It-Yourself-Surgery

Das Operationsspiel ist ein beliebtes Kinderspiel – es ist aber auch ganz einfach, es selbst zu basteln.
Also Do It Yourself – DIY



## Was wird benötigt?

- Makey Makey
- Schachtel
- Kroko Klemmen oder Jumper Kabel
- Alufolie
- Piezolautsprecher
- LED
- Essstäbchen
- Karton
- Schere
- eventuell Bildbearbeitungsprogramm

1 Lese dir die Spieleanleitung des Originalspiels im Internet durch

2 Überlege was man alles braucht, wenn man das Spiel selber herstellen möchte

3 Finde ein passendes Bild über den menschlichen Körper im Internet, drucke es aus und klebe es auf eine Schachtel

4 Gestalte Aufgabenkärtchen – schreib darauf, welches Organ entfernt werden soll und welche Aufgabe es im menschlichen Körper hat

5 Schneide vorsichtig die wichtigsten Organe aus, achte dabei darauf, dass die Organe etwas kleiner als die Löcher sind (gegebenenfalls kann das Bild auch noch in verkleinerter Form ausgedruckt werden)

6 Umrande die Organe, die Löcher und die Essstäbchen mit Alufolie

7 Verbinde die Löcher mit dem Makey Makey. Dazu kann die Alufolie benutzt werden

8 LED und Piezo kann an den Key Out Pin angelegt werden

# Blackout Poetry

## by Colleen Graves

Colleen Graves verwendet Makey Makey im Englisch-Unterricht und bezieht sich dabei auf David Saunders. Vor allem im Sprachunterricht ist es oft eine Herausforderung, Jugendliche für Poesie zu interessieren.

### Was wird benötigt?

- Makey Makey
- PC/Mac mit Scratch
- Gedicht (Internet, aus Büchern, etc)
- Graphitstifte
- Kroko Klemmen oder Jumper Kabel

1 Finde ein spannendes Gedicht (aus dem Internet ausgedruckt oder aus einem Buch)

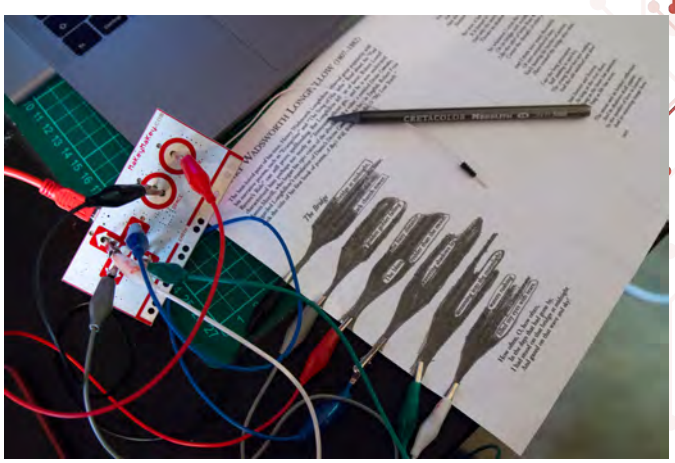
2 Suche 10 oder mehr Worte, die du interessant findest und übermale alle anderen Stellen des Absatzes mit einem Graphitstift. Diese Worte und Wortgruppen ergeben nun ein neues Gedicht

3 Lerne das Makey Makey kennen

4 Lerne Scratch kennen

5 Um das Gedicht nun interaktiv zu gestalten, nimmst du die einzelnen Wörter und Wortgruppen in Scratch auf

6 Verbinde Makey Makey mit Scratch und verbinde nun mit Hilfe des Graphitstifts die Gedichtpassagen mit Makey Makey



► https://colleengraves.org/2017/03/24/resource-interactive-blackoutpoetry-with-makey-makey-and-scratch/

# eInstruments

Gestalte dein eigenes Musikinstrument mit Hilfe von Makey Makey.

