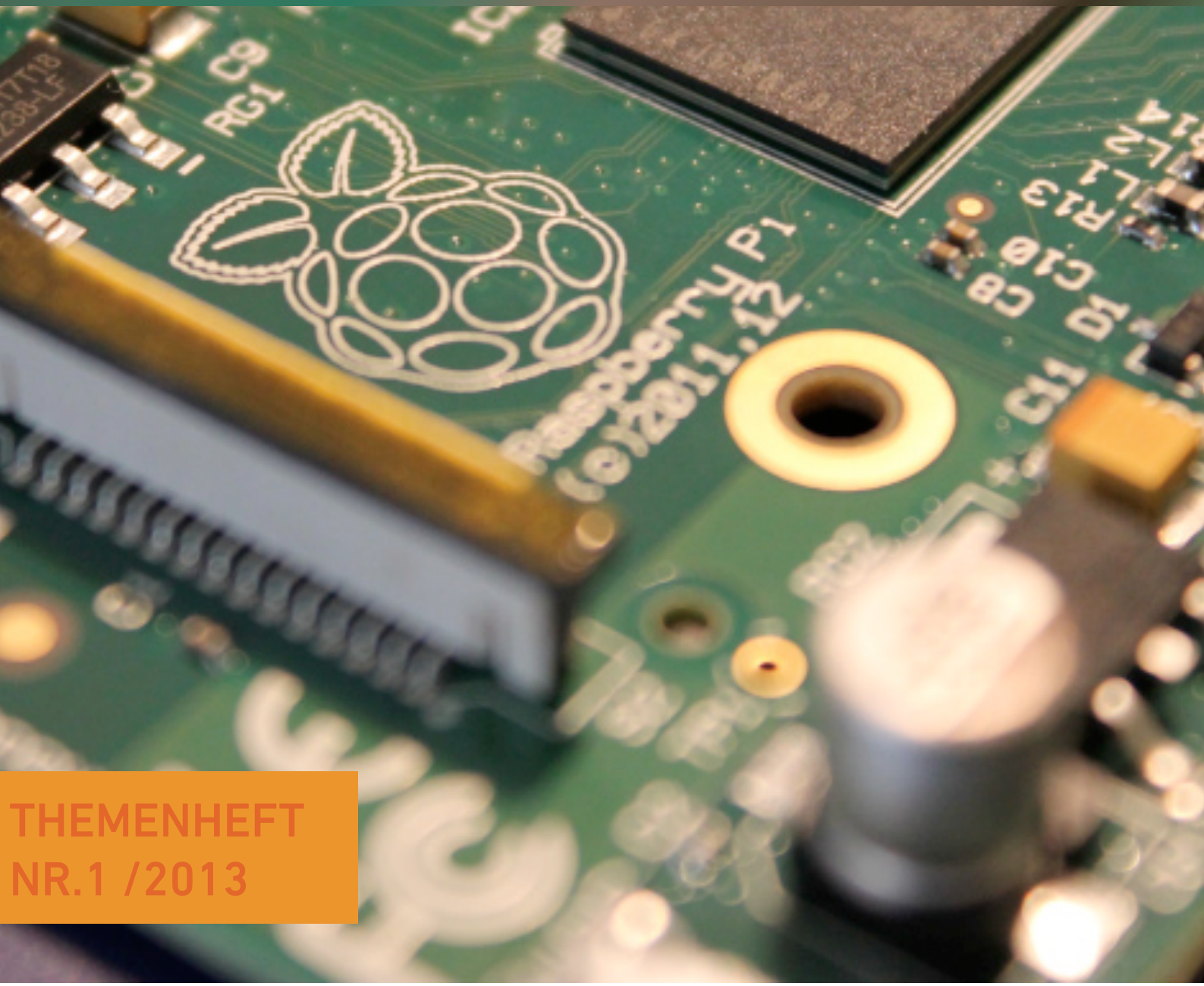
MEDIEN ISTIK .DE

Creative Commons
Unterrichtsmaterialien

Herausgegeben von Tobias Hübner

UNTERRICHTEN MIT DEM RASPBERRY PI

Version 3.0



THEMENHEFT
NR.1 /2013

VORWORT

Die Entwicklung der Computertechnik kennt eigentlich nur eine Richtung: Schneller, mehr Arbeitsspeicher, mehr Festplattenkapazität. Selten kommt es vor, dass ein Computer im Mittelpunkt der Fachwelt steht, dessen Entwickler genau die andere Richtung eingeschlagen haben. Die Rede ist natürlich vom Raspberry Pi, um den es schwerpunktmäßig in diesem Themenheft gehen soll.

Nachdem man anfangs noch mehrere Monate auf ein Exemplar des Minicomputers warten musste, ist das Gerät nun auch in Deutschland problemlos in größeren Stückzahlen erhältlich und so langsam schwappt die Welle der Begeisterung von England auf den Rest Europas.

Auch wenn sich derzeit noch vor allem Hobbybastler auf den Raspberry Pi stürzen, ist der scheckkartengroße Rechner eigentlich für den Einsatz in der Schule konzipiert worden und genau dorthin habe ich in diesem Jahr 10 Raspberry Pi gebracht, um gemeinsam mit Schülerinnen und Schülern der 6. Klasse zu entdecken, wie Computer eigentlich funktionieren und wie man sie programmiert.

Ich hoffe, dass dieses Themenheft für andere Lehrerinnen und Lehrer ein Anreiz ist, sich selbst einmal mit dem Raspberry Pi zu beschäftigen und eine ähnliche Arbeitsgemeinschaft ins Leben zu rufen, denn in Bezug auf die technische Ausbildung unserer Kinder und Jugendlichen an den öffentlichen Schulen ist Deutschland eher ein Entwicklungsland.

Solange keine staatlichen Initiativen bestehen, die diesen Missstand beheben, kommt es auf das Engagement einzelner Lehrkräfte an. Sie können in Arbeitsgemeinschaften und Projekten den Spaß am Umgang mit Technik wecken, denn nur so werden Schülerinnen und Schüler dazu befähigt, ein mündiges Leben in einer immer stärker durch Computer beherrschten Umwelt zu führen.

Tobias Hübner

UNTERRICHTEN MIT DEM RASPBERRY PI

1 / 2013

INHALT

02 *Vorwort*

03 *Inhaltsverzeichnis*

06 *Was ist der Raspberry Pi?*

08 *Die Planung der Raspberry Pi-AG*

10 *1.-3. Stunde: Theorie: Wie funktioniert ein Computer?*

11 *1. Stunde: Der Comptometer*

14 *2. Stunde: Transistoren und Binärcode*

15 *3. Stunde: Logikgatter / Bestandteile eines Computers*

16 *4.-5. Stunde: Der Raspberry Pi*

16 *4. Stunde: Präsentation des Raspberry Pi*

17 *5. Stunde: Die Kommandozeile / Open Source Software / Libre Office*

18 *6.-10. Stunde: Programmieren mit Scratch*

19 *6. Stunde: Einführung in Scratch mit „Scratch Cards“*

19 *7.-9. Stunde: Partnerarbeit mit Scratch*

19 *10. Stunde: Physical Computing mit der LedBorg-Platine*

20 11.-13. Stunde: Programmieren mit Python

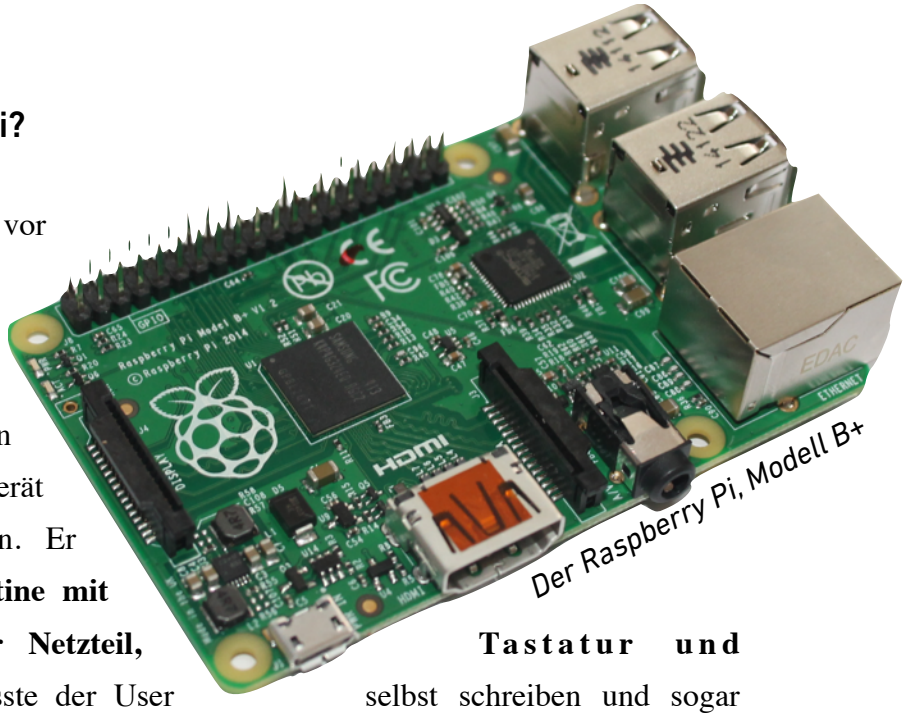
21 14.-19. Stunde: Programmieren mit Minecraft
21 14. Stunde: Python & Minecraft I
21 15. Stunde: Python & Minecraft II
22 16. - 19. Stunde: Physical Computing mit LEDs, Taster und Minecraft
23 20.-30. Stunde: Eigenständiges Stationenlernen

MATERIALIEN

24 M1 - Der Comptometer
25 M2 - Der Transistor
26 M3 - Wie addiert ein Computer Zahlen? - Teil 1: Der Binärcode
27 M4 - Wie addiert ein Computer Zahlen? - Teil 2: Logikgatter
28 M5 - Wie addiert ein Computer Zahlen? - Teil 3: Eine arithmetisch-logische Einheit
29 M6 - Aus welchen Bauteilen besteht ein Computer?
30 M7 - Der Raspberry Pi
31 M8 - Wie entstand der Raspberry Pi?
32 M9 - Die Kommandozeile des Raspberry Pi
33 M10 - Open Source Software
34 M11 - Libre Office
35 M12 - Scratch - Der Aufbau des Programms I
36 M13 - Scratch - Der Aufbau des Programms II
37 M14 - Scratch - Eine eigene Figur entwerfen
38 M15 - Scratch - Ein Programm schreiben I
39 M16 - Scratch - Ein Programm schreiben II
40 M17 - Scratch - Ein Programm schreiben III
41 M18 - Scratch - Ein Programm schreiben IV
42 M19 - Scratch - Physical Computing mit der LEDBorg-Platine
43 M20 - Programmieren mit Minecraft I

44 *M21 - Programmieren mit Minecraft II*
45 *M22 - Programmieren mit Minecraft III*
46 *M23 - Physical Computing mit dem Raspberry Pi I*
47 *M24 - Physical Computing mit dem Raspberry Pi II*
48 *M25 - Physical Computing mit dem Raspberry Pi III*
49 *M26 - Physical Computing mit dem Raspberry Pi IV*
50 *M27 - Physical Computing mit dem Raspberry Pi V*
51 *M28 - Einen Taster an den Raspberry Pi anschließen*
52 *M29 - Den Taster programmieren*
53 *M30 - Den Taster in Minecraft verwenden I*
54 *M31 - Den Taster in Minecraft verwenden II*
55 *M32 - Den Taster in Minecraft verwenden III*
56 *M33 - Station 1: Pixel-Art*
57 *M34 - Station 1: Malvorlagen*
58 *M35 - Station 2: Makey Makey*
59 *M36 - Station 3: Einen elektronischen Würfel löten*
60 *M37 - Station 4: Bomberman-Spielstation*
61 *M38 - Station 5: Die Raspberry Pi-Bibliothek*
62 *M39 - Station 6: Einen Joystick anschließen*
63 *M40 - Station 7: Das Raspberry Pi-Kameraboard I*
64 *M41 - Station 7: Das Raspberry Pi-Kameraboard II*
65 *M42 - Station 8: Sonic Pi-Experimente I*
66 *M43 - Station 8: Sonic Pi-Experimente II*
67 *M44 - Station 9: Eine Pong-Uhr programmieren*
68 *M45 - Station 10: Ein Spiel mit Scratch programmieren I*
69 *M46 - Station 10: Ein Spiel mit Scratch programmieren II*

Was ist der Raspberry Pi?



Der Raspberry Pi, Modell B+

Für knapp 500.000 Euro hat vor wenigen Monaten ein funktionsfähiger Apple I den Besitzer gewechselt. Mit einem Computer, wie man ihn heute kennt, hat das Gerät allerdings nicht viel gemein. Er besteht lediglich aus einer **Platine mit Anschlussmöglichkeiten für Netzteil, Tastatur und Bildschirm**. Programme musste der User selbst schreiben und sogar ein einfaches Gehäuse fehlte.

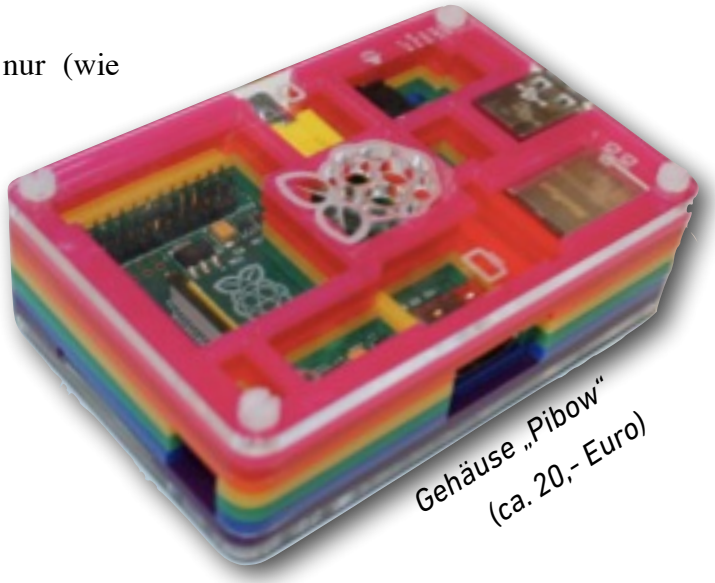
Der nur **ca. 30€ teure Raspberry Pi** ist eine Reminiszenz an diese Urform des Personalcomputers. Auch er kommt ohne Gehäuse oder Peripheriegeräte daher und wer ihn in Händen hält, stellt sich wie in den Anfangstagen des Computers die Frage, was sich mit dieser patenten Platine alles anfangen lässt.

Gehäuse „Bramble Pi" (9,- Euro)

Im Netz finden sich bereits zahlreiche Anregungen; beispielsweise verrichtet der mit Linux betriebene Mini-Computer seinen Dienst nicht nur als Mediaplayer (samt der faszinierenden Erweite-rung AmbiPi) und Jukebox, sondern auch als Wetterballon und Arcade-Maschine.

Was ist das Besondere am Raspberry Pi?

Der Raspberry Pi ermöglicht also nicht nur (wie jeder Computer) das Programmieren eigener Software, sondern macht es sehr einfach, in die **Welt des „Physical Computing"** einzusteigen, indem man eigene Projektideen mit Hilfe von computergesteuerten Sensoren, Tasten oder Leuchtdioden realisiert.


Gehäuse „Pibow" (ca. 20,- Euro)

Hierfür besitzt der Pi eine so genannten **„General Purpose Input / Output"-Schnittstelle**, mit deren Hilfe sich weitere elektronische Bauteile anschließen und steuern lassen.

Woher stammt der Raspberry Pi?

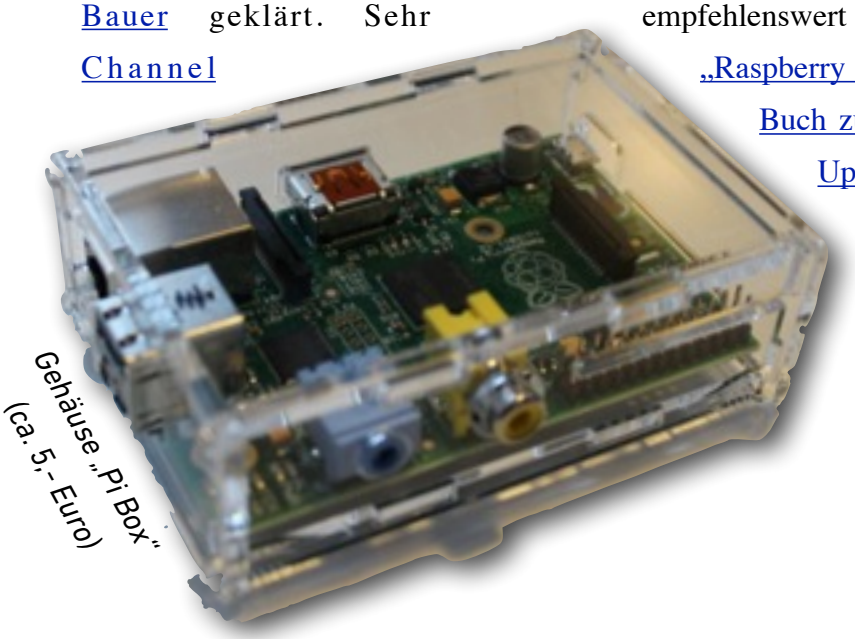
Hinter dem Mini-Computer-Projekt steckt die gemeinnützige Raspberry Pi Foundation, deren Ziel es ist, Kindern den kreativen Umgang mit Computern aufzuzeigen und zu diesem Zweck einen günstigsten, aber dennoch voll funktionsfähigen Computer zu entwickeln. Ein Treuhänder der Stiftung ist übrigens die **Programmierer-Legende David Braben**, Erfinder des Spiele-Klassikers Elite.

Viele spannende Projekte rund um den Raspberry Pi sind derzeit in der Entwicklung: Googles mobiles Betriebssystem Android wurde bereits auf dem Pi zum Laufen gebracht und auch das beliebte Spiel Minecraft gibt es in einer Version für den Raspberry Pi.

Der Raspberry Pi in der Schule

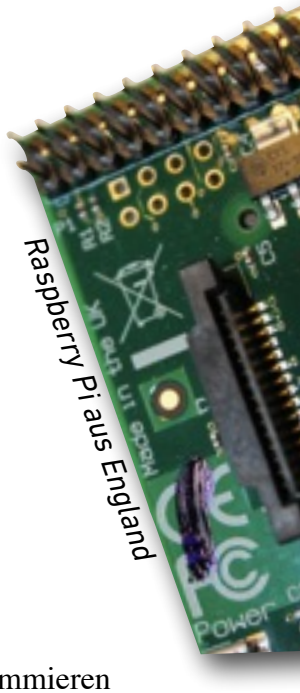
Im Netz kursieren zudem viele kostenlose Ressourcen zum Einsatz des Raspberry Pi in der Schule. In erster Linie sind hier der Newsletter „Switched On" der Initiative Computing At School (CAS) sowie die Zeitschrift „MagPi" zu nennen, von der es auch eine deutsche Ausgabe gibt.

Die meisten Fragen rund um den Aufbau des Computers und die Installation der Software werden in dem kostenlosen e-Book „Der Raspberry Pi Computer" von Peter Bauer geklärt. Sehr empfehlenswert sind auch der YouTube-Channel „Raspberry Pi IV Beginners" sowie das Buch zum Raspberry Pi von Eben Upton, einem der Gründer der Raspberry Pi Foundation.



Gehäuse „Pi Box" (ca. 5,- Euro)

Die Planung der Raspberry Pi-AG


Raspberry Pi aus England

Ende letzten Jahres wurde ich gefragt, ob ich die Computer-AG der 6. Klassen an unserer Schule übernehmen könnte. Anfangs wusste ich noch nicht ganz, was ich mit den Schülerinnen und Schülern in dieser AG anfangen sollte. Dann las ich vom Raspberry Pi und war sofort begeistert von der Idee einer offenen Hardware, die von einer gemeinnützigen Organisation für nur ca. 30,- Euro angeboten wird.

Ich beschloss daher, zehn dieser Geräte zu kaufen und ein Konzept auszuarbeiten mit dem Ziel, Kindern in einem halben Jahr die Funktionsweise eines Computers sowie Grundkenntnisse im Programmieren beizubringen. Mittlerweile bin ich der Überzeugung, dass der Raspberry Pi ein schlichtweg geniales Werkzeug ist. denn er ermöglicht einen spannenden und kreativen IT-Unterricht, der durch und durch von den Idealen der Open Source- und Open Education-Bewegung geprägt ist.

Wer hätte vor ein paar Jahren gedacht, dass man für den Preis eines einzigen Computers eine ganze Klasse mit aller Hard- und Software ausstatten kann, die man benötigt, um die Begeisterung fürs Programmieren zu wecken?

Raspberry Pi aus China

Unsere AG fand in einem Computerraum statt, so dass wir bereits genügend Arbeitsplätze mit Bildschirm, Tastatur und Maus hatten. Insgesamt fielen für die AG Kosten in Höhe von ca. 610,- Euro an. Diese Kosten setzten sich wie folgt zusammen:

- 10 Raspberry Pi	*321,29 Euro*
- 10 DVI-HDMI-Kabel	*37,60 Euro*
- 10 Netzteile	*79,95 Euro*
- 10 SD-Karten (8GB)	*59,99 Euro*
- 10 Gehäuse (Pi Box)	*48,74 Euro*
- 10 LedBorg-Platinen	*63,46 Euro*

Darüber hinaus wurden folgende Gegenstände angeschafft:

- DVD „Bibliothek der Sachgeschichten“ C3	*9,99 Euro*
- DVD „Meilensteine 9“ (mit Film „Der Transistor“)	*11,99 Euro*
- 100 Transistoren	*3,74 Euro*
- 10 DVDs zum Brennen von Ubuntu (aus 50er Box)	*ca. 3 Euro*
- MaKey MaKey-Tastaturkontroller	*53,40 Euro*

Die Vorbereitung bestand dann aus folgenden Schritten:

- *Einbau der Raspberry Pi-Computer in die Hüllen*
- *Erstellen eines „SD-Masters“[1] & Kopie auf 9 SD-Karten (Anleitung im Heft „The Mag Pi“, Ausgabe 9)*
- *Brennen der Ubuntu-DVDs*
- *10x Ausdrucken und Binden des Buches „Schlangengerangel“*
- *Ausdrucken und Laminieren der „Scratch-Cards“*
- *10x Ausdrucken und Binden der Scratch-Anleitung von www.swisseduc.ch (siehe Stunde 12)*

Im Folgenden wird versucht, eine Schritt-für-Schritt-Anleitung zum Umgang mit dem Raspberry Pi zu geben. Sollten dennoch Fragen offen bleiben, empfehle ich einen Besuch der offiziellen Website des Projekts: www.raspberrypi.org. Dort gibt es auch eine sehr engagierte deutschsprachige Community, die sehr hilfsbereit ist und Fragen in kurzer Zeit beantwortet. Natürlich können Sie auch mich unter der E-Mail-Adresse info@medienistik.de erreichen.

Zum Schluss noch ein Tipp: Der Raspberry Pi wird von manchen Anbietern zu recht hohen Preisen verkauft. Man sollte daher das Gerät nur von einem der offiziellen Verkäufer erwerben, auch wenn dieser manchmal etwas längere Lieferzeiten hat. Alle Anbieter sowie weitere Informationen zum Kauf findet man ebenfalls auf der Seite www.raspberrypi.org.

---

[1] Das Betriebssystem Raspbian „wheezy“ ist kostenlos erhältlich unter: http://www.raspberrypi.org/downloads. Zusätzlich installiert wurden die Programme „Libre Office“ (erhältlich im vorinstallierten Pi Store), Minecraft (erhältlich unter: http://pi.minecraft.net) und die Scratch GPIO-Erweiterung (Beschrieben in der Zeitschrift „The Mag Pi“, Ausgabe 9 & 10). Die Installation enthält einige Fallstricke, aber die deutschsprachige Community auf www.raspeberrypi.org hat viele Probleme bereits gelöst & dokumentiert und bietet schnelle Hilfe bei Problemen.

1.-3. Stunde: Wie funktioniert ein Computer?

Apple II von 1982

Kinder und Jugendliche wachsen heute in einer Welt auf, die wie nie zuvor von Technik und Computern geprägt ist. Diese wurde im Laufe der Jahrzehnte immer bedienerfreundlicher, so dass sich heute niemand mehr mit Kommandozeilen beschäftigen muss, um eine Datei aufzurufen oder einen Text zu bearbeiten. So erfreulich diese Entwicklung einerseits ist, führt sie andererseits jedoch auch dazu, dass sich immer weniger Menschen dafür interessieren, wie der (für den Laien gar nicht mehr zu öffnende) Tablet-Computer auf dem Wohnzimmertisch innen aussieht und aus welchen Bauteilen er besteht.

iPhone 3G - Platine & Gehäuse („selbstgebautes" Anschauungsobjekt)

Da unser Schulmuseum glücklicherweise über eine kleine IT-Abteilung verfügt, lag es nahe, dass unsere AG mit einer kleinen Zeitreise in die Welt vor Google und iPhone beginnt; in eine Welt, in der „Computer" noch mechanisch funktionierten und auf das Addieren von Zahlen beschränkt blieben.

Die ersten drei Stunden beschäftigen sich daher zunächst mit den technischen Grundlagen. Diese sind zwar keine unbedingt notwendige Voraussetzung für das Programmieren und den

Umgang mit dem Raspberry Pi, aber im Umgang mit einem Computer auf einer Platine stellt man sich fast zwangsläufig die Frage, wie so eine Maschine überhaupt funktioniert.

Sehr bei der Vorbereitung geholfen haben mir übrigens die folgenden beiden YouTube-Videos: „How computers work in less than 20 minutes“ von EngineeringIsMagic (http://www.youtube.com/watch?v=WIDzNyfVVg0) und „See How Computers Add Numbers In One Lesson von InOneLesson (http://www.youtube.com/watch?v=VBDoT8o4q00).

1. Stunde: Der Comptometer

Tastatur des Comptometer

Die erste Station unserer Reise war der Comptometer - eine Rechenmaschine, die bereits im 19. Jahrhundert entwickelt wurde und heute weitgehend in Vergessenheit geraten ist. In den „IBM Archives“ wird das Gerät wie folgt beschrieben:

> *Dorr E. Felt (1862-1930) achieved the technological breakthrough of a key-driven adding machine. The Comptometer's key set the number to be added or subtracted, and provided the power for performing the operation. Both fast and reliable, the machine was used in both business and science. It not only added and subtracted but also performed multiplication and division.*[2]

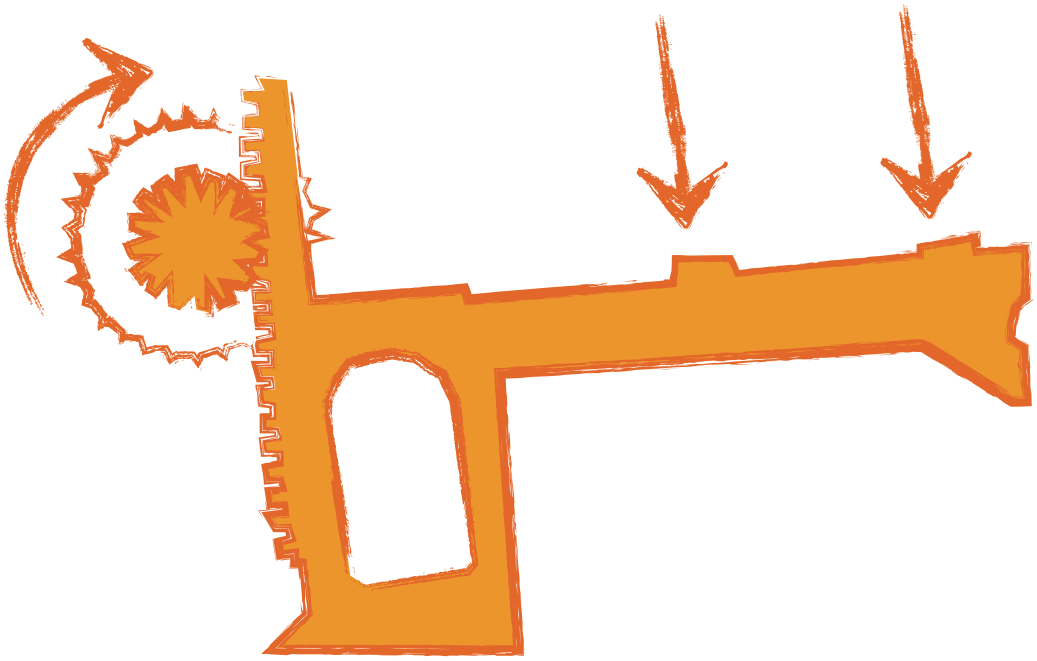
Das Modell aus unserem Schulmuseum kam kurz vor dem zweiten Weltkrieg auf den Markt. Es beherrscht im Gegensatz zu den allerersten Modellen nicht nur die Addition von Zahlen, sondern auch deren Subtraktion, Multiplikation und Division. Auch wenn die Bedienung recht kompliziert ist (hier findet sich ein schönes Anschauungsvideo: http://www.youtube.com/watch?v=2y6feYyBjxo), ist der Comptometer aufgrund seiner einfachen mechanischen Funktionsweise ein anschaulicher Einstieg in die Funktionsweise von Computern.

[2] http://www-03.ibm.com/ibm/history/exhibits/attic2/attic2_047.html

M1 Lösungen:

1 Die Funktionsweise des Comptometers wird anschaulich in folgendem Video erklärt: http://www.youtube.com/watch?v=SbJpufimfdM

Alternativ lässt sich auch die folgende Grafik verwenden. Durch das Drücken einer Taste bewegt sich eine Zahnstange nach unten, die ein Zahnrad in Bewegung setzt. Je weiter oben sich die Taste befindet, desto tiefer bewegt sich der Hebel und desto weiter dreht sich das Zahnrad, das wiederum die Zahl auf der Anzeige hochsetzt.



2 Zum Lösen dieser Aufgabe muss die Rechnung in eine Additionsaufgabe umgewandelt werden, d. h. man rechnet zunächst. 12+12+12 (=36) und anschließend 120+120 (=240). Auf dem Comptometer muss man dazu folgende Schritte ausführen:

Zunächst müssen diese beiden Tasten („1“ und „2“) gleichzeitig drei Mal hintereinander gedrückt werden.

Anschließend rutscht man mit beiden Fingern eine Spalte nach links und drückt diese beiden Tasten zwei Mal.

3 Der größte Unterschied zwischen Computer und Comptometer besteht darin, dass der Comptometer nur eine sehr geringe Speichermöglichkeit hat. Er kann genau eine Zahl speichern und Rechenoperationen lassen sich nicht rückgängig machen. Schülerinnen und Schülern wird außerdem auffallen, dass der Comptometer rein mechanisch funktioniert und es in einem Computer kaum bewegliche Teile (abgesehen vom Lüfter oder der Festplatte) gibt. Dies bereitet auf die nächste Stunde vor, in der es um die Erfindung des Transistors geht.

Wer sich genauer mit dem Comptometer auseinander setzen möchte, findet ein interessantes Erklärvideo des Programmierers Brooke Boering hier: http://www.youtube.com/watch?v=aDDAbKURGZI

2. Stunde: Transistoren und Binärcode

In der nächsten Stunde haben wir uns damit beschäftigt, welche Erfindung es eigentlich genau war, die das Ende des mechanischen Computers eingeläutet hat. Die Rede ist natürlich vom Transistor. Nur wenn man dessen grundlegenden Aufbau versteht, kann man verstehen, wie Computer funktionieren. Als Verstehenshilfe diente das Arbeitsblatt M2, auf dem die Funktionsweise des Transistors kurz erklärt wird.
Die Stunde begann mit der Frage nach der wichtigsten Erfindung des 20. Jahrhunderts. Die Antwort ist natürlich umstritten, aber es spricht einiges dafür, dass dem Transistor diese Ehre zuteil wird. Als Anschauungsobjekt habe ich einige Transistoren gekauft (Packungen mit hundert Transistoren kosten ca. 2,- Euro). Nachdem wir M2 gemeinsam gelesen und besprochen haben, kam der Film „Meilensteine der Menschheit 9 - Der Transistor“ zum Einsatz, der auf DVD erhältlich ist (gezeigt wurden die letzten 5 Minuten). Zusätzlich wurde das kurze Intel-Werbevideo „The Power of Smaller“ gezeigt (http://www.youtube.com/watch?v=EtcpueRBMA4), in dem die schier unfassbaren Fortschritte der Transistorentwicklung deutlich gemacht werden.

Anschließend wurde noch der Binärcode eingeführt. Da Transistoren keine Zahlen speichern, sondern sich nur ein- oder ausschalten lassen, benötigt man ein Zahlensystem, das nur aus Einsen und Nullen besteht. Als Einstieg kann man die Schülerinnen und Schüler alle dreistelligen Zahlen aufschreiben lassen, die nur aus Nullen und Einsen bestehen. Anschließend wird auf dem Arbeitsblatt M3 erklärt, wie sich mit Binärzahlen die Ziffern von 0 bis 7 darstellen lassen und wie man diese addiert. Dieser Schritt ist eine wichtige Voraussetzung für die nächste Stunde, in der erläutert wird, wie Computer einfache Rechenoperationen ausführen. Die beiden Arbeitsanregungen können zu Hause erledigt werden.

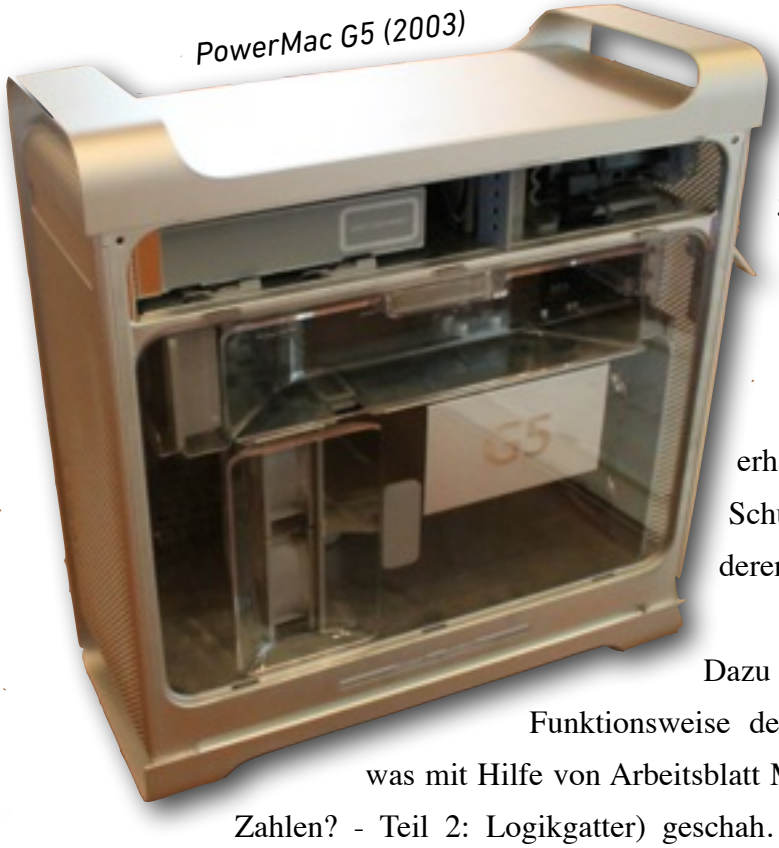
M2 Lösung

1 Gemeint ist das berühmte *Silicon Valley* (Silizium-Tal). Selbst in der ersten Auflage der Steve Jobs-Biographie wurde engl. *silicon* übrigens fälschlicherweise mit *Silikon* übersetzt. So wurde das Silicon Valley beschrieben als Ort, „in dem Silikon in Gold verwandelt wird“. Der Name spiegelt die Bedeutung des Transistors wider, der aus Silizium hergestellt wird.

M3 Lösung

1 Ein Gigabyte sind 1 Milliarde Byte (je nach Kontext, in dem das Wort gebraucht wird, sind manchmal jedoch auch 1.073.741.824 Bytes gemeint. Genaueres dazu lässt sich hier nachlesen: http://de.wikipedia.org/wiki/Byte.

3. Stunde: Logikgatter / Bestandteile eines Computers



In der letzten Theoriestunde wurde mit Hilfe der Arbeitsblätter M4 und M5 gezeigt, wie genau Computer mit Hilfe von Transistoren Zahlen addieren. Auch wenn damit noch nicht erklärt wird, wie Prozessoren funktionieren, erhalten die Schülerinnen und Schüler einen ersten Eindruck von deren Funktionsweise.

Dazu müssen aber noch die Funktionsweise der Logikgatter erklärt werden, was mit Hilfe von Arbeitsblatt M4 - Wie addiert ein Computer Zahlen? - Teil 2: Logikgatter) geschah. Nachdem die relativ simplen Schaltungen erklärt wurden, wurde mit Hilfe des Arbeitsblattes weiter erläutert, wie ein Computer mit Hilfe von Logikgattern Zahlen addieren kann.

Als Abschluss der Theorieeinheit kam ein Ausschnitt aus einem Film der „Bibliothek der Sachgeschichten“ zum Einsatz, in dem kindgerecht die Funktionsweise eines

Computers beschrieben wird. Zur Sicherung der wichtigsten Begriffe wurde das Arbeitsblatt M6 eingesetzt.
Der Film ist erhältlich auf der DVD „Bibliothek der Sachgeschichten - (C3) Computertastatur, Computer, Handy (Zum Einsatz kam der Film „Computer“). Zu Beginn der DVD befindet sich übrigens eine Warnung, dass der Einsatz in der Schule einer besonderen Lizenz bedarf. Dies ist jedoch rechtlich sehr umstritten. Wer sich näher mit der Rechtslage bezüglich der Filmschau im Unterricht beschäftigen möchte, findet dazu einen Artikel in meinem Blog: http://medienistik.wordpress.com/2012/02/24/durfen-im-handel-erworbene-dvds-im-unterricht-gezeigt-werden/

M6 Lösung

1 Ein Computer besteht aus **Hardware** und **Software**. Ein Teil eines Computers ist die CPU, man nennt sie auch **Prozessor**. Sie besteht aus Milliarden kleiner Transistoren. Schaltet man den Computer ein, wird zunächst das **BIOS** aufgerufen, das die angeschlossene Hardware überprüft, z. B. die **Tastatur**, **Grafikkarte** oder **Festplatte**. Es ist in einem Speicher abgelegt, den man nur lesen, aber nicht beschreiben kann. Einen solchen Speicher nennt man **ROM**.

4.-5. Stunde: Der Raspberry Pi

Nachdem die Schülerinnen und Schüler einen für eine AG ungewöhnlich langen Theorieteil von 3 Stunden über sich ergehen lassen mussten, folgte nun die Präsentation des Raspberry Pi, der fortan der einzige Rechner war, mit dem wir gearbeitet haben.

4. Stunde: Präsentation des Raspberry Pi

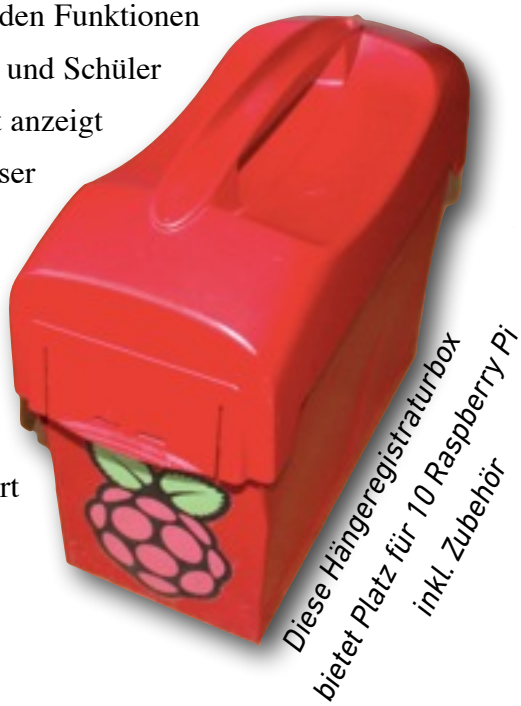
In der ersten Praxis-Stunde wurden die Schülerinnen und Schüler in Zweierteams aufgeteilt. Diese Teams wurden zu Beginn jeder Stunde neu ausgelost. Hierzu bekam jede Schülerin und jeder Schüler eine Karte mit dem Teil eines Wortes und musste dann die jeweilige Partnerin / den jeweiligen Partner finden.
Zu Beginn der Stunde stellte ich den Raspberry Pi kurz vor. Anschließend wurden die Gruppen ausgelost und jede Gruppe bekam einen Computer und jedes Gruppenmitglied das Arbeitsblatt M7, das zeigt, aus welchen Bausteinen der Raspberry Pi besteht. Auch

die Idee, die zur Erfindung des Raspberry Pi geführt hat, wurde mit Hilfe von M8 besprochen.

Danach schloss jede Gruppe ihren Raspberry Pi an Monitor, Maus und Tastatur an. Sobald die SD-Karte eingesetzt und das Netzteil angesteckt wird, startet der Raspberry Pi - er besitzt keinen Ein- oder Ausschalter. Falls beim ersten Start Probleme auftauchen, sollte versucht werden, Maus und Tastatur erst später anzuschließen. Bei uns hat diese Vorgehensweise jedenfalls das Problem, dass der Raspberry Pi nicht richtig startete, behoben.

Zu Beginn der AG wurde absichtlich noch nichts von den Funktionen erzählt, die der Raspberry Pi besitzt. Die Schülerinnen und Schüler sollten nur sehen, dass der Computer eine Menge Text anzeigt und dann nach einem Benutzernamen fragt. Auch dieser wurde anfangs absichtlich noch nicht verraten.

Stattdessen besprachen wir gemeinsam das Arbeitsblatt M8, auf dem erklärt wird, wie der Raspberry Pi entstanden ist. Die Stunde endete mit dem Abbau der Geräte und dem Verweis darauf, dass in der nächsten Stunde die Funktionen genauer erläutert werden.



Diese Hängeregistraturbox bietet Platz für 10 Raspberry Pi inkl. Zubehör

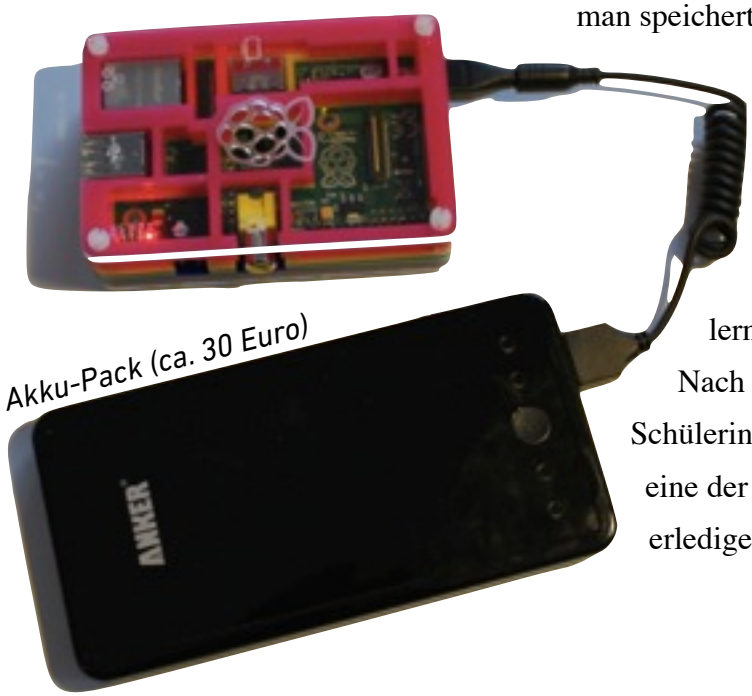
5. Stunde: Die Kommandozeile / Open Source Software / Libre Office

Diese Stunde begann damit, dass ich einen alten Apple II-Computer an unser digitales Whiteboard angeschlossen habe. nach dem Booten zeigt dieser - wie jeder „klassische“ Computer - eine Kommandozeile an, mit deren Hilfe sich z. B. Dateien anzeigen, kopieren und ausführen lassen. Zur Demonstration wurden einige Programme dieser simplen Programme von einem Schüler ausgeführt, z. B. eine sehr einfache textbasierte Adressenverwaltung.