## Was wird benötigt?

- Makey Makey
- Schachtel, Karton
- Kroko Klemmen oder Jumper Kabel
- Alufolie
- PC/Mac mit Scratch
- sonstiges Bastelmaterial
- optional Smartphone zum Dokumentieren und Remixen der Aufnahmen

1 Überlege dir ein Instrument und schneide die Form aus Karton aus

2 Überlege dir, an welchen Stellen die Tasten sein könnten, beklebe diese Stellen mit Alufolie

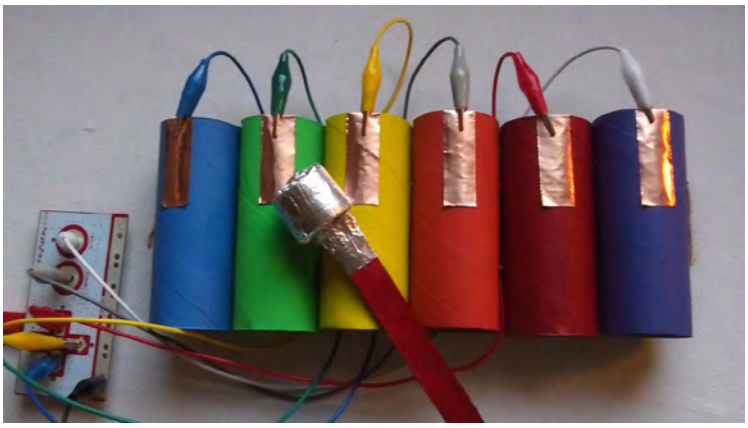
3 Verbinde die Tasten mit dem Makey Makey

4 Überlege, welches Programm am besten für die Tastenbelegung mit verschiedenen Tönen geeignet ist. Recherchiere dazu auf:

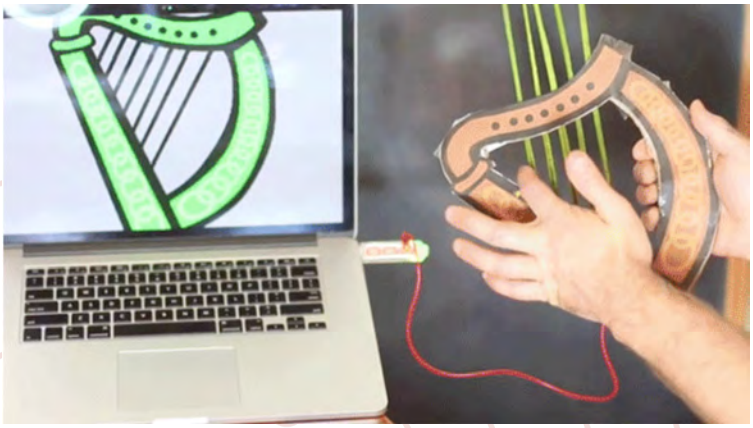
► https://makeymakey.com/apps/

5 Spiele ein Lied mit dem Instrument und nimm es mit der Handykamera auf

6 Mische verschiedene Instrument-Aufnahmen am PC/Mac mit einer entsprechenden Software zusammen



https://twitter.com/1001011k/status/635395999696072705
Xylophon von Digital K



http://www.wired.com/2015/05/19-gizmo-turns-anything-touchpad/

# Sprechende Gemälde

Jedes Gemälde erzählt eine Geschichte. Belebe dein Lieblingsgemälde mit Makey Makey und bringe sie so dazu wortwörtlich eine Geschichte zu erzählen.

## Was wird benötigt?

- Makey Makey
- Abbildungen von bekannten Gemälden
- Karton, Schachteln
- Reißzwecken
- Alufolie
- Kroko Klemmen
- Schere
- Kleber
- PC/Mac mit Scratch

1 Wähle ein ansprechendes Gemälde und drucke es aus

2 Mach dir zu folgenden Punkten Überlegungen:

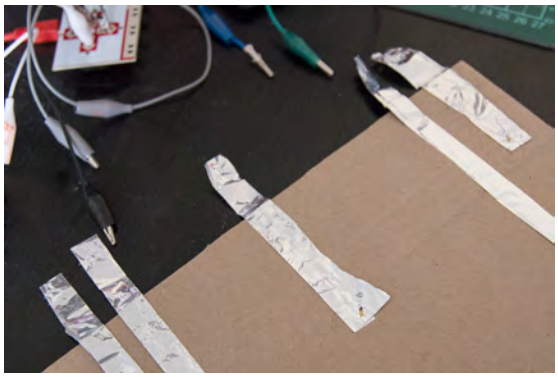
- wann wurde das Bild gemalt
- wer hat das Bild gemalt
- was zeigt das Bild
- überlege dir eine Geschichte
- überlege dir Hintergrundinformationen zu den Personen auf dem Bild
- was fällt besonders auf