Box zur Aufbewahrung der SD-Karten

Anschließend wurde das Arbeitsblatt M9 ausgeteilt, auf dem einige Kommandos aufgelistet sind, die die Schülerinnen und Schüler ausprobieren können. Als Aufgabe soll das Datum

eingestellt werden, wozu auf dem Arbeitsblatt auch der Begriff „sudo“ erläutert wird. Anschließend folgte mit Hilfe von M10 eine Einführung in die Open Source Software. Am Ende der Stunde wurden die zuvor gebrannten Ubuntu-DVDs ausgeteilt mit dem Auftrag, das Betriebssystem (und die installierten Office-Programme) am heimischen PC einmal auszuprobieren. Dabei werden übrigens keine Einstellungen oder Daten des auf dem PC installierten Betriebssystems geändert. Alle Änderungen, die man vornimmt und Dateien, die man erstellt, sind nach dem Ausschalten des PCs verloren, es sei denn, man speichert sie auf einem USB-Stick.



Am Ende ging es darum, die Möglichkeiten des Raspberry Pi durch die Verwendung von „Libre Office“ (früher bekannt als „Open Office“) etwas genauer kennen zu lernen.[3]

Nach einer kurzen Einführung sollten die Schülerinnen und Schüler in Zweiergruppen eine der Aufgaben auf dem Arbeitsblatt M11 erledigen.

6.-10. Stunde: Programmieren mit Scratch

Bei Scratch handelt es sich um eine sehr leicht zu erlernende Programmiersprache, die speziell vom berühmten Massachusetts Institute of Technology für Kinder entwickelt wurde und - so wie Python - bereits auf der „wheezy“-Distribution installiert ist. Zu „Scratch“ gibt es viele Veröffentlichungen, sowohl im Internet als auch in der Buchhandlung, diese sind jedoch leider zumeist in englischer Sprache, z. B. das sehr empfehlenswerte Buch „Super Scratch Programming Adventure!“.

[3] LibreOffice lässt sich über den PiStore kostenlos installieren. Möchte man das Programm auf Deutsch nutzen, benötigt man jedoch zusätzlich das Paket „libreoffice-l10n-de“ (Befehl: sudo apt-get install libreoffice-l10n-de). Danach lässt sich unter dem Menü „Tools“, dann „Options“ und „Language Settings“ die Sprache auf Deutsch umstellen.

6. Stunde: Einführung in Scratch mit „Scratch Cards“

Ein guter (wenn auch sehr rudimentärer) deutschsprachiger Einstieg in Scratch stellen die kostenlosen „Scratch Cards“ dar, die man herunterladen und in ausreichender Anzahl ausdrucken kann. Sie sind erhältlich unter: http://www.brandhofer.cc/?p=65.
In der ersten Stunden können die Schülerinnen und Schüler mit diesen Karten die Funktionsweise des Programms näher kennen lernen. Die Reihenfolge, in der die Karten bearbeitet werden, ist dabei nicht wichtig.

7.-9. Stunde: Partnerarbeit mit Scratch

Auf der Seite swisseduc.ch (http://www.swisseduc.ch/informatik/programmiersprachen/scratch_werkstatt/) wird ein sehr umfangreiches und hervorragend aufbereitetes deutschsprachiges Tutorial zu Scratch kostenlos angeboten, das sich gut zum Selbstlernen in Zweiergruppen eignet. Es wird sogar eine Vorlage für einen „Scratch Pass“ angeboten, auf dem die Schülerinnen und Schüler ihre Erfolge dokumentieren können. Mit diesen Materialien lassen sich auch mehrere Stunde füllen, falls die AG länger als ein Halbjahr dauern soll. Weitere Anregungen für Scratch-Projekte finden sich übrigens auch in der monatlich erscheinenden Zeitschrift „The Mag Pi“, die ebenfalls kostenlos erhältlich ist.
Die Arbeitsblätter M12-M18 geben ebenfalls einen Überblick über die wichtigsten Funktionen des Programms und erklären, wie man ein einfaches Programm mit Scratch schreibt.

10. Stunde: Physical Computing mit der LedBorg-Platine

Eine AG, die sich mit Raspberry Pi beschäftigt, sollte nicht zu Ende gehen, ohne dass die Schülerinnen und Schüler zumindest ein kleines Projekt mit der GPIO-Schnittstelle realisiert haben. Über diese Schnittstelle lassen sich zahlreiche elektronische Bauteile anschließen und steuern. Das Problem ist jedoch, dass man mit einer falschen Verkabelung den Raspberry Pi leicht zerstören kann. Komplexere Projekte sollten daher sorgfältig vorbereitet werden. Eine Zusatzhardware, mit der auch Kinder nichts falsch

machen können, ist die „LedBorg-Platine“, die sich auf der Seite www.piborg.org für 5 Britische Pfund das Stück bestellen lässt. Diese Platine ist mit einer LED ausgestattet, die die Farbe wechseln kann und interessante Projekte ermöglicht.
In „The Mag Pi“, Ausgabe 10 ist erklärt, wie man diese Platine mit der Programmiersprache „Scratch“ ansteuern lässt. Den dort abgedruckten Code kann man als Vorlage für alle Schülerinnen und Schüler kopieren.
Wer noch mehr basteln möchte, sollte einen Blick auf die für die Arbeit mit Kindern sehr interessante (wenn auch mit ca. 50,- Euro leider etwas teure) Hardware „MaKey MaKey“ werfen, mit der sich jedes beliebige leitende Objekt, z. B. ein Stück Obst, in eine Taste verwandeln lässt. Auf www.makeymakey.com kann man sich einige verrückte Projekte anschauen, die mit der Hardware realisiert wurden.
Darüber hinaus gibt es mittlerweile einen großen Markt von Anbietern, die spezielle Hardwareerweiterungen für den Raspberry Pi anbieten - einer der größten ist www.adafruit.com.

11.- 13. Stunde: Programmieren mit Python

Heutzutage lernen die wenigsten Schülerinnen und Schüler, wie man selbst ein Programm am Computer schreibt. Den Grund dafür, warum das wichtig ist, kann man u. a. auf www.code.org erfahren. Die Einführung in die Programmierung mit Python erfolgte mit Hilfe des kostenlosen eBooks „Schlangengerangel für Kinder - Programmieren lernen mit Python“ von Jason R. Briggs (übersetzt von Joe Ehrensberger). Das Buch (das unter der freien Creative Commons Lizenz herausgegeben wurde) wurde dazu im Vorfeld zehnmal ausgedruckt und gebunden. Es kann unter folgender Adresse heruntergeladen werden: http://code.google.com/p/swfk-de/downloads/list („Linux Ausgabe“ auswählen).
So hatten die Schülerinnen und Schüler drei Stunden lang die Möglichkeit, sich in Zweiergruppen im eigenen Tempo mit der Python-Programmierung auseinander-zusetzen. Das Buch ist glücklicherweise in einer für Kinder verständlichen Sprache geschrieben und eignet sich daher hervorragend zum Selbststudium.

14.- 19. Stunde: Programmieren mit Minecraft

Eine der „Killer-Applications“ des Raspberry Pi ist das beliebte Spiel Minecraft. Die Version für den Raspberry Pi bietet nicht alle Möglichkeiten der PC-Version, ist dafür jedoch kostenlos. Ein Alleinstellungsmerkmal der Version für den Raspberry Pi ist, dass der Spieler die Möglichkeit hat, mit Hilfe von Befehlen in der Python-Konsole direkt in die Spielwelt einzugreifen.
Alles zur Einrichtung von Minecraft auf dem Raspberry Pi findet man auf der Seite https://mcpipy.wordpress.com. Alle Programme, die auf den Arbeitsblättern erwähnt werden, finden sich auf der Seite: https://github.com/brooksc/mcpipy. Bei Fragen zum (nicht ganz leichten, aber absolut lohnenswerten) Einrichten von Minecraft auf dem Raspberry Pi und dem Starten der Programme, melden Sie sich am besten per E-Mail (info@medienistik.de) bei mir.

14. Stunde: Python & Minecraft I

Mit Hilfe von M12 und M13 wird erklärt, wie sich das Spiel starten lässt und wie man sich mit Python in das Spiel „einhackt“. Auf dem Arbeitsblatt befinden sich zudem einige Befehle, die es ermöglichen, Blöcke in die Welt zu setzen. Das sollte ausreichen, um die Schülerinnen und Schüler für die erste Stunde zu beschäftigen.

15. Stunde: Python & Minecraft II

In der zweiten Stunde können die Schülerinnen und Schüler mit Hilfe von M14 komplexere Minecraft-Programme, die in Python geschrieben wurden, ausprobieren. Diese wurden von den jeweiligen Programmierern im Internet veröffentlicht und lassen sich dort kostenlos herunterladen. In der zweiten Stunde können die Schülerinnen und Schüler ein wenig mit den Programmen herumexperimentieren und ggf. den Quellcode anschauen und verändern, um zu sehen, welchen Effekt die Änderungen auf das Programm haben. So lassen sich die bereits gewonnen Erkenntnisse in der Python-Programmierung noch vertiefen.

16.-19. Stunde: Physical Computing mit LEDs, Taster und Minecraft

Nachdem die Schülerinnen und Schüler Erfahrungen mit Scratch und der LED-Borg-Platine gesammelt haben, bestand der nächste Schritt darin, „echte“ elektronische Bauteile wie LEDs und Taster an den Raspberry Pi anzuschließen.
Die dafür benötigten Bauteile kosten nur wenige Euro. Benötigt werden pro Schüler:

- *1 Steckbrett (es reicht eine sehr einfache Version, die es ab ca. 1,- Euro gibt)*
- *1 rote LED (5mm)*
- *1 gelbe LED (5mm)*
- *1 grüne LED (5mm)*
- *6 flexible Drahtbrücken (auch „Jumper-Kabel“ genannt) Buchse-Buchse*
- *3 flexible Drahtbrücken Stecker-Buchse*
- *1 Kurzhubtaster*
- *3 Widerstände 1/4 W, 220 Ohm*
- *1 Widerstand 1/4 W, 10 kOhm*

Die Arbeitsblätter M15-M19 zeigen Schritt für Schritt, wie man zuerst eine und dann drei LED an den Raspberry Pi anschließt und programmiert.
Noch einen Schritt weiter gehen die Aufgaben auf den Arbeitsblättern M20-M21. Hier lernen die Schülerinnen und Schüler, wie man einen Taster an den Raspberry Pi anschließt und programmiert.
Der hierfür erstellte Programmcode wird anschließend (siehe Arbeitsblätter M22-M24) dazu verwendet, Aktionen im Spiel „Minecraft“ auszulösen. Mit Hilfe der Arbeitsblätter lernen die Schülerinnen und Schüler die wichtigsten Befehle für das „Hacken“ des Spiels kennen und werden dadurch in die Lage versetzt, eigene kleine Programme in Minecraft zu schreiben, die durch einen Knopfdruck ausgelöst werden.
So kann die verbleibende Zeit dazu genutzt werden, eigene Projekte mit dem Raspberry Pi zu entwickeln und auszuprobieren und vielleicht beschäftigen sich die Schülerinnen und Schüler auch über die AG hinaus mit Python. Ein Motivationsanreiz kann es sein, ihnen mitzuteilen, dass auch Firmen wie YouTube und Google auf Python setzen. Aber selbst wenn sie „nur“ gelernt haben, fortan Computer und Software mit anderen Augen zu sehen, war die AG schon ein voller Erfolg.

20. - 30. Stunde: Eigenständiges Stationenlernen

Zum Abschluss der AG hatten die Schülerinnen und Schüler Zeit, eigenständig an zehn verschiedenen Stationen zu arbeiten, die am Ende dieses Themenheftes dokumentiert sind.

Zu diesem Zeitpunkt haben sich auch schon einige interessierte Schüler gewünscht, sich intensiver mit einer Programmiersprache beschäftigen zu dürfen, was natürlich ebenfalls möglich war. Hierzu eignet sich besonders die Arbeit mit dem kostenlosen eBook „Schlangengerangel für Kinder“. Einige haben auch kleine Spiele in Scratch aus dem (leider nur auf Englisch erhältlichen) Buch „Super Scratch Programming Adventure“ nachprogrammiert.

Folgende Hardware wurde für das Stationenlernen zur Verfügung gestellt:

- *1 Makey Makey (Station 2)*
- *Bausätze für einen elektronischen Würfel (Station 3)*
- *1 selbst gebaute Bomberman-Spielestation (Station 4)*
- *gesammelte Literatur / Zeitschriften zum Raspberry Pi (Station 5)*
- *1 Joystick mit zwei Arcade-Buttons in einem selbst gebauten Gehäuse aus LEGO™ (Station 6)*
- *1 Raspberry Pi Kameraboard (Station 7)*

M1 - Der Comptometer

Eine Rechenmaschine aus dem 19. Jahrhundert

Ein Comptometer aus dem Jahr 1939

Der Comptometer wurde 1887 von Dorr E. Felt entwickelt. Er ermöglichte es, Zahlen in nie zuvor gekannter Geschwindigkeit zu addieren oder subtrahieren.

Es gab früher sogar eigene Schulen, in denen man lernen konnte, mit dem Comptometer umzugehen, um als „Comptometrist“ zu arbeiten.

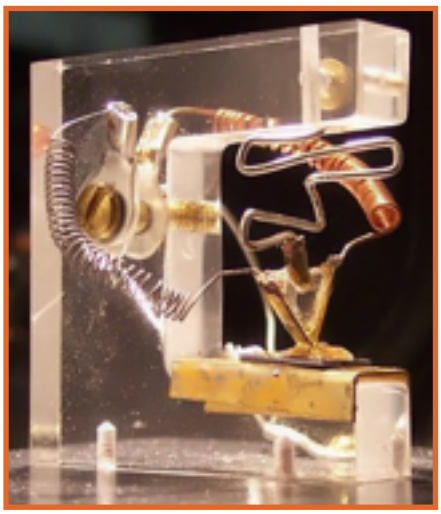
Arbeitsanregungen

1 Überlege gemeinsam mit deiner Partnerin / deinem Partner, wie der Comptometer funktionieren könnte und erstelle eine kleine Skizze.

2 Mit dem Comptometer konnte man nicht nur addieren und subtrahieren, sondern auch multiplizieren und dividieren. Überlegt gemeinsam, wie ihr die Rechenaufgabe 12x23 mit dem Comptometer lösen könnt.

3 Schau dir den Comptometer an. Obwohl die erste Version dieses Gerätes bereits weit über 100 Jahre alt ist, besitzt es doch Ähnlichkeiten mit einem Computer. Überlege zusammen mit deiner Partnerin / deinem Partner, welche Gemeinsamkeiten und Unterschiede euch auffallen.

M2 - Der Transistor



Bildquelle: dt. Wikipedia

Nachbau des ersten Transistors aus dem Jahr 1947

Die mechanische Funktionsweise des Compto-meters hat sich nicht durchgesetzt. Würden Computer heute mechanisch funktionieren, wären sie nicht nur riesengroß und laut, sondern auch sehr fehleranfällig.

Die Entwicklung des modernen Computers wurde überhaupt erst möglich durch die Erfindung des Transistors, der für viele Fachleute als bahnbrechendste Erfindung der letzten 100 Jahre gilt.

Ein Transistor besteht aus Silizium und ist ziemlich kompliziert aufgebaut. Die Funktionsweise eines Transistors ist jedoch leicht zu verstehen: er kann ein elektronisches Signal verstärken oder es ein- und ausschalten. Im Gegensatz zu einem Schalter besitzt ein Transistor jedoch keine beweglichen Teile, er wird nur durch elektrische Spannung ein- und ausgeschaltet.

Kleiner und kleiner und kl...

Noch heute gilt *Moore`s Law* (das mooresche Gesetz), das besagt, dass sich die Größe von Transistoren alle 24 Monate halbiert. 1965 passten nur 60 Transistoren auf einem Chip - man konnte sie einfach mit der Hand aufsetzen; Heutzutage passen auf die Fläche des Punktes am Ende dieses Satzes ca. 4 Millionen Transistoren, wobei sich jeder 300 Milliarden mal in der Sekunde ein- und ausschalten lässt. Würde man einen Lichtschalter so oft an- und ausschalten, wäre man 4.000 Jahre beschäftigt.

Arbeitsanregung

1 Unternehmen wie Intel, Google, eBay, Apple, Amazon und Facebook haben ihren Firmensitz in einer kleinen Region in der Nähe von San Francisco. Wie heißt diese Region und was bedeutet der Name?

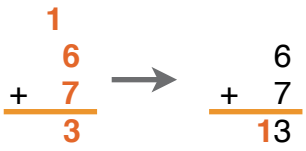
M3 - Wie addiert ein Computer Zahlen? - Teil 1: Der Binärcode

Transistoren lassen sich an- oder ausschalten. Daher müssen alle Informationen, die ein Computer verarbeitet, in Nullen und Einsen gespeichert werden. Ein Computer braucht daher eine eigene Sprache, in der alle Informationen nur mit diesen beiden Ziffern dargestellt werden.

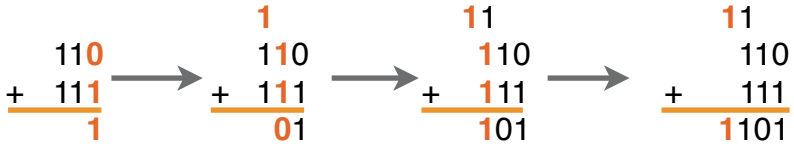
Zu diesem Zweck wurde der Binärcode entwickelt. Eine Informationseinheit, die entweder aus „0“ oder „1“ besteht, nennt man ein Bit[4]. Mit 3 Bits lassen sich beispielsweise die Ziffern von 0-7 darstellen[5]:

Ziffer	0	1	2	3	4	5	6	7
Binärwert	0	1	10	11	100	101	110	111

Binärwerte lassen sich genau so addieren wie Dezimalzahlen. Möchte man 6 plus 7 rechnen[6], geht das mit dieser Rechnung (*S* steht für *Summe*, *Ü* für *Übertrag*):



Auf die gleiche Weise kann man auch im Binärsystem addieren:



Das Ergebnis lautet also „1101“, was im Binärcode für „13“ steht.

Arbeitsanregung

1 Wie viele Bytes passen auf einen USB-Stick mit 1 Gigabyte?

Es sind _____ Bytes.

---

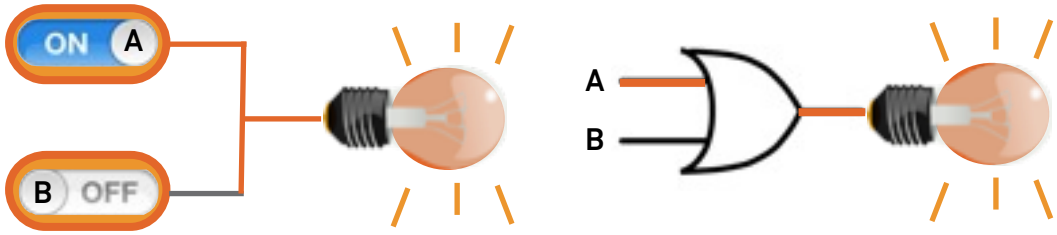
[4] *8 Bit hintereinander nennt man übrigens ein Byte.*

[5] *Dieses Beispiel stammt von der Seite: http://de.kioskea.net/contents/base/binaire.php3*

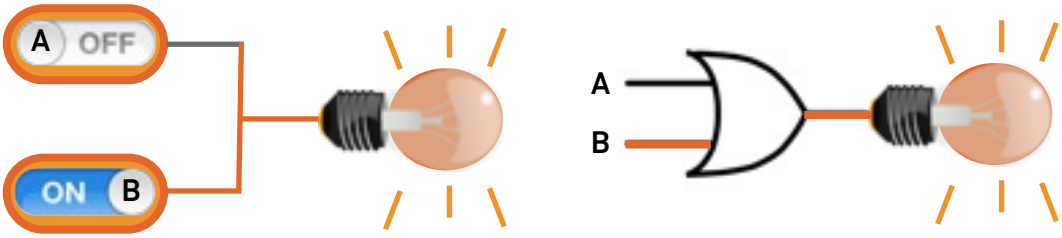
[6] *Diese und weitere Rechnungen werden erklärt auf der Seite: http://improve.dk/archive/2012/02/06/adding-67-at-the-logic-gate-level.aspx*

M4 - Wie addiert ein Computer Zahlen? - Teil 2: Logikgatter

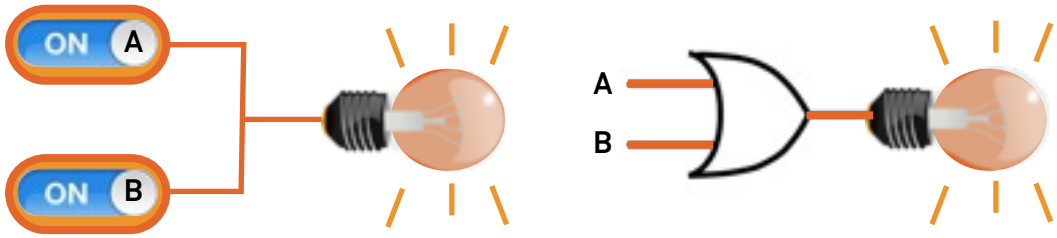
Das praktische an Transistoren ist, dass man mit ihnen logische Verknüpfungen bilden kann. Das folgende Beispiel zeigt Transistor A und Transistor B, die durch ein so genanntes *ODER-Gatter* miteinander verbunden sind. Schaltet man Transistor A ein, leuchtet die Lampe. Rechts daneben siehst du das Symbol für ein *ODER-Gatter*.



Die Lampe leuchtet auch, wenn ich nicht Transistor A, sondern Transistor B einschalte.

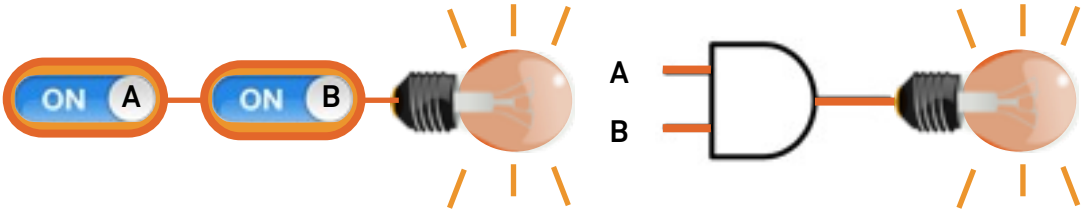


Und sie leuchtet natürlich auch, wenn ich beide Transistoren einschalte.



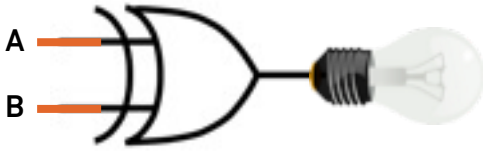
Eine andere Verknüpfung ist das *UND-Gatter*. Wenn man nur Transistor A anschaltet, erreicht der Strom die Lampe nicht, sie bleibt aus. Wie du siehst, wird das *UND-Gatter* auch durch ein anderes Symbol dargestellt.

Nur wenn beide Transistoren angeschaltet sind, brennt die Lampe.

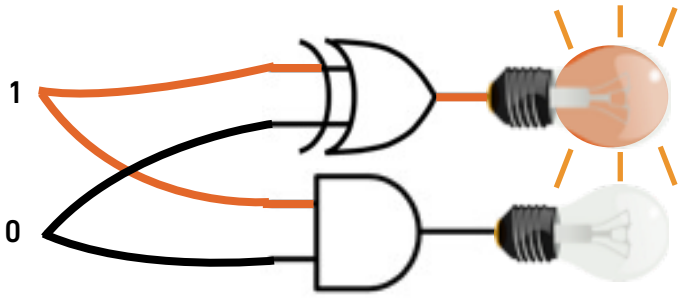


M5 - Wie addiert ein Computer Zahlen? - Teil 3: Eine arithmetisch-logische Einheit

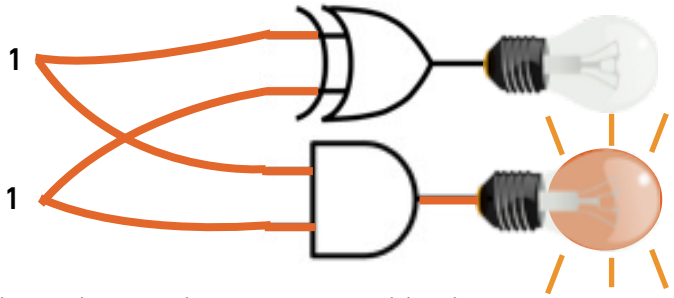
Nun können wir uns mit Hilfe von Transistoren eine einfache arithmetisch-logische Einheit (engl. arithmetic logic unit, kurz: ALU) bauen, die die Addition der Zahlen übernimmt, so dass wir das nicht mehr im Kopf machen müssen.
Dazu benötigen wir noch ein weiteres Logikgatter, nämlich das *Exklusiv-ODER-Gatter* (kurz: *XOR-Gatter*). Es funktioniert wie ein *ODER-Gatter* mit der Ausnahme, dass die Lampe nicht brennt, wenn beide Transistoren eingeschaltet sind. Das Zeichen für so ein *XOR-Gatter* sieht so aus:



Nun können wir einen kleinen Prozessor bauen, der zwei Binärzahlen addiert. Addiert man die Binärzahlen 1 und 0 kommt in der folgenden Schaltung oben als Summe 1 und unten als Übertrag 0 heraus.



Addiert man nun 1 und 1, kommt oben als Summe 0 heraus und unten als Übertrag 1.



Und schon hat man einen Rechner, der 1 und 1 zusammenzählen kann!

M6 - Aus welchen Bauteilen besteht ein Computer?

Ein Computer erfüllt heutzutage unzählige Aufgaben, aber im Prinzip kann er eigentlich nur vier verschiedene Dinge erledigen:

1.) Logische und arithmetische Operationen ausführen (z. B. Zahlen addieren).
2.) Daten speichern und auslesen (z. B. auf einer Festplatte).
3.) Ein Programm ausführen (z. B. ein Betriebssystem wie Windows).
4.) Angeschlossene Geräte steuern (z. B. eine Maus oder einen Drucker).

Dabei kennt ein Computer eigentlich nur zwei Zustände: „Strom an“ und „Strom aus“. Damit aus diesen beiden Zuständen Zahlen und Buchstaben werden, benutzt der Computer einen eigenen Code, den so genannten Binär-Code.

Arbeitsanregung

1 Schau dir den Filmausschnitt aus der Sachgeschichten-DVD zum Thema „Computer“ an und fülle anschließend den folgenden Lückentext aus.

Ein Computer besteht aus _____ und _____ .

Ein Teil eines Computers ist die CPU, man nennt sie auch _____ . Sie besteht aus Milliarden kleiner Transistoren. Schaltet man den Computer ein, wird zunächst das _____ aufgerufen, das die angeschlossene Hardware überprüft, z. B. die _____, _____ oder _____ .

Es ist in einem Speicher abgelegt, den man nur lesen, aber nicht beschreiben kann.

Einen solchen Speicher nennt man _____ .

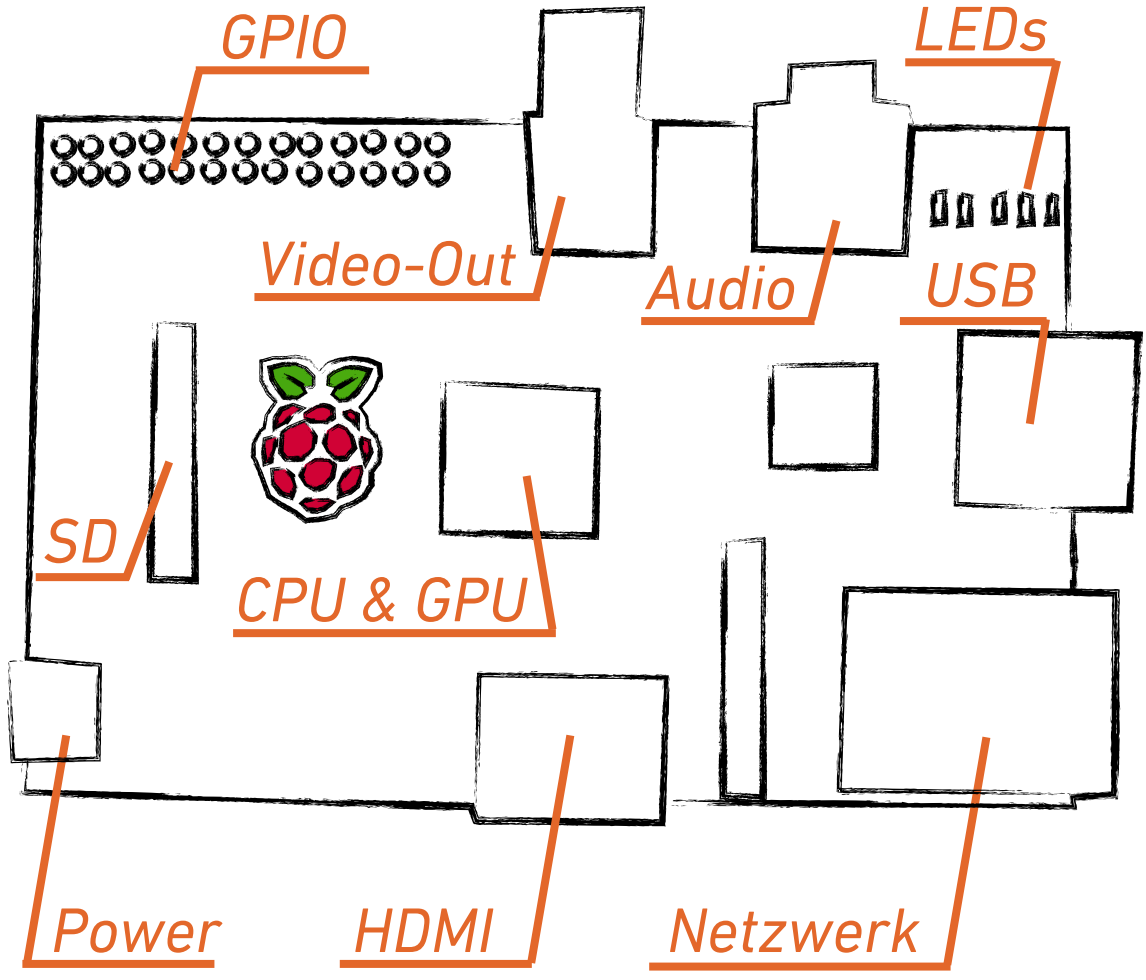
Wortspeicher: Festplatte, BIOS, Software, Tastatur, ROM, Hardware, Prozessor, Grafikkarte

M7 - Der Raspberry Pi

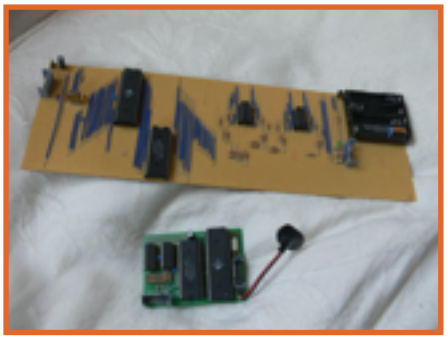
Der Raspberry Pi kostet nur ca. 32 Euro und wurde von einer gemeinnützigen Organisation in England entwickelt.

Die kleine Platine ist ein vollwertiger Computer, der jedoch einige Besonderheiten aufweist. Er besitzt z. B. kein BIOS und keine Festplatte. Stattdessen werden die Daten auf einer SD-Karte gespeichert. Er besitzt auch keinen Ein- oder Ausschalter. Sobald man ein Stromkabel anschließt, startet der Raspberry Pi, was durch das Leuchten der LEDs angezeigt wird.

Monitore lassen sich über den HDMI-Anschluss oder den Video-Ausgang anschließen. Über USB können Maus und Tastatur verbunden werden, Lautsprecher oder Kopfhörer steckt man in den Audio-Ausgang.



M8 - Wie entstand der Raspberry Pi?



Bildquelle: www.raspberrypi.org

Die Idee, einen kleinen und günstigen Computer für Kinder zu bauen, entstand 2006, als Eben Upton und seine Kommilitonen im Computerlabor der Universität in Cambridge, darunter Rob Mullins, Jack Lang und Alan Mycroft, sich Sorgen darüber machten, dass die Fähigkeiten der neuen Studentinnen und Studenten im Fach Computerwissenschaften von Jahr zu Jahr schlechter wurden. 1990 waren die meisten Neuankömmlinge noch erfahrene Hobbyprogrammierer, im Jahr 2000 war das schon ganz anders; ein typischer Erstsemester-Student hatte lediglich etwas Erfahrung im Erstellen von Internetseiten.

Etwas hat sich also geändert im Umgang von Kindern mit Computern und dafür gibt es mehrere Gründe: Zum Beispiel beschränkt sich der Lehrplan für das Fach Informatik immer mehr darauf, lediglich den Umgang mit Word und Excel bzw. dem Erstellen von Internetseiten beizubringen. Außerdem platzte die Dotcom-Blase und PCs und Spielekonsolen verdrängten die alten Amigas, BBC Micros, Spectrum ZX und Commodore 64, auf denen frühere Generationen das Programmieren gelernt haben.

Es gibt nicht viel, was eine kleine Gruppe von Leuten tun kann, um Probleme wie unzureichende Lehrpläne oder das Ende einer Spekulationsblase anzugehen. Aber wir hatten das Gefühl, dass wir es zumindest versuchen können, etwas daran zu ändern, dass Computer so teuer und undurchsichtig geworden sind, dass das Herumspielen daran von den Eltern verboten wird. Wir wollten ein System schaffen, das wie die alten Heimcomputer mit einer Programmieroberfläche starten kann. Von 2006 bis 2008 entwarf Eben verschiedene Versionen von dem Gerät, das heute der Raspberry Pi ist. Der erste Prototyp sah übrigens so aus:

[...] Wir behaupten nicht, dass wir alle Antworten haben, Wir glauben auch nicht, dass der Raspberry Pi die Lösung aller auf Computern basierenden Probleme dieser Welt ist; aber wir glauben, dass wir ein Katalysator sein können. Wir wollen, dass es überall günstige, zugängliche und programmierbare Computer gibt. Wir rufen andere Firmen dazu auf, unsere Idee nachzuahmen. Wir wollen nicht, dass Familien nur dann das Internet benutzen können, wenn sie hunderte Pfund ausgeben. Wir wollen, dass es normal wird, dass jedes Kind einen wirklich eigenen Computer besitzt. Wir glauben, dass 2012 ein sehr spannendes Jahr wird.

Übersetzung aus dem Englischen - Quelle: http://www.raspberrypi.org/about

M9 - Die Kommandozeile des Raspberry Pi

Wenn du den Raspberry Pi zum ersten Mal einschaltest, sieht du nur eine Kommandozeile. Jeder Computer besitzt so eine Kommandozeile (auch Geräte wie das iPhone), sie wird jedoch meist durch eine graphische Benutzeroberfläche (z. B. Windows) „verdeckt“.

Um den Raspberry Pi nutzen zu können, muss sich zunächst einmal anmelden:

raspberrypi login: **pi**

Password: **raspberry**

(Achte darauf, dass das „y“ und „z“ bei der englischen Tastatur, die zunächst Standard ist, vertauscht sind.)

Probiere anschließend einmal die folgenden Befehle aus:

free	*zeigt an, wie viel Arbeitsspeicher verfügbar ist.*
df /	*zeigt an, wie viel Platz auf der SD-Karte verfügbar ist.*
lsusb	*zeigt an, was alles an den beiden USB-Ports angeschlossen ist.*
pwd	*zeigt an, in welchem Verzeichnis man sich befindet.*
ls	*zeigt die Dateien des aktuellen Verzeichnisses an.*
ls -l	*zeigt noch mehr Informationen über diese Dateien an.*
startx	*startet die graphische Benutzeroberfläche. (Führe diesen Befehl erst aus, nachdem du die Arbeitsanregung unten erledigt hast.)*

Arbeitsanregung

1 Lasse dir das aktuelle Datum mit diesem Befehl anzeigen:

date

Wenn es falsch ist, gib es erneut ein mit folgendem Befehl:

sudo date 02111500

Die Zahl in dem Beispiel würde das Datum auf den 11. Februar und die Uhrzeit auf 15.00 Uhr setzen. Mit „sudo“ ändert man übrigens Einstellungen, die den gesamten Computer betreffen. Der Begriff „sudo“ ist also eine Art Schutz davor, dass der Benutzer aus Versehen etwas ändert. Man sollte den Befehl daher nur benutzen, wenn man genau weiß, was man tun will.

M10 - Open Source Software

Wenn Software als „Open Source“ bezeichnet wird, bedeutet das vier Dinge[7]:

1.) Jeder hat das Recht, die Software nach Belieben zu kopieren und weiterzugeben

2.) Jeder hat das Recht, den Quelltext der Software einzusehen.

3.) Jeder hat das Recht, die Software zu verändern und in veränderter Form weiterzugeben.

4.) Jeder hat das Recht, die Software für jeden Zweck einzusetzen.

Vielleicht nutzt du schon Open Source Software, ohne es zu wissen. Der Browser „**Firefox**“ ist zum Beispiel Open Source, ebenso der bekannte Mediaplayer „**VLC**“ oder das Audiobearbeitungsprogramm „**Audacity**“.

Es gibt jedoch nicht nur einzelne Programme, sondern auch ganze Betriebssysteme, die Open Source sind. Das bekannteste ist „**Linux**“, das auch auf dem Raspberry Pi zum Einsatz kommt. Da es sich um eine leicht abgewandelte Version handelt, heißt das Betriebssystem jedoch nicht Linux, sondern **Raspbian „wheezy**“.

Es laufen auch andere Betriebssysteme auf dem Raspberry Pi, z. B. „**Android**“, das sonst auf Smartphones und Tablets zum Einsatz kommt. Windows funktioniert nicht auf dem Raspberry Pi, da es nicht mit dem ARM-Prozessor des Geräts kompatibel ist.

Arbeitsanregung

1 Probiere zu Hause einmal selbst das freie Betriebssystem Linux aus, indem du deinen Computer zu Hause mit der ausgeteilten DVD startest. Du musst eventuell zuvor im BIOS einstellen, dass der Rechner nicht von der Festplatte, sondern dem DVD-Laufwerk startet. Meistens muss man dazu die Taste „DEL“ „F1“ oder „ENTF“ nach dem Einschalten drücken. Da jedes BIOS ein wenig anders aufgebaut ist, gibt es keine allgemeingültige Anleitung. Meist muss man mit den Pfeiltasten in das Register „Boot“ wechseln und dann mit der Taste „+“ oder „-“ die Boot-Reihenfolge so ändern, dass das DVD-Laufwerk ganz oben steht.

---

7 Quelle: http://www.heise.de/open/artikel/Die-Woche-Was-ist-Open-Source-222121.html

M11 - Libre Office

Auf dem Raspberry Pi ist ein komplettes Office-Paket installiert, das nahezu alle Funktionen bietet, die du vielleicht von Programmen wie „Word“; „Excel“ oder „Powerpoint“ kennst.

Um „Libre Office“ zu starten, musst du zunächst einmal die graphische Benutzeroberfläche aufrufen. Dazu musst du in der Kommandozeile folgenden Befehl eintippen:

```
startx
```

Durch einen Klick auf das Symbol unten links kannst du unter dem Menü „Office“ die verschiedenen Office-Anwendungen starten. Zur Auswahl stehen:

Writer - *Textverarbeitungsprogramm mit Rechtschreibkorrektur*

Calc - *Tabellenkalkulation mit der Möglichkeit, Diagramme zu erstellen*

Impress - *Präsentationsprogramm*

Draw - *Zeichenprogramm*

Base - *Erstellen von Datenbanken*

Arbeitsanregung

1 Einigt euch zu zweit auf eine der folgenden Aufgaben, für die ihr dann 30 Minuten Zeit habt:

- Erstellt eine **Einladung** zu einer Geburtstagsfeier mit dem Programm „**Writer**“

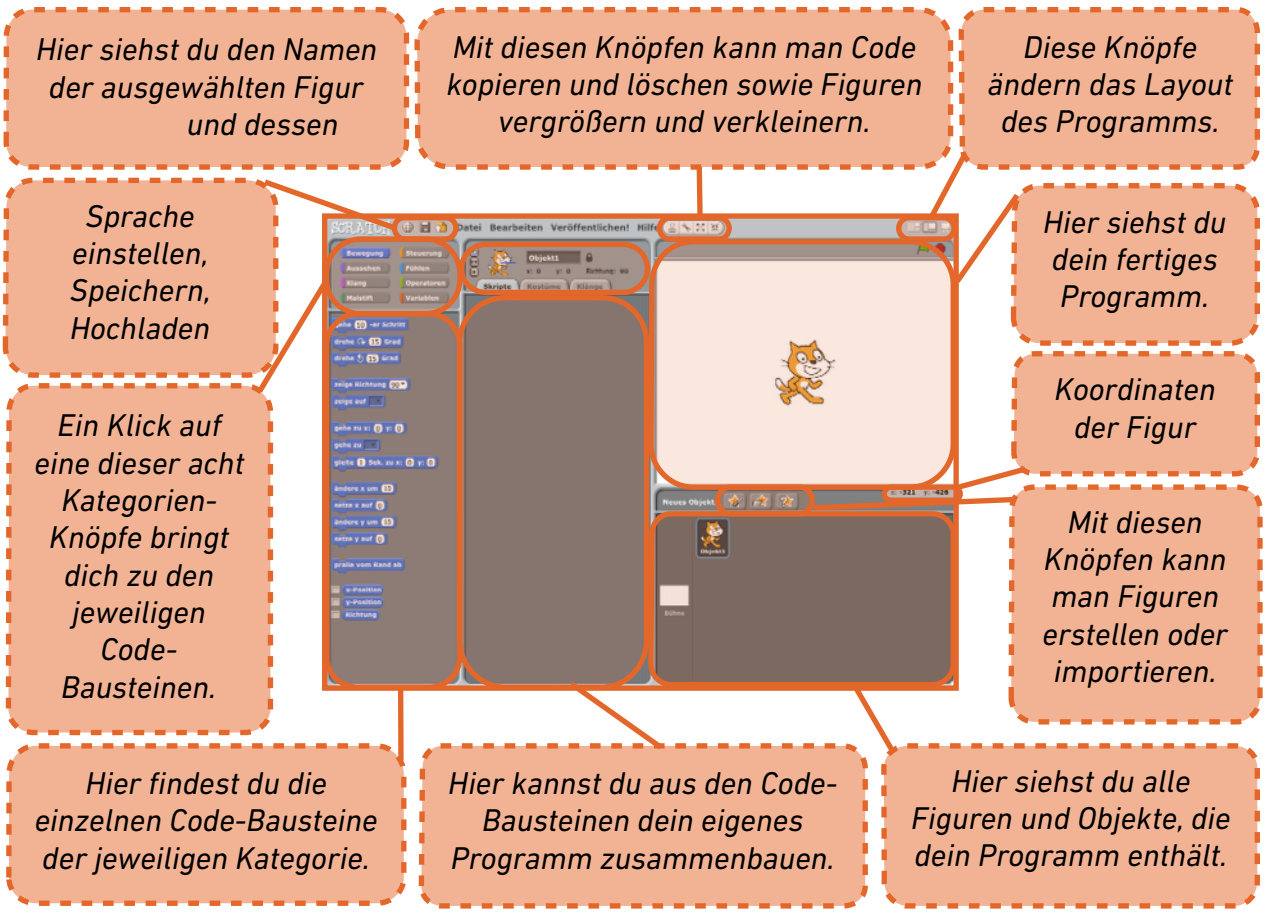
- Sammelt einige **Daten** über die Teilnehmerinnen und Teilnehmer der Computer-AG und stellt diese mit Hilfe von „**Calc**“ in Diagrammen dar. Ihr könnt z. B. Erstellen zum Geschlecht, der Haarfarbe und -länge, der Augenfarbe, dem Alter, dem Geburtsmonat etc.

- Erstellt eine **Präsentation** mit „**Impress**“, in der ihr Grundschülern die Vorteile eurer Schule zeigt.