3 Befestige Reißzwecken im Bild

4 Verbinde die Reißzwecken auf der Rückseite mit dem Makey Makey

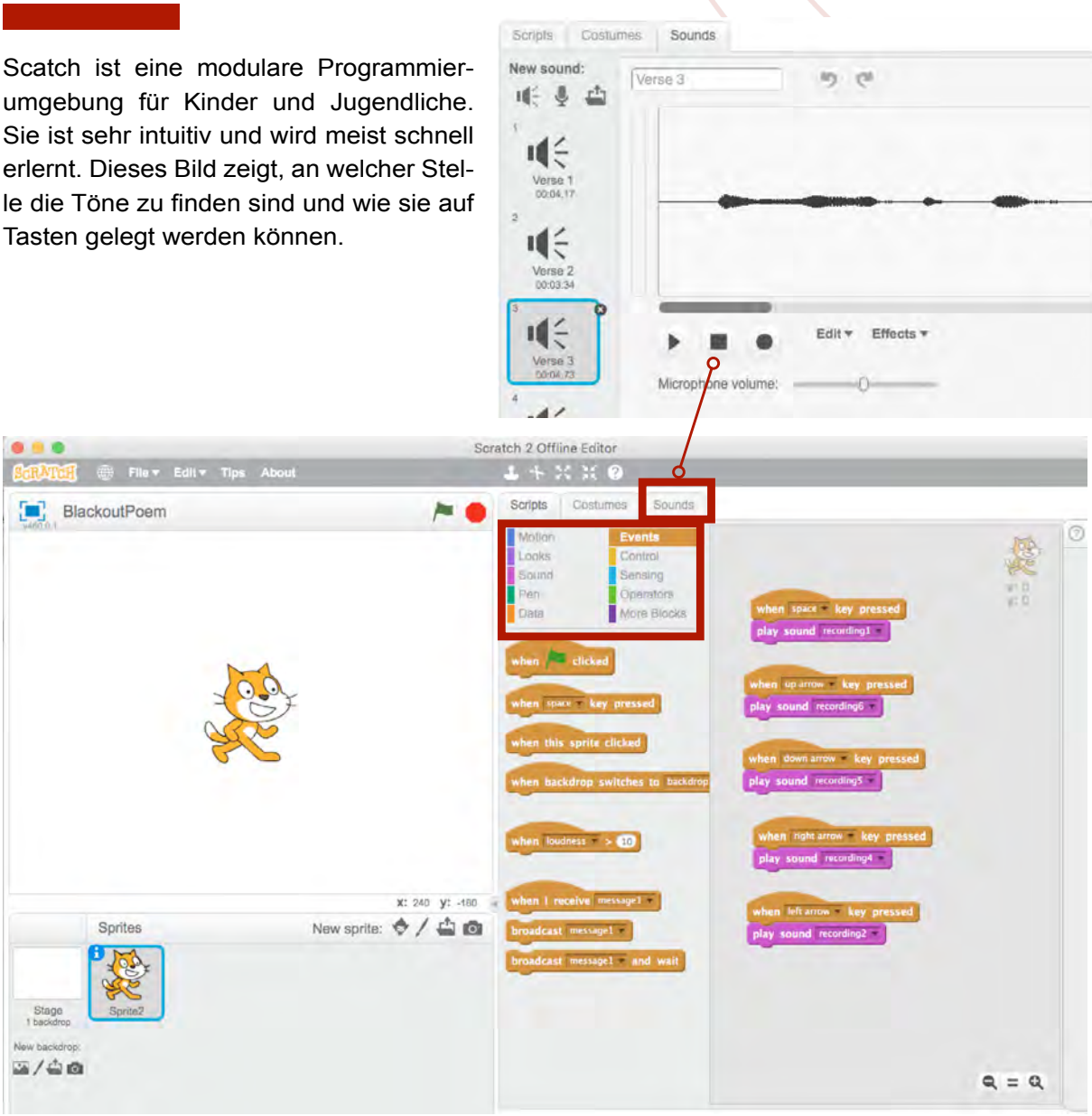
5 Spreche mit Hilfe von Scratch Stichwörter, Teile einer Geschichte oder Informationen zu den einzelnen Teilen des Bildes ein

das Bild stammt von Seurat und wurde von https://www.metmuseum.org/toah/works-of-art/51.112.6/ entnommen.