M12 - Scratch - Der Aufbau des Programms I

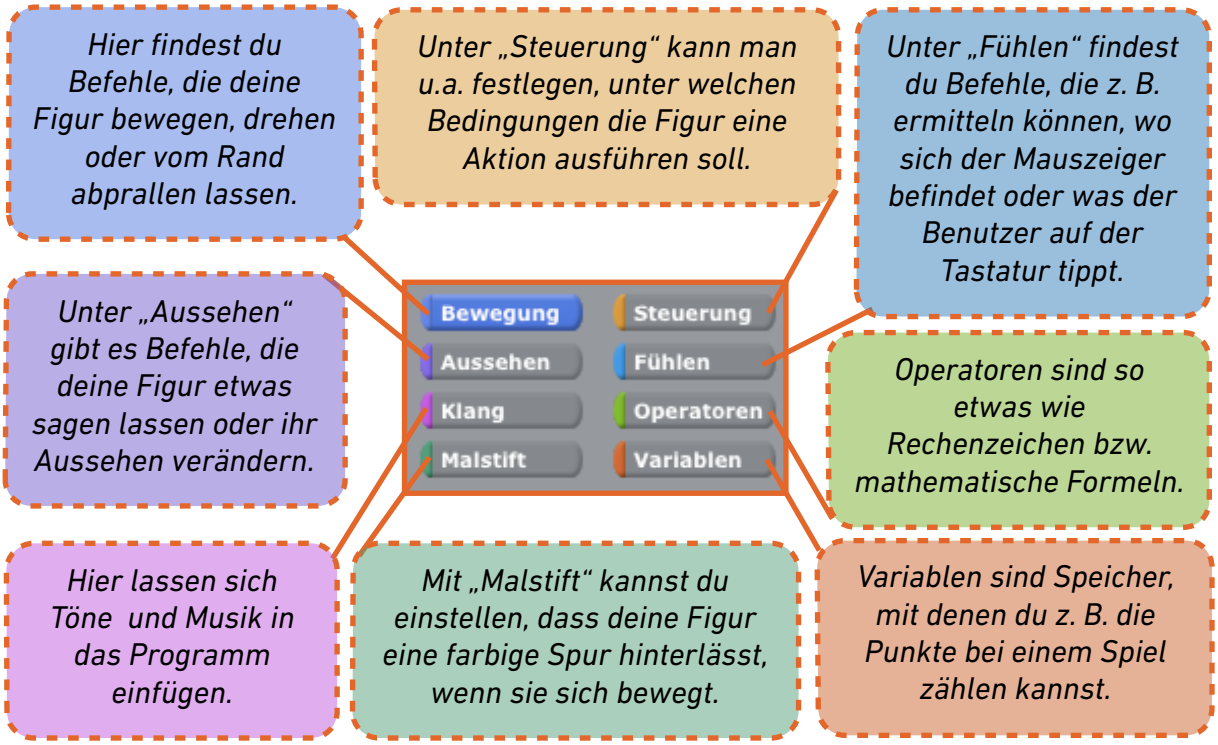
Scratch ist eine kostenlose Programmiersprache, die vom weltberühmten Massachusetts Institute of Technology entwickelt wurde. Sie soll den Einstieg ins Programmieren so leicht wie möglich machen. Auf der Homepage http://scratch.mit.edu kannst du die Software herunterladen und dir Programme ansehen, die Kinder auf der ganzen Welt mit Scratch erstellt haben.

im Gegensatz zur Programmiersprache „Python“, die wir uns später etwas genauer ansehen werden, besitzt Scratch eine graphische Oberfläche und kann fast komplett mit der Maus bedient werden. Um ein Programm zu schreiben, muss man lediglich die einzelnen Befehle wie Bauklötze aufeinander stapeln. Nach dem Start des Programms siehst du folgende Oberfläche:

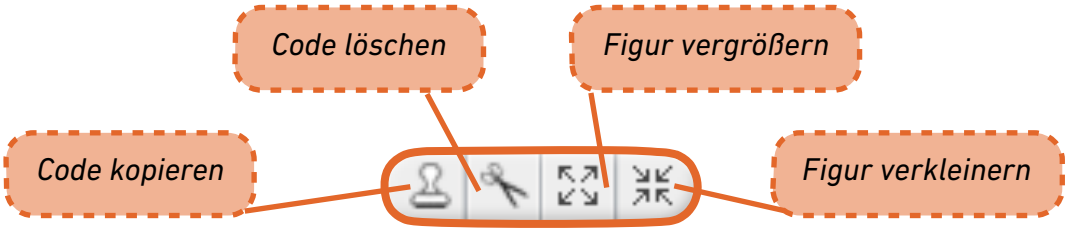


M13 - Scratch - Teil 1: Der Aufbau des Programms II

Die Code-Bausteine, mit denen du dein eigenes Programm schreiben kannst, sind in verschiedene Kategorien eingeteilt, damit du sie leichter finden kannst:



Beachte, dass jede Kategorie eine eigene Farbe hast. So kannst du immer genau erkennen, zu welcher Kategorie ein Befehl gehört. Bevor wir unser erstes Programm mit Scratch schreiben, solltest du noch vier anderen Knöpfe kennen lernen, die sich am oberen Rand befinden:

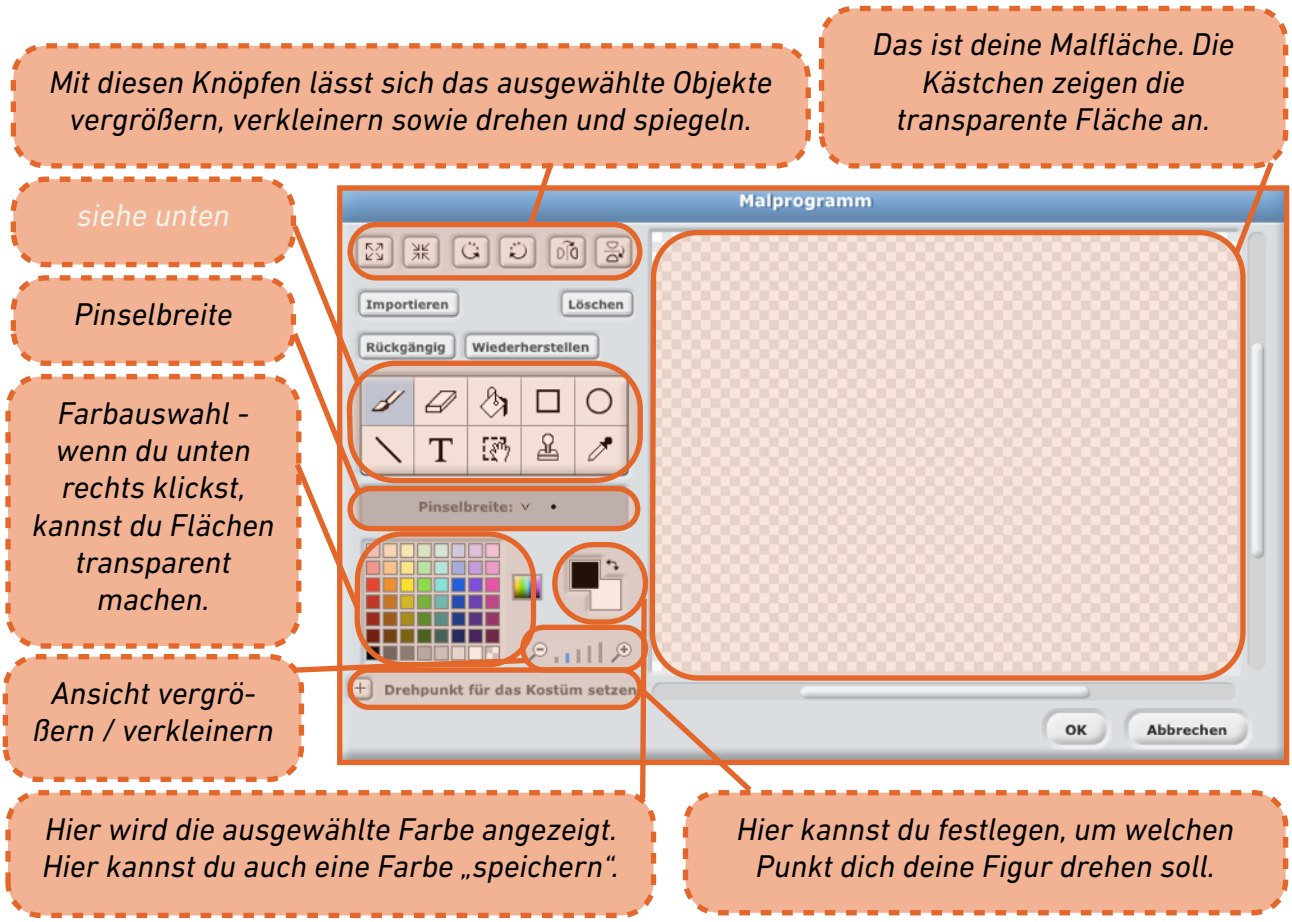


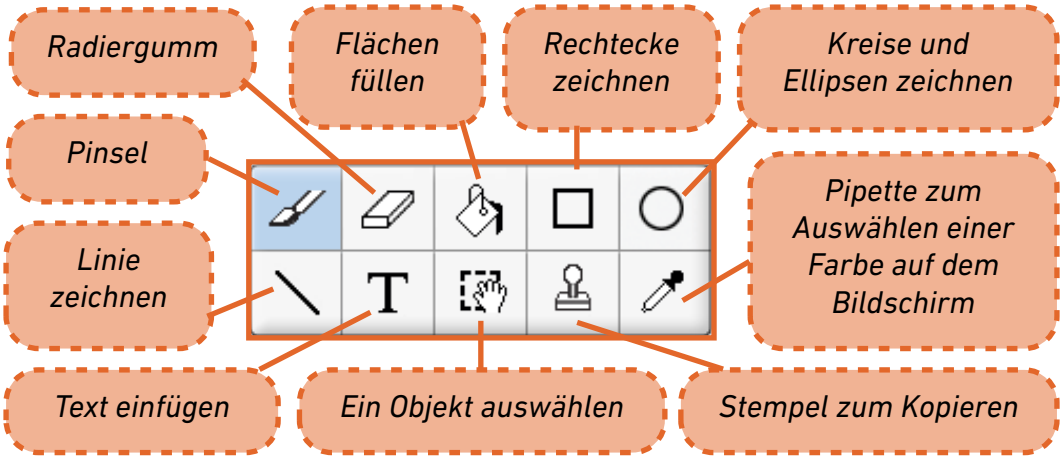
Vor allem der Knopf „Code kopieren“ kann dir später viel Arbeit ersparen, da du bereits vorhanden Code einfach kopieren und abändern kannst, statt ihn neu zu tippen.

M14 - Scratch - Eine eigene Figur entwerfen

Mit Scratch kannst du auch eigene Figuren erstellen, indem du den eingebauten Editor aufrufst. Drücke dazu einfach auf diesen Knopf:

Folgende Werkzeuge stehen dir zum Erstellen deiner Figur zur Verfügung:

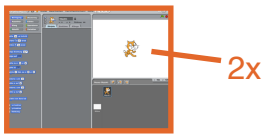




M15 - Scratch - Ein Programm schreiben I

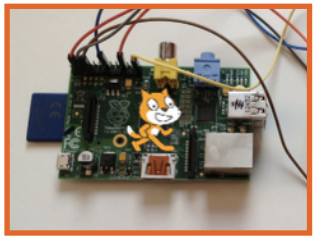
Nun weißt du alles, was nötig ist, um ein eigenes kleines Programm zu schreiben. Wie wäre es mit einem kleinen Lernprogramm für den Raspberry Pi?

Wir beginnen damit, einen Hintergrund einzufügen. Klicke dazu zwei Mal auf den weißen Hintergrund hinter der Katze.



Klicke anschließend auf „Hintergründe“ und schließlich auf „Importieren“. Wähle anschließend die Datei „Pi“ aus[8], indem du doppelt auf die klickst.

Nun sollte dein Hintergrund wie folgt aussehen:

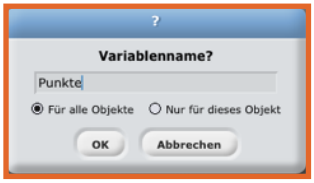


Für unser Programm brauchen wir ein neues „Kostüm“ für die Katze, so dass sie in die andere Richtung schaut. Klicke dafür doppelt auf die Katze und anschließend auf „Kostüme“. Kostüm Nummer zwei kannst du löschen, indem du auf das kleine „x“ klickst.

Von Kostüm 1 hingegen brauchen wir eine Kopie. Klicke daher auf „Kopieren“ und anschließend auf „Bearbeiten“.

Nun musst du auf „Horizontal spiegeln“ klicken und anschließend auf „OK“.

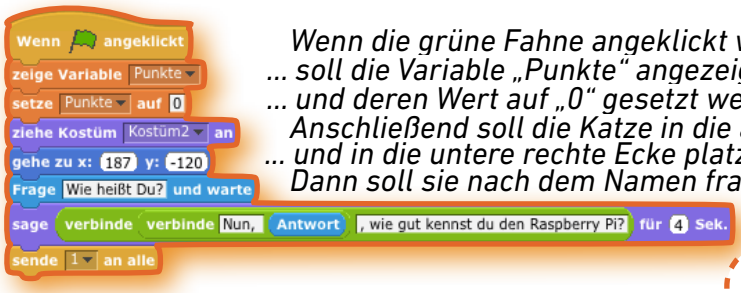
Nun brauchen wir nur noch eine Variable, die die Punkte zählt. Klicke dazu in der Kategorie „Variablen“ auf den Knopf „Neue Variable“. Als Name für die Variable solltest du „Punkte“ wählen und mit „OK“ bestätigen.



[8] http://www.medienistik.de/Pi.jpg

M16 - Scratch - Teil 4: Ein Programm schreiben II

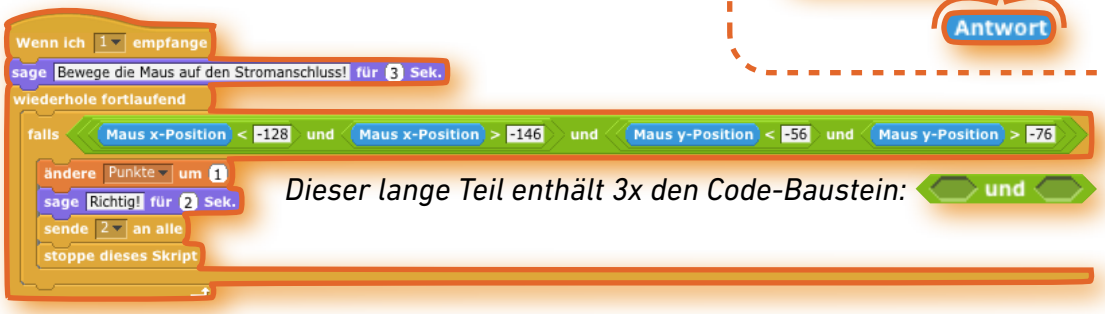
So - nun haben wir alle Vorarbeiten erledigt und können mit dem eigentlichen Programmieren beginnen. Das fertige Programm sollte wie folgt aussehen:



Wenn die grüne Fahne angeklickt wird...
... soll die Variable „Punkte“ angezeigt ...
... und deren Wert auf „0“ gesetzt werden.
Anschließend soll die Katze in die andere Richtung schauen...
... und in die untere rechte Ecke platziert werden.
Dann soll sie nach dem Namen fragen und auf Antwort warten.

Nicht erschrecken, das nächste Programm sieht etwas kompliziert aus, ist es aber nicht. Ziehe einfach wieder die entsprechenden Code-Bausteine an die entsprechende Stelle.

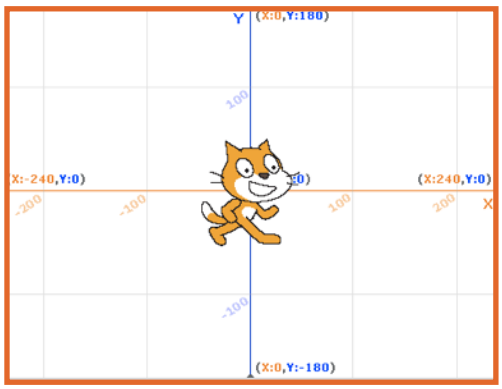
Um diesen Code nachzubauen, ziehe die jeweiligen Bausteine an die angegebene Stelle und ändere den Text entsprechend.



Dieser lange Teil enthält 3x den Code-Baustein: und

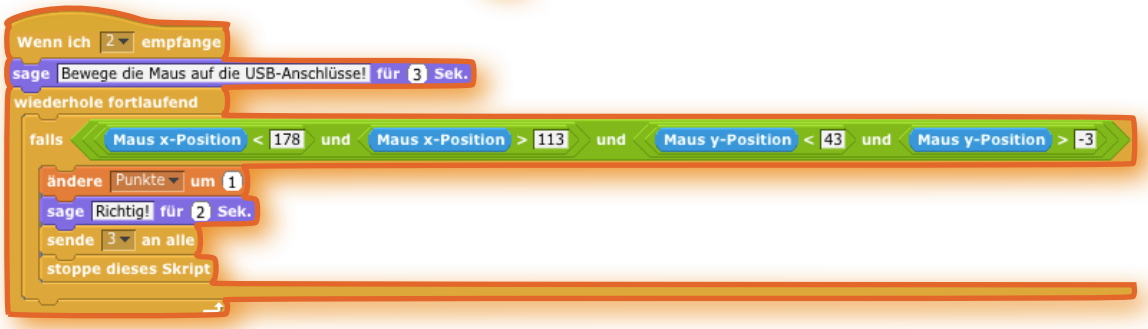
Dieses Programm enthält eine Schleife, in der fortlaufend die Position der Maus abgefragt wird. Sobald die Maus in den angegebenen Bereich bewegt wird, wird die Schleife beendet und die Punktezahl um 1 erhöht.

Scratch benutzt übrigens ein Koordinatensystem, um die Position eines Objekts zu bestimmen. Die y-Achse geht von oben nach unten, die x-Achse von links nach rechts, in der Mitte ist der Nullpunkt.



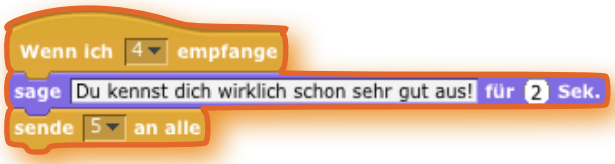
M17 - Scratch - Teil 4: Ein Programm schreiben III

Nun fügen wir noch ein paar neue Aufgaben hinzu, indem wir den zuletzt geschriebenen Code-Block kopieren (mit der Taste) und etwas abändern, nämlich wie folgt:

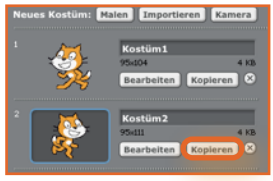


Und das Ganze noch einmal:

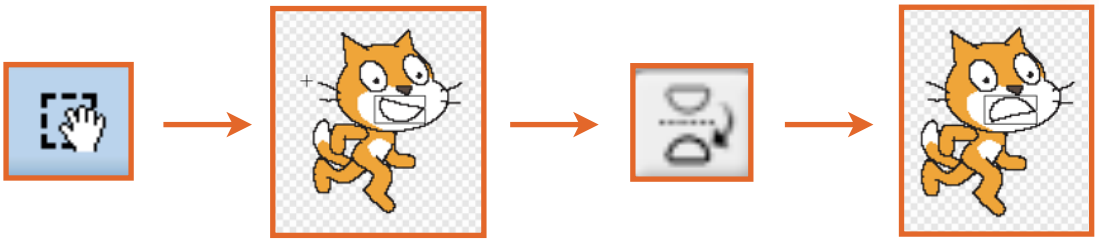
Nun beenden wir diesen Teil des Programms mit folgendem Code:



Für den nächsten Teil benötigen wir noch ein zusätzliches Kostüm. Klicke daher doppelt auf die Katze, kopiere Kostüm 2 und klicke anschließend auf „Bearbeiten“.



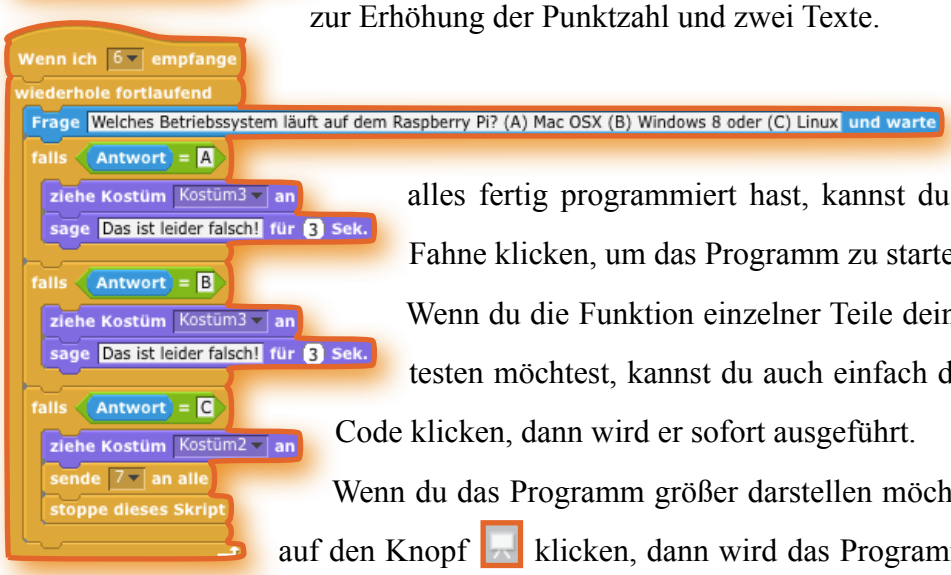
Nun wollen wir aus dem lachenden einen traurigen Mund machen. Markiere dazu den Mund mit dem Auswahl-Werkzeug und drücke anschließend den „Vertikal spiegeln“-Knopf. Auf diese Weise kannst du schnell einfache Animationen erstellen.



M18 - Scratch - Teil 4: Ein Programm schreiben IV

Um unsere Spielfigur mit den Pfeiltasten steuern zu können, benötigen wir wieder eine Schleife, die überprüft, ob der Benutzer eine Taste drückt. Zudem soll das Programm überprüfen, ob die Spielfigur die SD-Karte berührt. Dazu benötigen wir den wird Farbe berührt?-Baustein. Wenn du auf die Farbe klickst, verwandelt sich der Mauszeiger in eine Pipette. Wenn du anschließend auf die SD-Karte klickst, merkt sich das Programm diese Farbe.

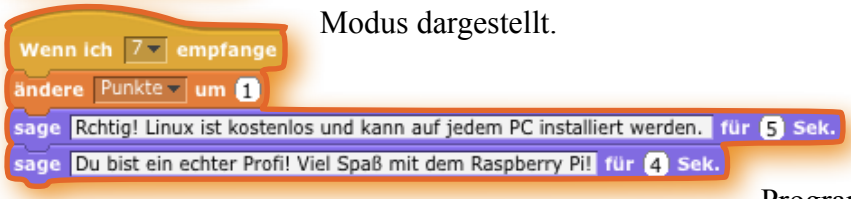
Jetzt fehlen nur noch zwei Code-Blocks. Der eine stellt dem Benutzer eine Frage, die er beantworten soll, der andere beinhaltet nur einen Befehl zur Erhöhung der Punktzahl und zwei Texte.