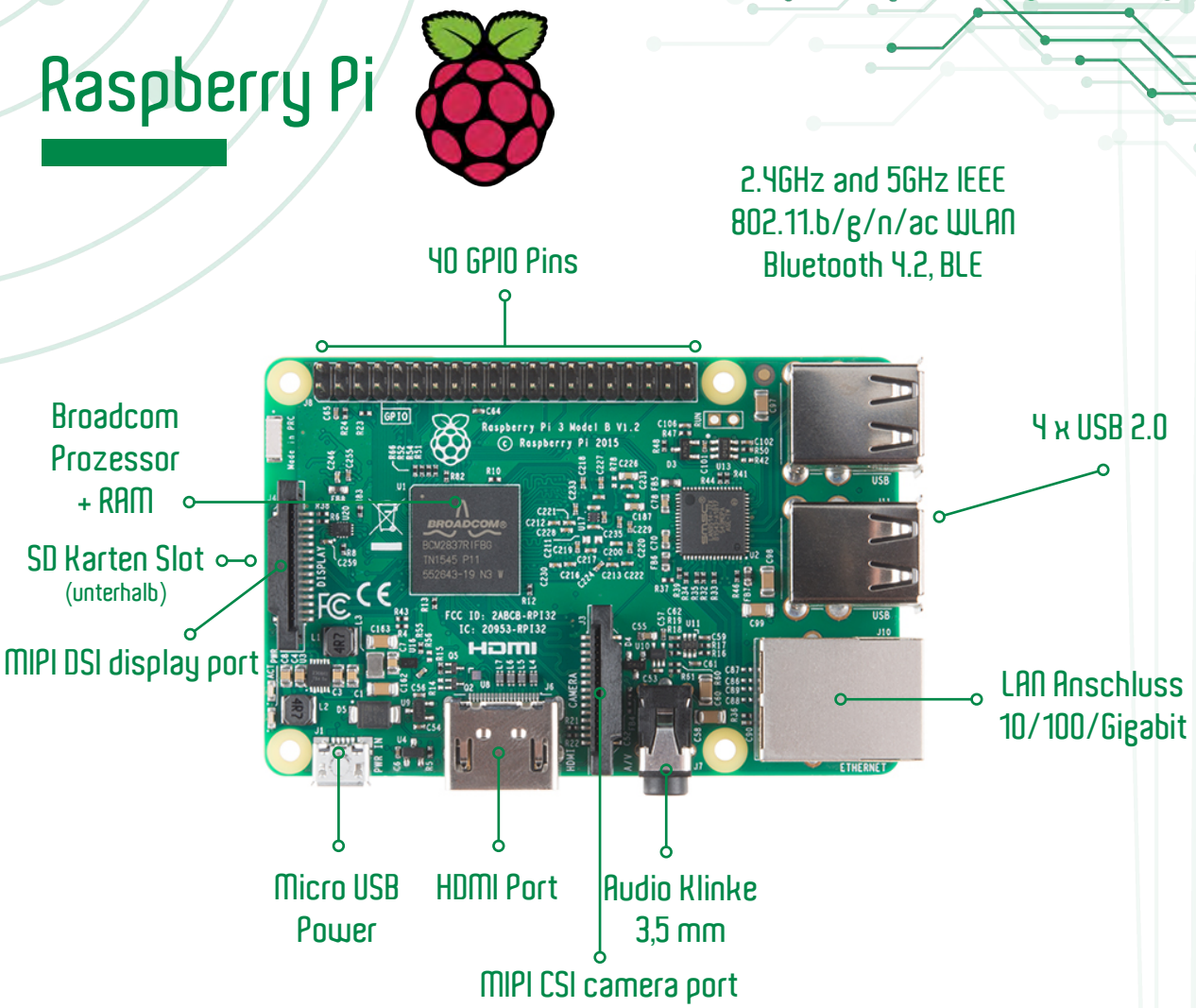


# Scratch

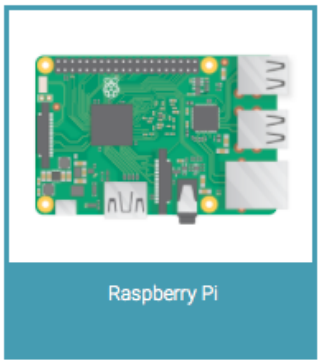
Scatch ist eine modulare Programmierumgebung für Kinder und Jugendliche. Sie ist sehr intuitiv und wird meist schnell erlernt. Dieses Bild zeigt, an welcher Stelle die Töne zu finden sind und wie sie auf Tasten gelegt werden können.



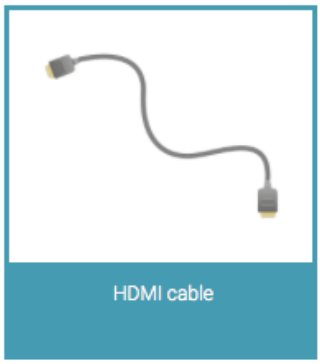
# Raspberry Pi




Der Raspberry Pi ist ein Einplatinencomputer, dh er ist ein komplett einsatzfähiger Computer und daher auch leistungsfähiger.


Raspberry Pi

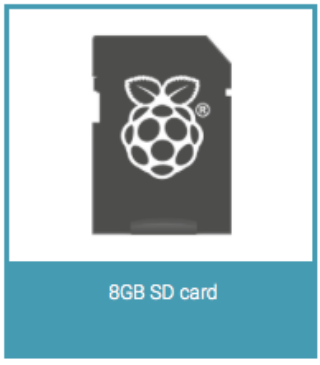
Monitor or TV


HDMI cable

USB keyboard

USB mouse

Power supply


8GB SD card

empfohlene Hardware https://www.raspberrypi.org/learning/hardware-guide/

# Betriebssysteme
(operating system OS)

Software, die den Betrieb eines Computers ermöglicht
Das Betriebssystem vermittelt zwischen Hardware, Software und dem Anwender

## Raspbian

**Raspbian** ist ein Linux Derivat und das beliebteste OS für den Raspberry Pi.

## Noobs

**New Out Of the Box Software**
Noobs ist genau genommen kein Betriebssytem, sondern ein Installer und ermöglicht nur die Installation eines OS.

## Windows 10 IoT

Windows ist wohl eines der bekanntesten Betriebssysteme. **Windows 10 IoT** lässt sich auf dem Raspberry Pi installieren. IoT steht dabei für Internet of Things.

## LibreELEC

**LibreELEC** ist ein Medien-Player, der das Medien-Center Kodi mit einem OS kombiniert. Ein Medien-Center spielt Videos, sowie Musik ab und zeigt Bilder an. Durch installieren von Plug-ins lässt sich der Funktionsumfang erweitern. Jeder Fernseher lässt sich so auch in einen Smart-TV verwandeln.

## RetroPie

https://retropie.org.uk

**RetroPie** ist ein Spiele Emulator. Ein Emulator ahmt die Funktionsweise einer anderen Hardware nach, sodass die originale Software glaubt, sie befindet sich in der originalen Hardware. Mit Hilfe von RetroPie lassen sich Nintendo und Playstation Spiele spielen.

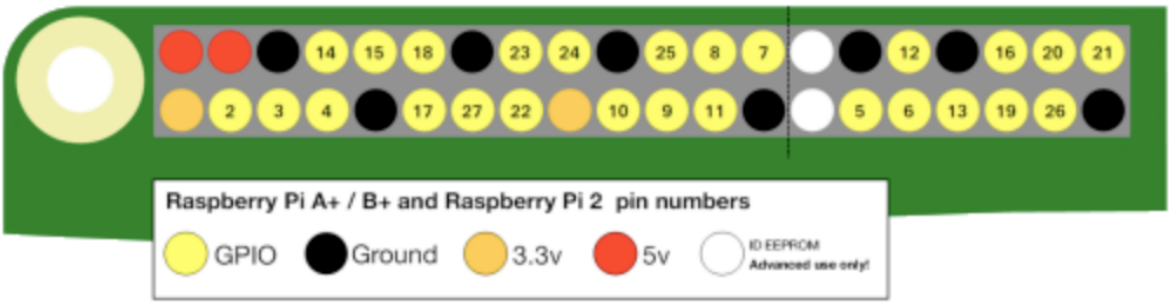
## Installation des OS

**Mit Noobs:** Mit Hilfe von **SD Formatter** (Windows und Mac) ist es möglich, eine SD Karte so zu formatieren, dass anschließend lediglich Noobs per drag&drop raufzuziehen ist. Über Noobs lässt sich anschließend ein OS installieren, von der Raspberry Pi Foundation wird dabei Raspbian empfohlen. Noobs bleibt auf der SD Karte nach installieren des OS erhalten und kann beim Start über die Shift-Taste geöffnet werden und das Betriebssystem erneut installiert werden oder ein anderes ausgewählt werden.