Wenn du alles fertig programmiert hast, kannst du auf die grüne Fahne klicken, um das Programm zu starten:

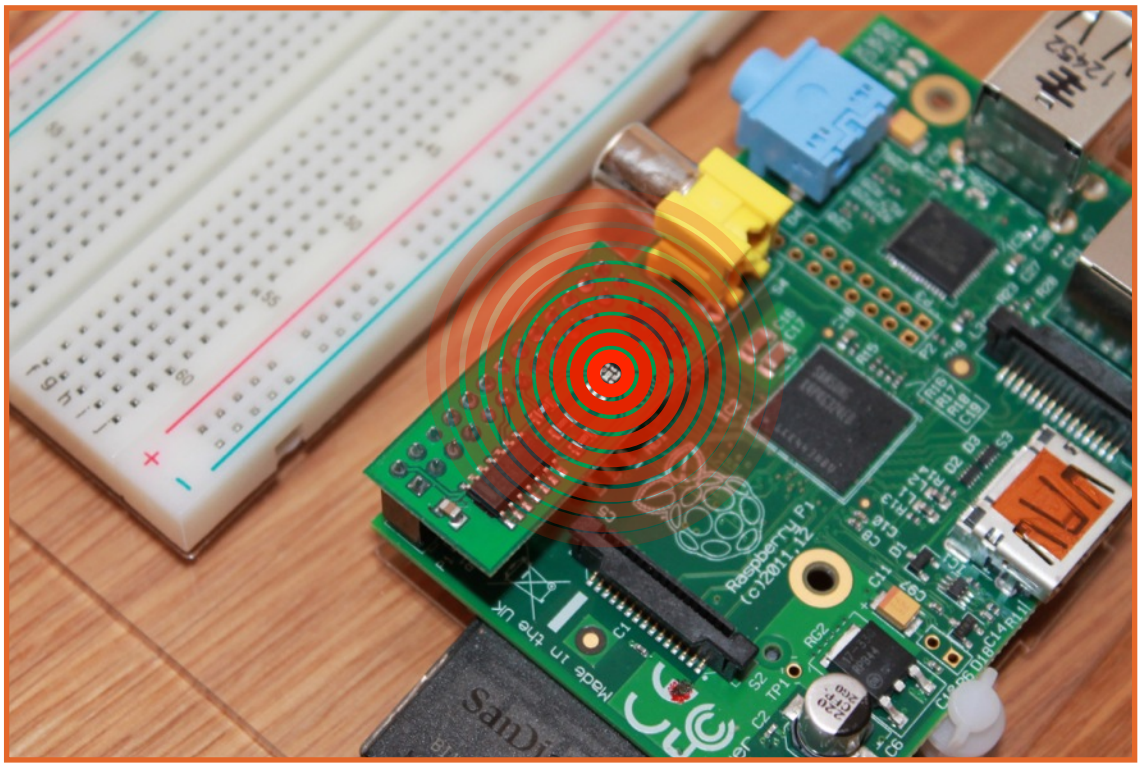
Wenn du die Funktion einzelner Teile deines Programms testen möchtest, kannst du auch einfach doppelt auf den Code klicken, dann wird er sofort ausgeführt.

Wenn du das Programm größer darstellen möchtest, kannst du auf den Knopf klicken, dann wird das Programm im Vollbild-Modus dargestellt.



Nun kannst du dich daran machen, das Programm auszubauen und zu verbessern, z. B. indem die Katze am Ende die Punktzahl verkündet oder automatisch wieder in die untere rechts Ecke wandert.

M19 - Scratch - Physical Computing mit der LEDBorg-Platine



Wenn du die LEDBorg-Platine auf deinen Raspberry Pi steckst, kannst du sie ganz einfach mit Scratch zum Leuchten bringen.

Starte dafür zunächst das Programm „ScratchGPIO“ auf deinem Desktop und erstelle folgende drei Variablen:

... für blaues Licht
... für grünes Licht
... für rotes Licht

Jede dieser Variablen kann einen Wert von 0 (aus) bis 100 (sehr hell) haben. Um die LED grün blinken zu lassen, kannst du also das folgende Programm schreiben:

Nun kannst du die LED in das Lernprogramm einbinden, das du zuvor erstellt hast. Ändere dazu den Code so um, dass die LED immer dann grün leuchtet, wenn der Benutzer einen Punkt erhält. Wenn das Programm zu Ende ist, kann die LED als Belohnung bunt leuchten.

M20 - Programmieren mit Minecraft I

Bildquelle: www.minecraftwiki.net

Mit dem Spiel Minecraft kann man nicht nur viel Spaß haben, sondern auch die Grundlagen des Programmierens erlernen. Dafür benutzen wir eine extra vom Programmierer für den Raspberry Pi entwickelte Version des Spiels.

Starte zunächst das Spiel, indem du die graphische Benutzeroberfläche mit dem Befehl „**startx**“ startest. Klicke anschließend mit der Maus auf den Knopf unten links, wähle dann „**Accessories**“ und anschließend „**File Manager**“. Mache dann einen Doppelklick auf den Ordner „**mcpi**“ und klicke dann auf das kleine Zehnrad mit der Unterschrift „**minecraft-pi**“ und anschließend auf „**execute**“. Starte nun das Spiel und erstelle eine neue Welt.

Um die Programmierfunktion zu aktivieren, musst du anschließend das Symbol „**LXTerminal**“ auf dem Desktop doppelklicken. Falls du Probleme damit hast, dass die Maus nicht angezeigt wird, klicke einfach auf die **Tabulatortaste** links neben der Taste „Q“. Damit „befreist“ du die Maus und kannst woanders hinklicken.

Um dich in das Spiel „einzuhacken“, musst du einige Befehle eingeben. Nach jeder Zeile musst du die Enter-Taste drücken. Achte sorgfältig darauf, dass du dich nicht vertippst.

Zunächst musst du in das richtige Verzeichnis wechseln mit dem Befehl:

```
cd mcpi/api/python
```

Starte anschließend die Programmiersprache Python, indem du einfach eingibst:

```
python
```

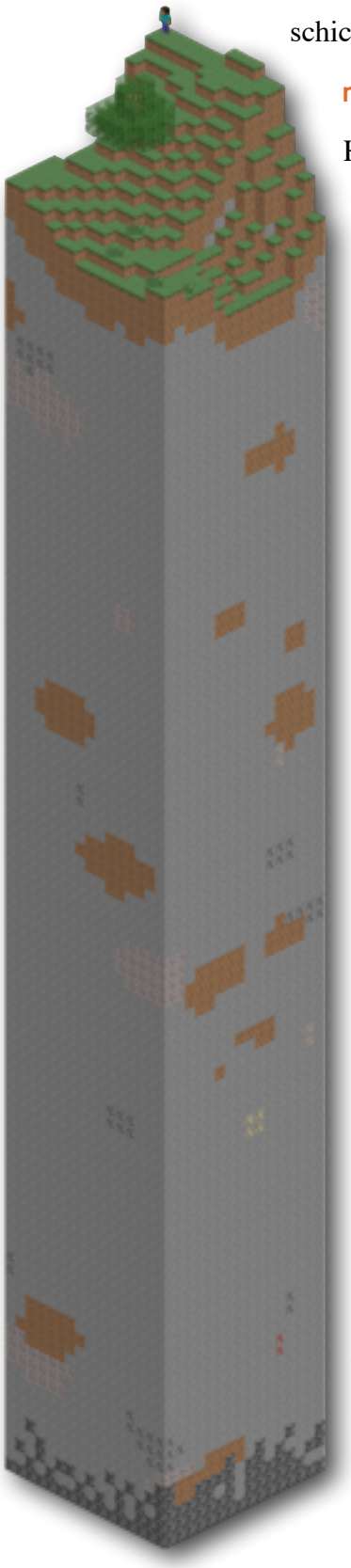
Nun solltest du links am Rand folgende Zeichen sehen:

```
>>>
```

Jetzt gib einfach Folgendes ein:

```
import minecraft as minecraft
mc = minecraft.Minecraft.create()
```

M21 - Programmieren mit Minecraft II



Bildquelle: www.minecraftwiki.net

Wenn alles funktioniert hat, kannst du jetzt kleine Botschaften an das Spiel schicken, z. B.:

```
mc.postToChat(„Hallo Minecraft!“)
```

Hat das funktioniert? Dann kannst du jetzt verschiedene Daten aus dem Spiel abrufen, z. B. deine Position:

```
mc.player.getPos()
```

Diese Position wird auch immer oben links am Bildschirm angezeigt. Nun kannst du ein wenig zurückgehen und an diese Position einen Stein setzen mit dem Befehl (erst weiterlesen, den Befehl jetzt noch nicht abtippen):

```
mc.setBlock(x,y,z,[block-ID])
```

Die Buchstaben „x“, „y“ und „z“ stehen für die Koordinaten, die du mit dem vorherigen Befehl bekommen hast. Angenommen du hast mit dem vorherigen Befehl mc.player.getTilePos() folgende Rückmeldung bekommen:

```
Vec3(90,7,12)
```

Dann kannst du jetzt ein wenig zurückgehen und mit diesem Befehl einen Stein platzieren:

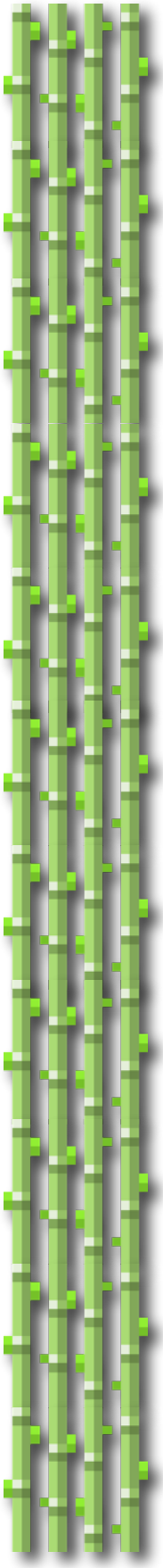
```
mc.setBlock(90,7,12,2)
```

Die Ziffer „2“ steht für das Material des Blocks (block-ID). Drücke die Pfeil nach oben-Taste, um den letzten Befehl erneut anzuzeigen und ändere die letzte Zahl, um andere Blöcke zu setzen, also z. B.:

```
mc.setBlock(90,7,12,4)
mc.setBlock(90,7,12,12)
mc.setBlock(90,7,12,22)
```

Versuche nun, mehrere Blöcke auf einmal zu setzen. Dies funktioniert mit dem Befehl:

```
mc.setBlocks(x1,y1,z1,x2,y2,z2,[block-ID])
```



Bildquelle: www.minecraftwiki.net

M22 - Programmieren mit Minecraft III

Nun kannst du einige Programme ausprobieren, die andere Programmierer bereits mit Python geschrieben haben. Öffne dazu wieder das „**LXTerminal**“, wechsle in das Verzeichnis **cd mcpi/api/python** und starte „**python**“. (Wenn ein Programm nicht funktioniert und du kein neues starten kannst, klicke auf „**File**“ und „**New Tab**“. Dann kannst du wieder „**python**“ eingeben und ein neues Programm starten.)

Wenn ein Programm nicht funktioniert, drücke die „ESC“-Taste in Minecraft, klicke auf „**Quit to title**“ und erschaffe unter „**Start Game**“ eine neue Welt mit dem Knopf „**Create new**“. Hier sind die Befehle zum Starten der Programme in Python:

execfile (´stuffaboutcode_basics.py´)
Ein Programm, das zeigt, was man mit Python und Minecraft machen kann.

execfile (´sleepyoz_digitalclock.py´)
Dieses Programm erschafft eine Digitaluhr in der Luft.

execfile (´stuffaboutcode_bridge.py´)
Lässt überall, wo du hintrittst, eine Brücke erstehen.

execfile (´davef21370_maze.py´)
Erschafft ein Labyrinth auf dem Boden.

execfile (´nt7s_sphere.py´)
Erschafft eine goldene „Kugel“ neben dem Spieler.

execfile (´gf_clearZone.py´)
Räumt in deinem Umfeld alle Steine aus dem Weg.

execfile (´brooksc_flatmap50.py´)
Räumt noch mehr Steine aus dem Weg.

execfile (´daviewales_minesweeper.py´)
Erschafft ein kleines Minesweeper-Spiel.

execfile (´zhuowei_rainbow.py´)
Erschafft einen wunderschönen Regenbogen.

execfile (´snowbound_flatmap.py´)
Löscht ALLE Steine - Achtung! Die Ausführung dauert sehr lange.

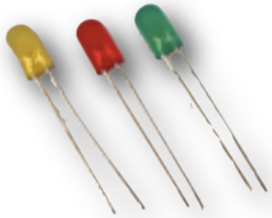
execfile (´gf_drawbuilding.py´)
Lässt in deiner Nähe ein Hochhaus erstehen.

execfile (´fleap_railgen.py´)
Schafft einen langen Gang mit einer Tür am Anfang.

execfile (´burnaron_bunkermatic.py´)
Lässt einen tiefen, einen seeeeehr tiefen Bunker erstehen.

M23 - Physical Computing mit dem Raspberry Pi I

Heute wollen wir uns etwas genauer mit den Möglichkeiten des Raspberry Pi beschäftigen und ein Programm schreiben, das eine LED zum Leuchten bringt. Die folgenden Ideen zu den Bastel-Projekten stammen von der englischen Weiterbildungs-Gesellschaft OCR. Das kleine „Bastelset“, das Du bekommen hast, enthält folgende Teile:

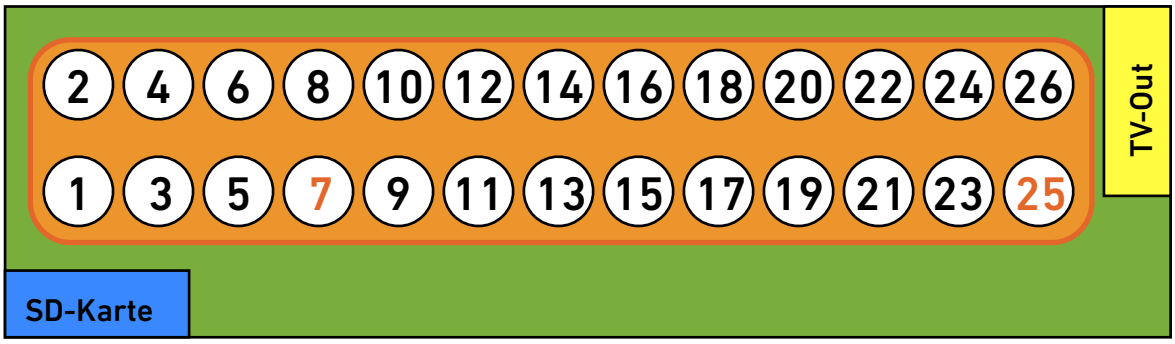
Drei LEDs	Drei 220 Ω Widerstände	6 Jumper-Kabel
		

WARNUNG:

Wenn du ein Kabel falsch anschließt, kann es sein, dass der Raspberry Pi kaputt geht. Schau also genau hin und frage lieber noch einmal nach, wenn Du Dir unsicher bist.

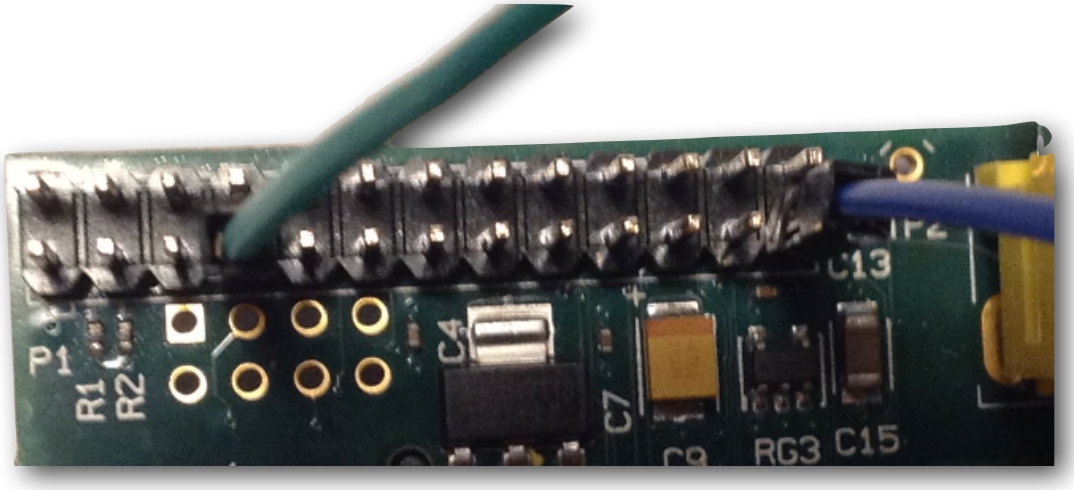
Zunächst wollen wir nur eine LED an den Raspberry Pi anschließen. Drehe dazu einen der Widerstände um das kürzere Ende der LED. Das sollte dann in etwa so aussehen:

Schließe nun die beiden Jumper-Kabel an und verbinde das Kabel, das mit dem Widerstand verbunden ist, mit **Pin 25** und das andere mit **Pin 7**. Die folgende Zeichnung zeigt, welche Pins das sind:

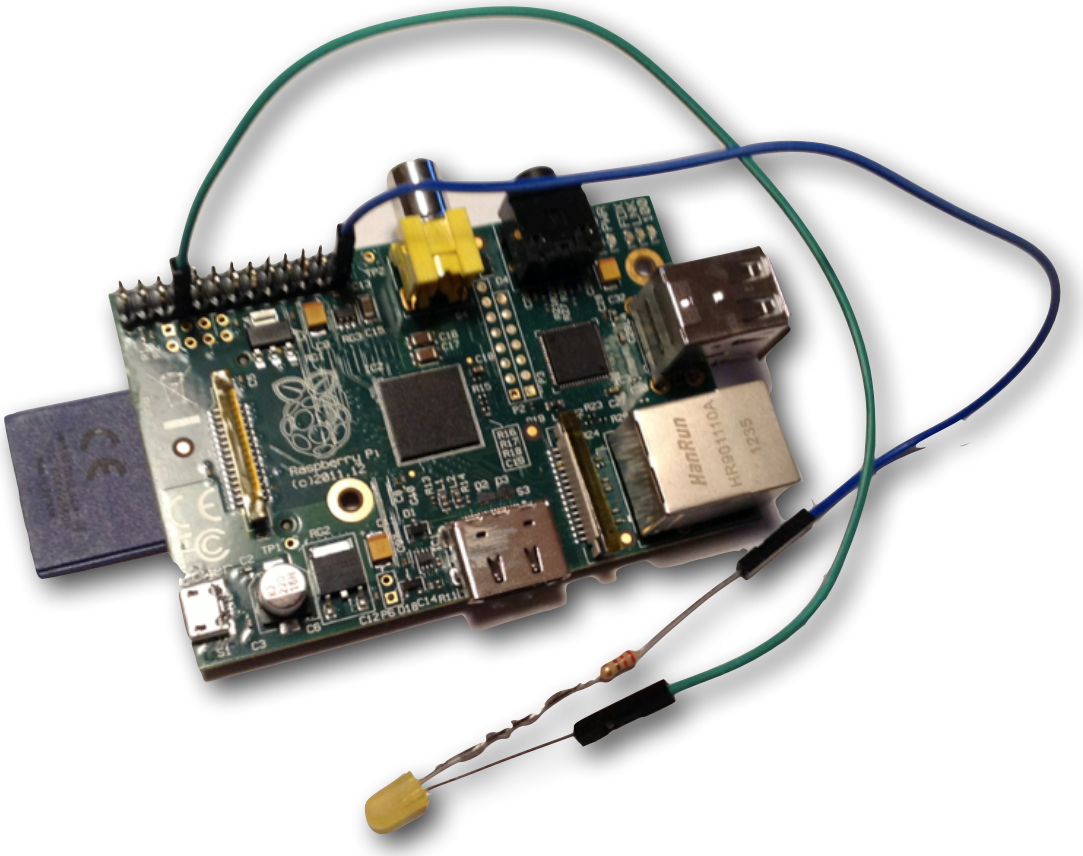


M24 - Physical Computing mit dem Raspberry Pi II

Hier siehst du noch einmal ein Foto, das dir zeigt, wo die Kabel angeschlossen werden müssen:



Und hier ist noch ein Bild der fertigen Bastelarbeit:



M25 - Physical Computing mit dem Raspberry Pi III

Nun muss noch das Programm geschrieben werden, das die LED leuchten lässt. Dafür musst du das Terminal öffnen und ein Programm schreiben. Um den Texteditor zu öffnen und eine Datei namens „Leuchte.py“ zu erstellen, musst du Folgendes eingeben:

```
nano Leuchte.py
```

Anschließend musst du die folgenden Codezeilen abtippen. Alle Codezeilen mit dem Zeichen # am Anfang sagen dir, was die jeweiligen Zeilen Code machen. Du brauchst sie nicht mit abschreiben.

```
# Am Anfang muss man eine so genannte „Library“, also eine Bibliothek
# importieren, damit wir dem Programm Befehle geben können, die nur
# eine bestimmte Zeit lang ausgeführt werden sollen.
import time
# Wir brauchen zudem noch eine weitere Bibliothek, um die GPIO-Schnittstelle
# anzusprechen.
import RPi.GPIO as GPIO
# Dann löschen wir erst einmal alle Belegungen dieser Schnittstelle, die
# eventuell noch vorhanden sind.
GPIO.cleanup()
# Nun müssen wir der Bibliothek sagen, wie wir die Pins nummerieren:
GPIO.setmode(GPIO.BOARD)
# Jetzt geben wir den Befehl, dass Pin 7 ein Output sein soll:
GPIO.setup(7,GPIO.OUT)
# Nun programmieren wir eine Schleife, in der wir...
while True:
        # ... die LED einschalten...
        GPIO.output(7,GPIO.HIGH)
        # ... eine Sekunde warten...
        time.sleep(1)
        # ...dann die LED wieder ausschalten...
        GPIO.output(7,GPIO.LOW)
        # ...und wieder eine Sekunde warten.
        time.sleep(1)
```

Nun kannst du die Datei speichern (STRG-X, dann „y“ drücken und anschließend mit Enter bestätigen) und mit folgendem Befehl aufrufen:

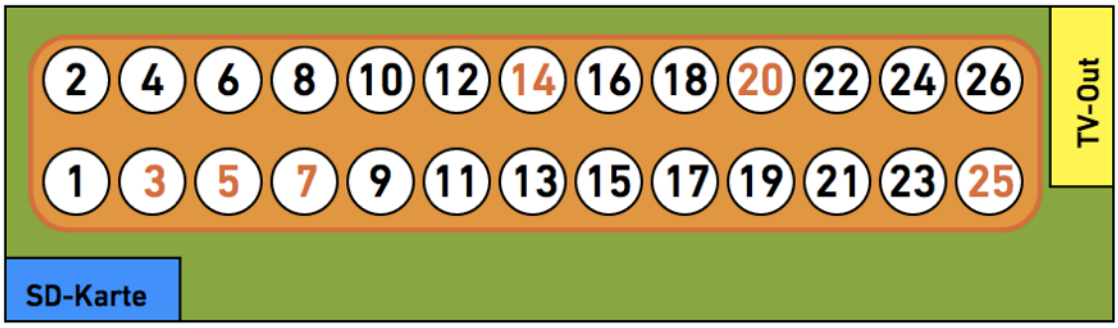
```
sudo python Leuchte.py
```

Die Tastenkombination „STRG-C“ bricht das Programm ab.