**Direkte Installation mit Etcher:** Mit **Etcher** (für Windows, Linux und Mac) lässt sich das OS direkt installieren und ist schneller als mit Noobs.

► https://www.raspberrypi.org/learning/software-guide/quickstart/

# Raspberry Pi Projekte




entnommen von https://www.raspberrypi.org/learning/parents-guide/physical/

GPIO steht für General Purpose Input Output. Hier können, wie beim Arduino verschiedene Schaltungen mtit Sensoren und Aktoren angehängt werden.

Auch die Sensoren, die im Arduino Kapitel vorgestellt wurden, können über den Raspberry Pi angesteuert werden.

Diese Pins können nun über Python oder Scratch angesprochen werden.

- https://projects.raspberrypi.org/en/projects/physical-computing-with-scratch
  Hier findet man eine einfache Anleitung, wie eine LED mit Hilfe von Scratch zum Leuchten gebracht werden kann.

- https://projects.raspberrypi.org/de-DE/projects/
  Hier findet man Schritt-für-Schritt Anleitungen, um kleine Projekte mit dem Raspberry Pi umzusetzen.

# Alexify My Pi

Du hast dich schon immer für Amazons Alexa interessiert, aber die dazugehörigen Boxen waren dir zu teuer? Dann bastel sie dir doch einfach selber. Der Vorteil zu einem gekauften Alexa-Geräts liegt außerdem darin, dass die Software so an deine Bedürfnisse angepasst werden kann. Auch Projekte, die eine sprachsteuerung bekommen sollen, können so entwickelt werden.

## Was wird benötigt?

- Raspberry Pi
- MicroSD-Karte 16 GB
- USB-Netzteil
- USB-Mikrofon
- kabelgebundene Lautsprecher
- ev USB Soundkarte für besseren Klang
- Monitor, Tastatur und Maus

## 1

Schließe alle Komponenten an und installiere Raspbian auf der MicroSD-Karte. Es empfiehlt sich, SSH zu aktivieren, dann sind Tastatur, Monitor und Maus nicht nötig, sondern der Raspi kann über eine Remote-Verbindung eingerichtet werden.

Remotedesktop-Verbindung einrichten:
► https://tutorials-raspberrypi.de/raspberry-pi-remote-desktop-verbindung/

## 2 Amazon Developer Account

Unter developer.amazon.com kann ein Entwickler-Account erstellt werden.
Es muss hier ein sogenanntes Security Profile erstellt werden. Dazu muss das zukünftige Alexa Gerät bekannt gegeben werden. Ein Wizard hilft durch die Installation.

## 3 Installation der Beispiel Amazon Alexa App

Zum Installieren wird das Terminal auf dem Pi aufgerufen. Mit folgendem Befehl kann das Projekt von Github in das Homeverzeichnis geladen werden:

```
git clone https://github.com/alexa/alexa-avs-sample-app.git
```

Danach müssen die Daten, die zuvor angegeben wurden, im automated_install.sh Skript angeben werden. Diese Bearbeitung kann über einen Text-Editor oder direkt über die Kommandozeile stattfinden.

```
cd ~/alexa-avs-sample-app
sudo nano automated_install.sh
```

Hier befinden sich 3 Zeilen, in denen ProductID, ClientID und ClientSecret angegeben werden müssen.

Nun kann das Skript gestartet werden. Es werden zunächst einige Fragen gestellt und die Spracheinstellungen eingestellt.

## 4 Starten von Amazon Alexa auf dem Pi

Es werden 3 Instanzen des Terminals benötigt um die Anwendung zu starten.

1. Zum Starten des Authentifikation Webservices

   **Terminal 1:** cd ~/alexa-avs-sample-app/samples/companionService && npm start

2. Zur Kommunikation der Anwendung mit AVS

   **Terminal 2:** cd ~/alexa-avs-sample-app/samples/javaclient && mvn exec:exec

3. Damit das Wort „Alexa“ automatisch erkannt wird und die Spracheingabe beginnt

   **Terminal 3:** cd ~/alexa-avs-sample-app/samples/wakeWordAgent/src && ./wakeWordAgent -e kitt_ai

### Genaue Anleitung unter:

► https://tutorials-raspberrypi.de/raspberry-pi-amazon-alexa-deutsch-installieren/

Nun lassen sich Alexa-kompatible Geräte – wie beispielsweise Funksteckdosen und Thermostate – ansteuern.

# Game Station

## mit Bianca Lindorfer

Eine Spielekonsole besteht aus zwei Teilen – einem Rechner und einem Controller. Mit einem Raspberry Pi und Makey Makey kann man sich ohne weiteres seine eigene Gaming-Konsole basteln. Spiele können selbst programmiert werden oder man greift auf einen Emulator zurück.

### Was wird benötigt?

- Raspberry Pi
- Makey Makey
- SD Karte
- Netzteil
- Maus
- Tastatur
- Krokoklemmen und Jumper Kabel
- Schachteln, Karton, ev. Holz
- Knetmasse
- Alufolie

1 Mache dir Gedanken, welche Arten von Spielen und Spielekonsolen es gibt. Wie funktioniert die Navigation durch das Spiel?

## 2 Vorbereiten des Raspberry Pis

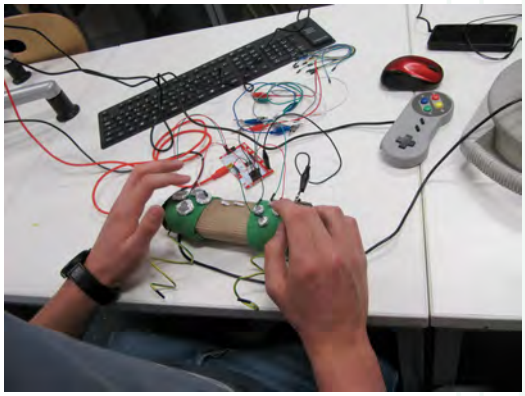
Lade ein passendes Betriebssystem auf den Raspberry Pi. Raspbian ermöglicht es mithilfe von Scratch, selber Spiele zu programmieren oder über den Browser aus dem Internet zu laden. Alternativ ermöglicht der Emulator Retro Pie es, klassische Nintendo-Spiele zu spielen.

## 3 Kreieren eines eigenen Spiels

Falls du dein eigenes Spiel mithilfe von Scratch kreieren möchtest, dann gehe online und suche dir Tutorials aus.
Bedenke, dass die Entwicklung eines eigenen Spiels sehr lange dauern kann.

## 4 Gestalte ein passendes Gehäuse

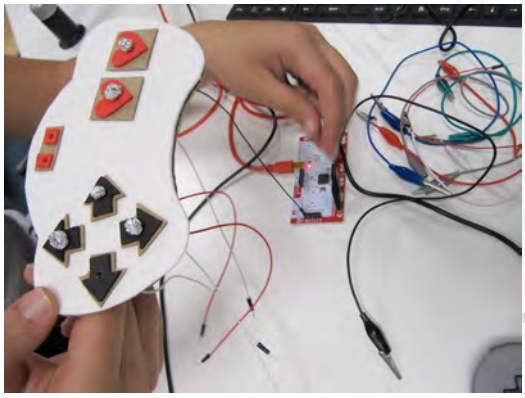
Das Gehäuse für die Konsole kann aus unterschiedlichen Materialien bestehen. Auf den Bildern sieht man einen Nachbau eines klassischen Arcade-Automaten. Es ist allerdings auch möglich, Modelle aus dem Internet zu laden und mittels eines 3D-Druckers zu drucken. Stehen dir diese Möglichkeiten nicht zur Verfügung, spricht auch nichts dageben, mit Karton zu arbeiten.



## 5 Gestalte einen passenden Controller

Mit Makey Makey ist es ein Leichtes, einen Controller selber zu bauen.
Überlege dir:

- Wie wird das Spiel gesteuert?
- Welche Buttons sind unbedingt nötig?
- Wie soll der Controller in der Hand liegen?
- Soll das Design passend zum Spiel sein? Was würde denn hier passen?



# Vom Consumer zum Maker

**Handbuch über Single-Board Platinen**
**für Lehrkräfte, SchülerInnen und Interessierte**

erhältlich unter **CodeAndArt.at**

CC BY NC SA

► Lisa Kuka