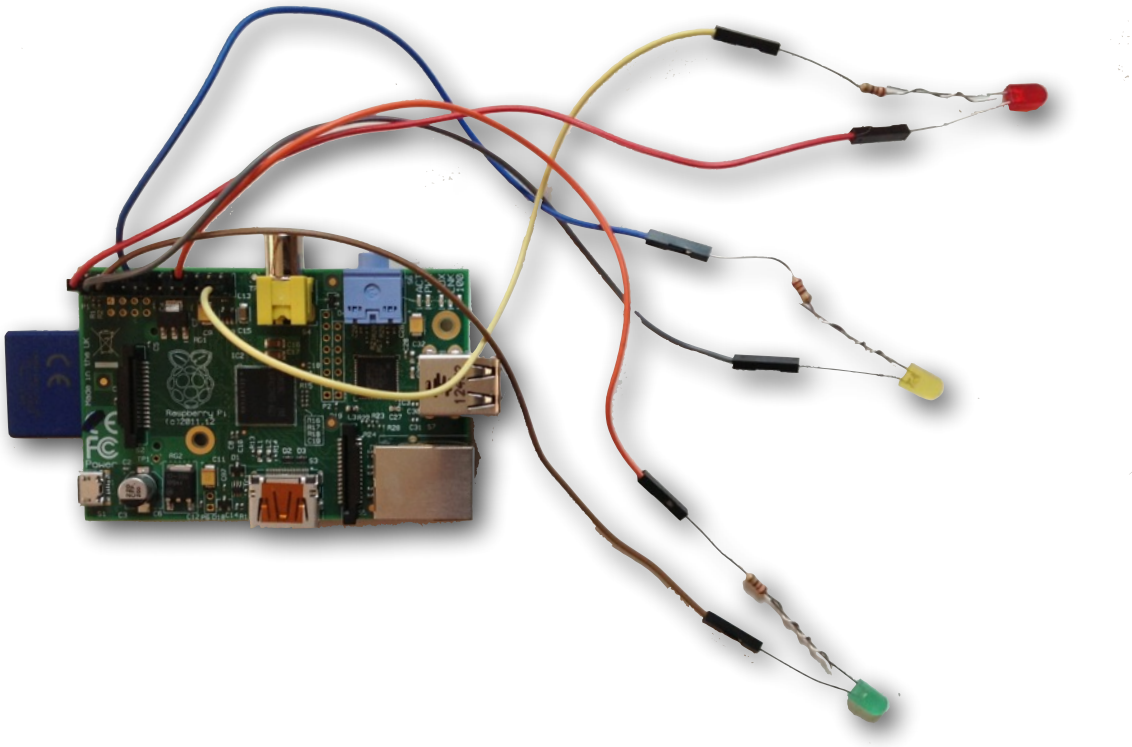
M26 - Physical Computing mit dem Raspberry Pi IV

Das nächste Projekt besteht darin, eine Ampel zu bauen. Dazu brauchst du alle drei LEDs, die du wieder so vorbereitest wie beim letzten Projekt. Denke daran, dass der Widerstand am kürzeren Ende der LED befestigt werden muss.

Stecke anschließend wieder Kabel an beide Enden der LED an und verbinde die Kabel, die an den Widerständen angeschlossen sind, mit den **Pins 14,20 und 25**. Anschließend kommt das andere Ende der **roten LED auf Pin3**, das der **gelben auf Pin5** und das der **grünen auf Pin7**. Hier ist zur Sicherheit noch einmal die Nummerierung der Pins:



Das Ergebnis sollte in etwa wie folgt aussehen:



M27 - Physical Computing mit dem Raspberry Pi V

Nun schreiben wir wieder ein Programm. Gib dafür folgenden Befehl im Terminal ein:

```
nano Ampel.py
```

Anschließend musst du folgenden Code eingeben:

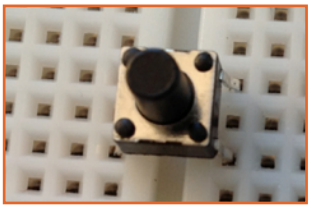
```
# Import der Bibliothek zur Zeitmessung
import time
# Import der Bibliothek für die GPIO-Schnittstelle
import RPi.GPIO as GPIO
# Löschen aller Belegungen dieser Schnittstelle
GPIO.cleanup()
# Nun müssen wir der Bibliothek sagen, wie wir die Pins nummerieren:
GPIO.setmode(GPIO.BOARD)
# Jetzt geben wir den Befehl, dass Pin 3 ein „Output“ sein soll:
GPIO.setup(3,GPIO.OUT)
# ... ebenso Pin 5...
GPIO.setup(5,GPIO.OUT)
# ... und Pin 7.
GPIO.setup(7,GPIO.OUT)
# Nun programmieren wir eine Schleife, in der wir...
while True:
    # ... die rote LED einschalten...
    GPIO.output(3,GPIO.HIGH)
    # ... zwei Sekunden warten...
    time.sleep(2)
    # ... die gelbe LED einschalten...
    GPIO.output(5,GPIO.HIGH)
    # ... zwei Sekunden warten...
    time.sleep(2)
    # ... die gelbe und rote LED ausschalten...
    GPIO.output(5,GPIO.LOW)
    GPIO.output(3,GPIO.LOW)
    # ... die grüne LED anschalten...
    GPIO.output(7,GPIO.HIGH)
    # ... zwei Sekunden warten...
    time.sleep(2)
    # ...dann die grüne LED wieder ausschalten...
    GPIO.output(7,GPIO.LOW)
    # ... und die gelbe LED einschalten...
    GPIO.output(5,GPIO.HIGH)
    # ...wieder zwei Sekunden warten....
    time.sleep(2)
    # ... und die gelbe LED ausschalten...
    GPIO.output(5,GPIO.OUT)
```

Nun kannst du die Datei speichern (STRG-X, dann „y“ drücken und anschließend mit Enter bestätigen) und mit folgendem Befehl aufrufen: `sudo python Ampel.py`

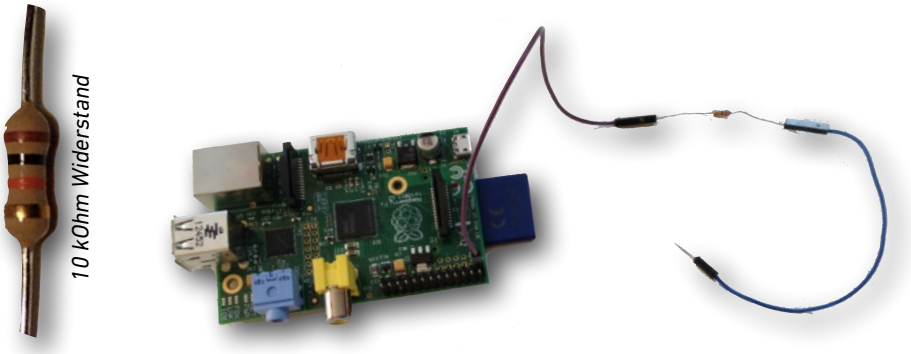
M28 - Einen Taster an den Raspberry Pi anschließen

Manche elektronische Bauteile lassen sich nicht einfach direkt an ein Kabel anschließen. Für unser nächstes Projekt benötigen wir daher ein sogenanntes „Steckbrett“.

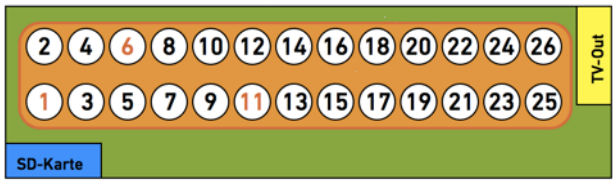
Stecke zunächst den Taster, den du bekommen hast, wie folgt in das Steckbrett:



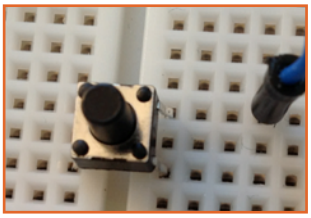
Anschließend nimmst du ein Jumper-Kabel mit zwei weiblichen Enden und schließt es an den **Pin 1** des Raspberry Pi. Nun musst du den Widerstand (10 kOhm) in das andere Ende des Kabels stecken. Verbinde das freie Ende des Widerstands mit einem weiteren Kabel (weiblich/männlich). Das Ergebnis sollte wie folgt aussehen:



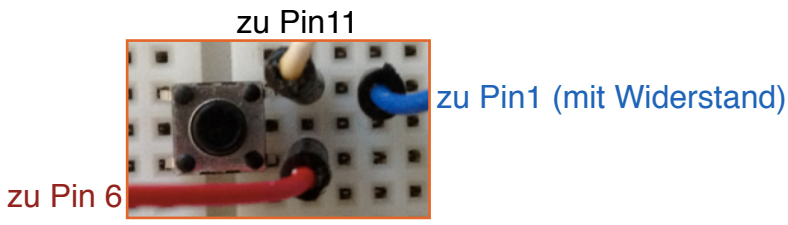
Hier ist zur Erinnerung noch einmal die Pin-Belegung:



Stecke nun das freie Kabelende an folgende Stelle ins Steckbrett:



Nun musst du zwei weitere Kabel anschließen. Eins an **Pin 6** und eins an **Pin 11**. Die anderen Enden kommen an folgende Stelle auf dem Steckbrett:



M29 - Den Taster programmieren

Um den Schalter benutzen zu können, musst du wieder ein Programm schreiben.

Gib dafür folgenden Befehl im Terminal ein:

```
nano Taster.py
```

Anschließend musst du folgenden Code eingeben:

```
# Import der Bibliothek für die GPIO-Schnittstelle
import RPi.GPIO as GPIO
# Löschen aller Belegungen dieser Schnittstelle
GPIO.cleanup()
# Nun müssen wir der Bibliothek sagen, wie wir die Pins nummerieren:
GPIO.setmode(GPIO.BOARD)
# Jetzt geben wir den Befehl, dass Pin 11 ein „Input“ sein soll:
GPIO.setup(11,GPIO.IN)
# Nun programmieren wir eine Schleife, in der wir...
while True:
    # ... alles, was von Pin 11 an Eingaben kommt, „Taster“ nennen.
    Taster = GPIO.input(11)
    # Wenn die Taste gedrückt ist, dann soll das Programm...
    if Taster == False:
        # „Klick!“ auf den Bildschirm schreiben.
        print (`Klick!`)
        # Und wenn die Taste losgelassen wird, ...
        while Taster == False:
            # ...soll nichts passieren.
            Taster = GPIO.input(11)
```

Nun kannst du die Datei speichern (STRG-X, dann „y“ drücken und anschließend mit Enter bestätigen) und mit folgendem Befehl aufrufen: `sudo python Taster.py`

Im nächsten Schritt werden wir mit diesem Programmcode den Taster auch in Minecraft benutzen.

M30 - Den Taster in Minecraft verwenden I

Wir können nun den Programmcode, den wir zum Bedienen des Tasters geschrieben haben, auch dazu verwenden, den Taster in Minecraft zu verwenden. Dazu kopieren wir zunächst das Programm in den Minecraft-Ordner mit folgendem Befehl:

```
mv Taster.py ./mcpi/api/python
```

Wechsle nun in das Verzeichnis mit dem Befehl:

```
cd mcpi/api/python
```

Und schaue mit folgendem Befehl, ob die Datei in dem Verzeichnis ist:

```
ls
```

Findest du die Datei „Taster.py“? Gut, dann editieren wir sie mit dem Befehl:

```
nano Taster.py
```

Und fügen folgenden Programmcode hinzu

```
# Zunächst importieren wir die Minecraft-Bibliothek...
import minecraft.minecraft as minecraft
# und erstellen eine Variable, die sich mit dem Spiel verbindet:
mc = minecraft.Minecraft.create()
import RPi.GPIO as GPIO
GPIO.cleanup()
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11,GPIO.IN)
while True:
        Taster = GPIO.input(11)
        if Taster == False:
                # Dann löschen wir die folgende Zeile
                print (`Klick!`)
                # Und ersetzen Sie durch den Befehl:
                mc.postToChat(`Klick!`)
                while Taster == False:
                        Taster = GPIO.input(11)
```

Nun kannst du die Datei speichern (STRG-X, dann „y“ drücken und anschließend mit Enter bestätigen). Starte dann Minecraft. Öffne anschließend die Konsole und starte das Programm mit dem Befehl: `sudo python Taster.py`

M31 - Den Taster in Minecraft verwenden II

Nun können wir die Taste auch dazu verwenden, andere Dinge in Minecraft zu tun, z. B. können wir die Position der Spielfigur verändern.

Dazu müssen wir wieder den Texteditor aufrufen, um die Datei „Taster.py“ verändern zu können:

```
nano Taster.py
```

Dann muss der Code wie folgt verändert werden:

```
import minecraft.minecraft as minecraft
mc = minecraft.Minecraft.create()

import RPi.GPIO as GPIO
GPIO.cleanup()
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11,GPIO.IN)
while True:
        Taster = GPIO.input(11)
        if Taster == False:
                # Dann löschen wir die folgende Zeile:
                mc.postToChat(`Klick!`)
                # Erstellen einer Variable, die die Position ermittelt:
                Pos = mc.player.getPos()
                # Konvertieren der Zahlen in das „integer“-Format:
                Pos = minecraft.Vec3(int(Pos.x),int(Pos.y),int(Pos.z))
                # Nun können wir die Spielfigur 20 Blöcke hochsetzen:
                mc.player.setPos(Pos.x,Pos.y + 20,Pos.z)
                while Taster == False:
                        Taster = GPIO.input(11)
```

Nun kannst du die Datei speichern (STRG-X, dann „y“ drücken und anschließend mit Enter bestätigen). Starte dann Minecraft. Öffne anschließend die Konsole und starte das Programm mit dem Befehl: `sudo python Taster.py`

M32 - Den Taster in Minecraft verwenden III

Wir können mit einem Tastendruck auch einen Block setzen, dazu müssen wir die Datei „Taster.py“ wie folgt verändern:

```
import minecraft.minecraft as minecraft
# Am Anfang müssen wir die Block-Bibliothek importieren
import minecraft.block as block
mc = minecraft.Minecraft.create()
import RPi.GPIO as GPIO
GPIO.cleanup()
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11,GPIO.IN)
while True:
        Taster = GPIO.input(11)
        if Taster == False:
                Pos = mc.player.getPos()
                Pos = minecraft.Vec3(int(Pos.x),int(Pos.y),int(Pos.z))
                # Dann löschen wir die folgende Zeile:
                mc.player.setPos(Pos.x,Pos.y + 20,Pos.z)
                # Und setzen vor uns einen Steinblock:
                mc.setBlock(Pos.x+1,Pos.y+1,Pos.z,block.STONE)
                while Taster == False:
                        Taster = GPIO.input(11)
```

Du kannst mit einem Tastendruck auch mehrere Blöcke gleichzeitig erschaffen, dafür löscht du die Zeile:

```
mc.setBlock(Pos.x+1,Pos.y+1,Pos.z,block.STONE)
```

Und schreibst stattdessen z.B.:

```
mc.setBlocks (Pos.x,Pos.y,Pos.z,Pos.x,Pos.y+10,Pos.z,block.STONE
```

Hier ist zum Abschluss noch eine Liste aller Blöcke, die du setzen kannst:

AIR	= Block(0)	COBWEB	= Block(30)	FURNACE_INACTIVE	= Block(61)
STONE	= Block(1)	GRASS_TALL	= Block(31)	FURNACE_ACTIVE	= Block(62)
GRASS	= Block(2)	WOOL	= Block(35)	DOOR_WOOD	= Block(64)
DIRT	= Block(3)	FLOWER_YELLOW	= Block(37)	LADDER	= Block(65)
COBBLESTONE	= Block(4)	FLOWER_CYAN	= Block(38)	STAIRS_COBBLESTONE	= Block(67)
WOOD_PLANKS	= Block(5)	MUSHROOM_BROWN	= Block(39)	DOOR_IRON	= Block(71)
SAPLING	= Block(6)	MUSHROOM_RED	= Block(40)	REDSTONE_ORE	= Block(73)
BEDROCK	= Block(7)	GOLD_BLOCK	= Block(41)	SNOW	= Block(78)
WATER_FLOWING	= Block(8)	IRON_BLOCK	= Block(42)	ICE	= Block(79)
WATER_STATIONARY	= Block(9)	STONE_SLAB_DOUBLE	= Block(43)	SNOW_BLOCK	= Block(80)
LAVA_FLOWING	= Block(10)	STONE_SLAB	= Block(44)	CACTUS	= Block(81)
LAVA_STATIONARY	= Block(11)	BRICK_BLOCK	= Block(45)	CLAY	= Block(82)
SAND	= Block(12)	TNT	= Block(46)	SUGAR_CANE	= Block(83)
GRAVEL	= Block(13)	BOOKSHELF	= Block(47)	FENCE	= Block(85)
GOLD_ORE	= Block(14)	MOSS_STONE	= Block(48)	GLOWSTONE_BLOCK	= Block(89)
IRON_ORE	= Block(15)	OBSIDIAN	= Block(49)	BEDROCK_INVISIBLE	= Block(95)
COAL_ORE	= Block(16)	TORCH	= Block(50)	STONE_BRICK	= Block(98)
WOOD	= Block(17)	FIRE	= Block(51)	GLASS_PANE	= Block(102)
LEAVES	= Block(18)	STAIRS_WOOD	= Block(53)	MELON	= Block(103)
GLASS	= Block(20)	CHEST	= Block(54)	FENCE_GATE	= Block(107)
LAPIS_LAZULI_ORE	= Block(21)	DIAMOND_ORE	= Block(56)	GLOWING_OBSIDIAN	= Block(246)
LAPIS_LAZULI_BLOCK	= Block(22)	DIAMOND_BLOCK	= Block(57)	NETHER_REACTOR_CORE	= Block(247)
SANDSTONE	= Block(24)	CRAFTING_TABLE	= Block(58)		
BED	= Block(26)	FARMLAND	= Block(60)		

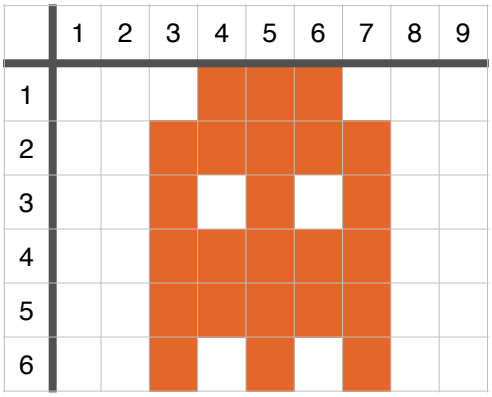
M33 - Station 1: Pixel-Art

Die Idee für dieses Programm kommt aus dem tollen Video-Podcast „Geek Gurl Diaries“. Es zeichnet das Pixel-Bild auf M34. Tippe dafür folgenden Code mit IDLE ab:

```
import pygame,
# Pygame ist eine Bibliothek mit Befehlen speziell für Spiele.
pygame.init()
# Mit diesem Befehl wird die Bibliothek initialisiert, also verfügbar gemacht.
screen = pygame.display.set_mode([225, 150])
# Dieser Befehl setzt die Größe des Bildes auf 225x150 Pixel.
r = pygame.Color("red")
w = pygame.Color(“white")
# Hier werden Abkürzungen für die Farben definiert, damit man weniger
# tippen muss.
data = [
        [ w, w, w, r, r, r, w, w, w ],
        [ w, w, r, r, r, r, r, w, w ],
        [ w, w, r, w, r, w, r, w, w ],
        [ w, w, r, r, r, r, r, w, w ],
        [ w, w, r, r, r, r, r, w, w ],
        [ w, w, r, w, r, w, r, w, w ]
        ]
# Das ist ein so genanntes Array, in dem die Daten für das Bild sind. „r“ steht
# für einen roten Pixel, „w“ für einen weißen.
for y, row in enumerate(data):
        for x, colour in enumerate(row):
# Mit dem „enumerate“-Befehl lassen sich die Pixel in der Reihenfolge des
# Arrays auf dem Bildschirm anzeigen.
                rect = pygame.Rect(x*25, y*25, 25, 25)
                screen.fill(colour, rect=rect)
# Dieser Code bestimmt die Größe eines Pixels und zeichnet das Bild.
pygame.display.update()
# Diese Zeile sorgt dafür, dass das Bild dauerhaft angezeigt wird.
while True:
        for event in pygame.event.get():
                if event.type == pygame.QUIT:
                        sys.exit()
# Dieser Code erlaubt es, das Programm zu beenden.
```

Wenn du fertig bist, speichere das Programm (Menüpunkt „File“ und dann „Save as“) und starte es mit dem Befehl „Run Module“ im Menüpunkt „Run“. Wenn alles funktioniert, versuche ein eigenes Bild mit drei oder mehr Farben zu malen und anschließend zu programmieren..

M34 - Station 1: Malvorlagen



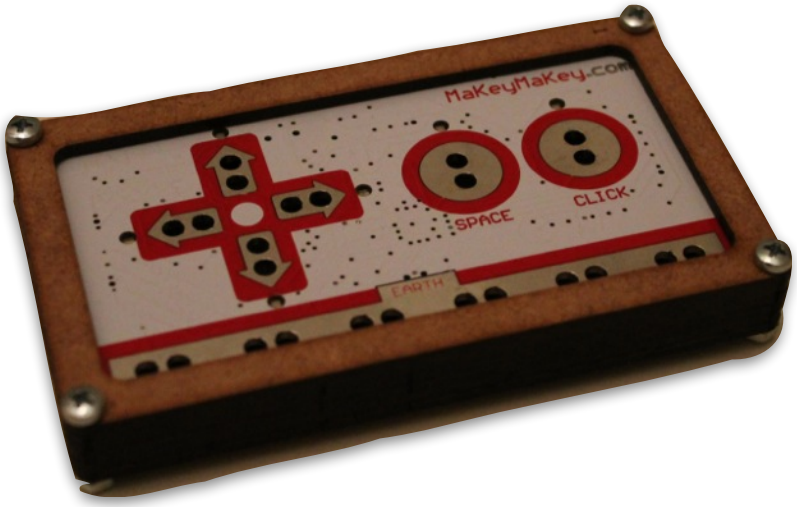
	1	2	3	4	5	6	7	8	9
1									
2									
3									
4									
5									
6									

	1	2	3	4	5	6	7	8	9
1									
2									
3									
4									
5									
6									

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1																			
2																			
3																			
4																			
5																			
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			

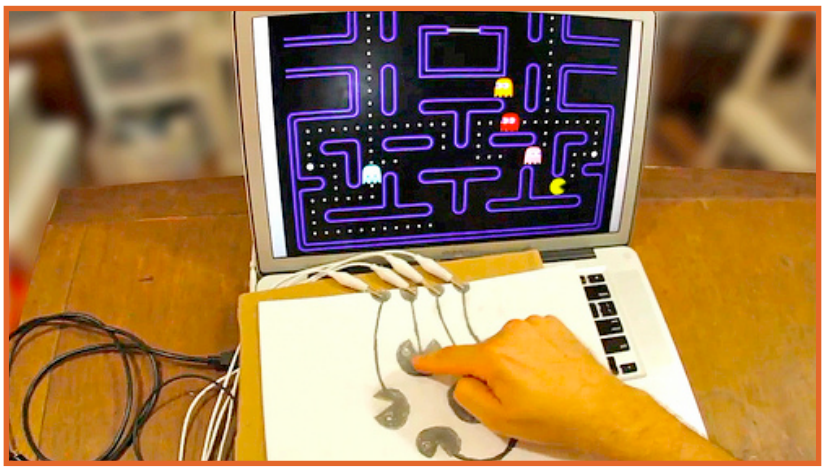
M35 - Station 2: Makey Makey

In dieser Station kannst du deinen eigenen Controller mit dem Makey Makey bauen. Der Makey Makey ist eine Platine, die du an den USB-Anschluss des Raspberry Pi anschließen kannst und die dann Funktionen der Tastatur bzw. Maus übernehmen kann.



Schließe dazu einfach eins der mitgelieferten Kabel am unteren Ende des Makey Makey an (da, wo „Earth“ steht) und halte das andere Ende in der Hand (so dass deine Haut das Metall berührt).

Nun kannst Du weitere Kabel einklemmen und an leitende Gegenstände (z. B. einen Apfel) „anschließen“. Wenn du dann den Apfel berührst, sendet der Makey Makey den entsprechenden Tastaturbefehl an den Computer. So kannst du eine eigene Computersteuerung bauen - probier es einfach einmal aus, z. B. indem Du einen Joystick mit einem Bleistift auf Papier malst!



Bildquelle: Kickstarter-Seite des Makey Makey

M36 - Station 3: Einen elektronischen Würfel löten

Hast Du schon einmal selbst gelötet? Nein? Dann wird es aber höchste Zeit. Für 4,- Euro kannst Du dir einen der Bausätze für einen elektronischen Würfel nehmen und ihn zusammenlöten.

Bevor Du anfängst, solltest Du allerdings das Comic lesen, das an dieser Station ausliegt. Darin findest Du alles, was Du übers Löten wissen muss. Du wirst sehen, dass es viel einfacher ist, als es zunächst aussieht.

Wenn Du Fragen hast, melde Dich einfach.

M37 - Station 4: Bomberman-Spielestation

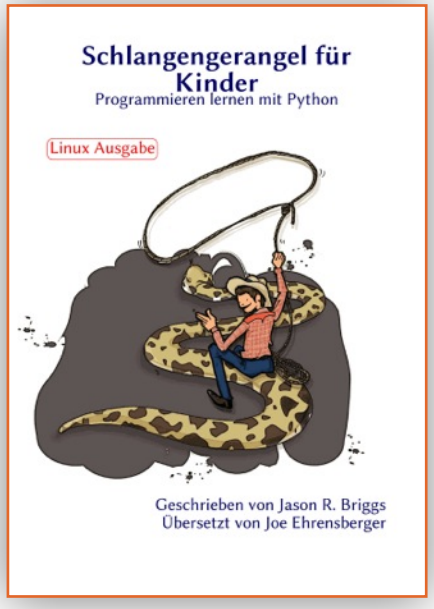
An dieser Station kannst Du Deinem Spieltrieb freien Lauf lassen. Schließ dazu einfach die selbstgebaute Bomberman-Konsole an einem Monitor an und stecke das Netzteil an.

Nun bootet der Raspberry Pi und startet automatisch die Emulations-Software. Wenn Du Dich im Menü befindest, wähle mit dem linken Joystick „Arcade“ und anschließend „Mame4All“. Nun kannst Du mit dem Zwei-Spieler-Knopf oben rechts ein neues Spiel beginnen.

Ihr könnt entweder mit- oder gegeneinander spielen. Tragt Euch am Ende auf jeden Fall in die Highscore-Liste ein, denn wer am Ende des Schuljahres die höchste Punktzahl hat, gewinnt einen Preis!

M38 - Station 5: Die Raspberry Pi-Bibliothek

An dieser Station hast Du ausgiebig Zeit, um Dir ein eigenes Projekt auszudenken, das Du mit Deinem Raspberry Pi machen kannst. Wirf dazu in dieser Stunde einen Blick in die Zeitschriften und Bücher, die Du bekommen hast. Darin findest Du jede Menge Ideen und Anregungen für tolle Bastelprojekte und jede Menge interessante Programme. Vielleicht findest Du ein Projekt, dass Du am Ende des Halbjahres bauen möchtest.



M39 - Station 6: Einen Joystick anschließen

An dieser Station kannst du einen Joystick an deinen Raspberry Pi anschließend und damit ein Spiel steuern. Dazu musst du zunächst alle Kabel des Joysticks an den GPIO-Port anschließen. Richte dich dabei nach der folgenden Beschriftung:

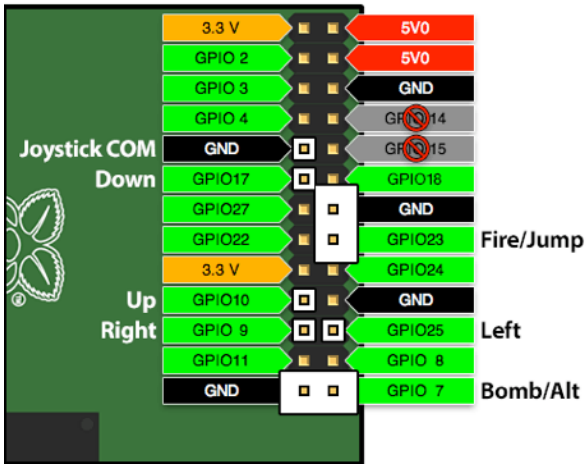


Illustration von Philipp Burgess (http://learn.adafruit.com/assets/8823)

Die Kabel, die mit „Ground“ beschriftet sind, müssen an einen mit „GND“ beschrifteten Pin befestigt werden. Schließe die beiden Knöpfe an GPIO23 und GPIO7 an. Schließe den Joystick so an, wie du es auf dem Bild siehst (Kabel „links“ an GPIO25, „rechts“ an GPIO9, „oben“ an GPIO10 und „unten“ an GPIO17).

Nun benötigst du das Programm „retrogame“, das du im Verzeichnis „retrogame“ findest. Wechsle mit dem Befehl cd retrogame in dieses Verzeichnis und starte das Programm mit dem Befehl „sudo ./retrogame“. Nun kannst du z. B. das vorinstallierte Tetris-Spiel mit dem Joystick steuern. Wenn du die Tastenbelegungen ändern möchtest, gib den Befehl „nano retrogame.c“ ein. Ab Zeile 77 findest du die Tastaturbelegungen. Wenn du jetzt z. B. die Zeile:

```
{ 23,    KEY_LEFTCTRL },
```

abänderst in:

```
{ 23,    KEY_A },
```

dann bewirkt ein Druck auf den Joystick-Knopf das Gleiche wie ein Druck auf die A-Taste der Tastatur. Wenn du eine Änderung in der Datei vornimmst, musst du sie speichern und anschließend folgenden Befehl eingeben:

```
make
```

M40 - Station 7: Das Raspberry Pi-Kameraboard I

Für den Raspberry Pi kannst Du für ca. 20,- Euro eine Kamera kaufen. Obwohl die Kamera unscheinbar aussieht, kann sie Fotos mit 5-Megapixeln (1.592 x 1.944 Pixel) und Videos mit 30 Bildern pro Sekunde aufnehmen.

Ein Exemplar findest Du an dieser Station.

Um sie an den Raspberry Pi anzuschließen, benötigst Du den so genannten „CSI-Anschluss“, der sich direkt hinter dem Netzwerkanschluss befindet. Hebe dazu vorsichtig die Plastikabdeckung des Anschlusses an, schließe das Flachbandkabel an (die blaue Seite muss zum Netzwerkanschluss zeigen) und schließe die Plastikabdeckung wieder. Boote nun den Raspberry Pi mit der SD-Karte, die an der Station ausliegt.

Probiere nun einmal die folgenden Befehle aus:

```
raspistill -o image.jpg
```

Nun solltest Du für einige Sekunden ein Vorschaufenster auf dem Raspberry Pi sehen, das nach wenigen Sekunden wieder verschwindet. Das Bild kannst du mit folgendem Befehl anschauen:

```
fbi image.jpg
```

Nun weißt Du, dass die Kamera funktioniert, und kannst ein kleines Programm schreiben, das ein Bild aufnimmt. Gib dazu den Befehl:

```
nano camera.py
```

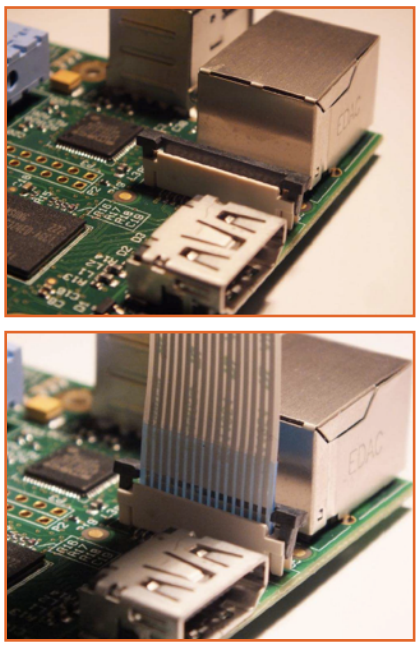
ein und tippe den folgenden Code ab:

```
import time
import picamera

with picamera.PiCamera() as camera:
    camera.start_preview()
    time.sleep(5)
    camera.capture('/home/pi/image.jpg')
    camera.stop_preview()
```

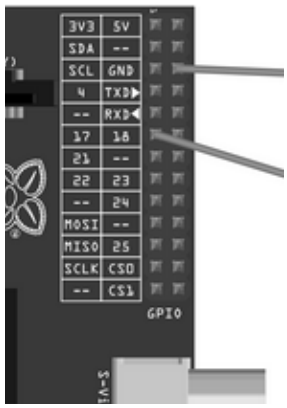
Schließe den nano-Editor mit STRG-X, bestätige, dass Du die Datei speichern willst und drücke Enter. Nun kannst du die Datei aufrufen mit dem Befehl:

```
sudo python camera.py
```



Bildquelle: The Mag Pi, Ausgabe 14, S. 5

M41 - Station 7: Das Raspberry Pi-Kameraboard II



Bildquelle: GitHub

Schließe nun den Knopf an, der an der Station ausliegt. Die beiden Kabel müssen an die GPIO-Pins angeschlossen werden, die Du links auf dem Bild siehst. Dabei ist es egal, welches der beiden Kabel wo angeschlossen wird.

Ändere dann das Programm, das du gerade geschrieben hast, wie folgt um:

```
import time
import picamera
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.IN, GPIO.PUD_UP)

with picamera.PiCamera() as camera:
  camera.start_preview()
  GPIO.wait_for_edge(17, GPIO.FALLING)
  camera.capture('/home/pi/image.jpg')
  camera.stop_preview()
```

Nun kannst du per Knopfdruck ein Bild aufnehmen. Wenn du das Programm noch weiter abänderst, kannst du sogar ein Video aufnehmen. Ändere dafür den letzten Teil des Programms wie folgt:

```
with picamera.PiCamera() as camera:
  camera.start_preview()
  GPIO.wait_for_edge(17, GPIO.FALLING)
  camera.start_recording('/home/pi/video.h264')
  time.sleep(1)
  GPIO.wait_for_edge(17, GPIO.FALLING)
  camera.stop_recording()
  camera.stop_preview()
```

Abspielen kannst Du das Video mit folgendem Befehl:

```
omxplayer video.h264
```

Mit diesem Befehl kannst Du abschließend ein paar Effekte sehen:

```
raspistill -d 500 -t 60000
```

M42 - Station 8: Sonic Pi-Experimente I

Sonic-Pi ist eine sehr interessante Programmiersprache, die Musik produziert. Du benötigst also den Kopfhörer, der an der Station ausliegt, und die SD-Karte. Wenn Du das Programm startest, siehst Du links einen „Workspace“, indem Du Dein Programm schreiben kannst. Beginne mit dem Befehl:

```
play 60
```

und klicke auf den „Play“-Knopf oben links. Wenn du stattdessen:

```
play 60
```

eingibst, siehst du, dass ein Fehler angezeigt wird. Schreibe nun folgenden Code:

```
play 60
play 67
play 69
```

Du hörst, dass die Töne sehr schnell hintereinander abgespielt werden. Füge also Pausen hinzu und wiederhole die Tonfolge einmal mit einer Schleife:

```
2.times do
  play 60
  sleep 0.5
  play 62
  sleep 0.5
  play 64
  sleep 0.5
  play 60
  sleep 0.5
end
```

Wenn du über diesen Code folgenden Befehl schreibst:

```
with_synth „fm"
```

änderst Du das Instrument, das den Ton abspielt. Es gibt auch andere Instrumente, nämlich: „pretty_bell“, „dull_bell“, „beep“ und „saw_beep“. Nun kannst du das folgende Programm von der Sonic Pi-Website abtippen:

```
#Hiermit setzt Du das Tempo der Melodie fest.
with_tempo 350
2.times do
   play_pattern [40,25,45,25,25,50,50]
   play_pattern [25,50,25,30,35,40,45,50]
   play_pattern [25,50,25,30,35,40,45,50].reverse
end
2.times do
   with_synth „saw_beep"
```

-Fortsetzung auf der nächsten Seite-

M43 - Station 8: Sonic Pi-Experimente II

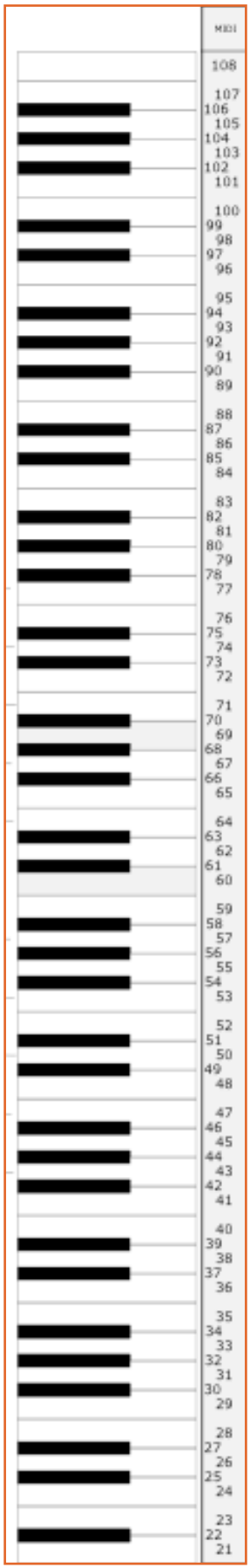
```
    play_pattern [25,50,25,30,35,40,45,50].shuffle
    play_pattern [25,50,25,30,35,40,45,50].reverse
  end
  # Nun spielen wir mehrere Melodien auf einmal.
  in_thread do
   with_synth "saw_beep"
   10.times do
    if rand < 0.5
      play 37
    else
      play 49
    end
    sleep 2
   end
  end
  in_thread do
   with_synth "pretty_bell"
   20.times do
    play 49
    sleep 1
   end
  end
```

Nun kannst du versuchen, eine eigene Melodie zu programmieren. Auf der rechten Seite siehst du, welche Taste auf dem Klavier welcher Zahl entspricht. Vielleicht kennst Du ja eine einfache Melodie, die Du mit Sonic Pi programmieren kannst, z. B, diese hier:

```
3.times do
 play 55
 sleep 0.5
 play 55
 sleep 0.3
 play 55
 sleep 0.1
 play 55
 sleep 0.25
 play 55
 sleep 0.1
 play 55
 sleep 0.25
 play 57
 sleep 0.25
 play 59
 sleep 0.25
end
```

Der Code geht hier weiter...

```
play 60
sleep 0.5
play 59
sleep 0.5
play 57
sleep 0.5
play 59
sleep 0.5
```



Bildquelle: Wikimedia Commons

M44 - Station 9: Eine Pong-Uhr programmieren

Mit dem Raspberry Pi lassen sich nicht nur Spielkonsolen emulieren, man kann aus ihm auch eine originelle Uhr im Design des Spielklassikers „Pong“ basteln. Alles, was man dazu braucht, ist ein kleiner Monitor und evtl. ein selbstgebasteltes Gehäuse.

Da der Raspberry Pi aus Kostengründen keine eingebaute Uhr mit eigener Batterie hat, muss man dafür nach dem Einschalten zunächst die Uhrzeit einstellen. Dies geschieht in der Kommandozeile mit dem Befehl:

```
sudo date 02111500
```

Die Zahl in dem Beispiel würde das Datum auf den 11. Februar und die Uhrzeit auf 15.00 Uhr setzen. Mit „sudo“ ändert man übrigens in Linux Einstellungen, die den gesamten Computer betreffen. Der Befehl ist also eine Art Schutz davor, dass der Benutzer aus Versehen gravierende Änderungen am System vornimmt. Man sollte „sudo“ daher nur benutzen, wenn man genau weiß, was man tun will.

Anschließend muss man das Programm „pypong-clock“ herunterladen, das unter der URL https://bitbucket.org/donclark/pypong-clock zu finden ist.
Man kann das Programm anschließend noch ein wenig anpassen, indem man die Datei „pypong-clock.py“ modifiziert. Dies geschieht mit dem Befehl:

```
nano pypong-clock.py
```

In der Datei kann man z. B. folgende Einstellungen ändern:

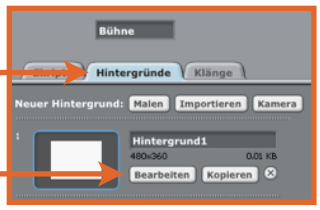
```
RESOLUTION = (896, 416)      # Die Auflösung, in der das Programm läuft
WHITE = (255, 255, 255)      # Die RGB-Farbe der Spieler und des Netzes
BLACK = (0, 0, 0)            # Die RGB-Farbe des Spielfelds
PADDLE_HEIGHT = 60           # Die Höhe der Spielfiguren in Pixeln
PADDLE_WIDTH = 14            # Die Breite der Spielfiguren in Pixeln
BALL_WIDTH = 28              # Höhe und Breite des Balles
BALL_HEIGHT = 28
```

Wenn man zudem den Eintrag `now.strftime(„%I")` durch `now.strftime(„%H")` ersetzt, wird die Uhrzeit im 24-Stunden-Format angezeigt. Gestalte nun deine ganz persönliche Pong-Uhr!

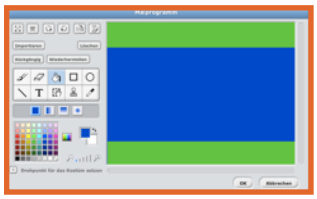
M45 - Station 10: Ein Spiel mit Scratch programmieren!

In dieser Station programmierst du ein kleines Spiel mit Scratch. Durch einen Doppelklick auf den weißen Hintergrund kann man diesen bearbeiten.

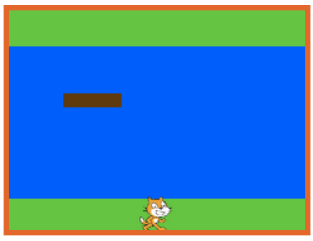
Dazu muss man in dem Reiter „Hintergründe“ auf „Bearbeiten“ klicken.



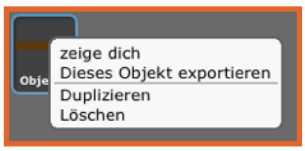
Mit der Funktion „Rechteck“ lassen sich nun zwei grüne Balken zeichnen. Die restliche Fläche kann mit der Funktion „Flächen füllen“ blau gemacht werden. Anschließend den Editor mit „OK“ beenden.



Nun kann man mit dem Knopf die Baumstämme zeichnen. Dazu muss man lediglich ein braunes Rechteck zeichnen und mit „OK“ bestätigen. Mit der Taste lässt sich die bereits vorgegebene Katze so verkleinern, dass sie auf den unteren grünen Streifen passt. Anschließend kann man die Objekte per drag and drop verschieben.



Als Nächstes muss der Baumstamm viermal kopiert werden. Dafür genügt jeweils ein Rechtsklick auf das Objekt und die Auswahl des Befehls „Duplizieren“.



Nach einem Doppelklick auf die Katze kann man nun den eigentlichen Code eingeben. Dazu zieht man die jeweiligen Code-Blöcke untereinander. Der folgende Code sorgt dafür, dass sich die Katze durch einen Druck auf die Pfeiltasten nach links, rechts bzw. unten bewegt und die Figur „springt“, wenn man die Leertaste drückt:

M46 - Station 10: Ein Spiel mit Scratch programmieren II

Dieser Code setzt die Spielfigur an ihren Startpunkt, dreht sie um 90° und überprüft, ob sie sich auf einem Baumstamm oder im Wasser befindet:

Nun fehlt noch der Code für die Baumstämme. Dieser ist bei allen vier Objekten ähnlich, daher kann der Code mit dem Stempel-Knopf kopiert und in das jeweilige Objekt eingefügt werden. Dann müssen nur noch die eingekreisten Werte verändert werden. Viel Spaß beim Spielen!