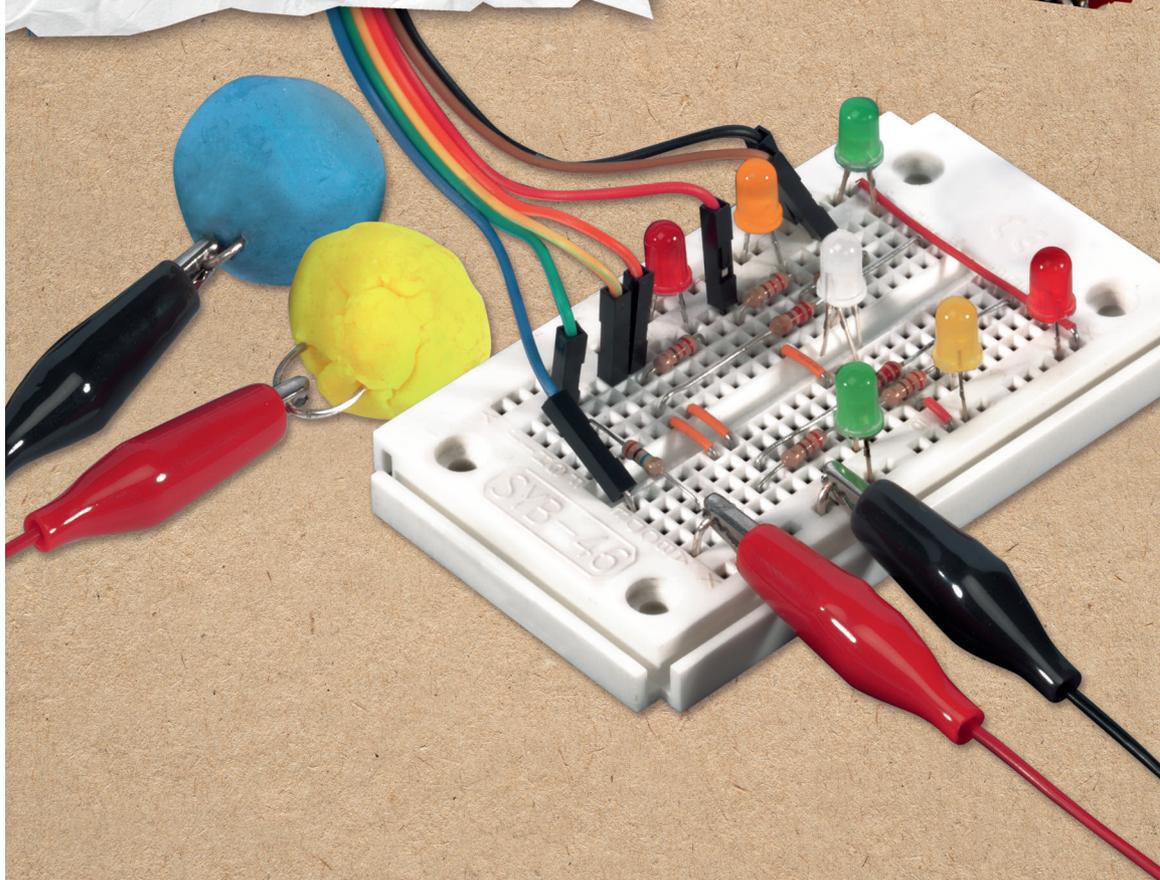


# Raspberry Pi-Projekte für Young Makers





# Raspberry Pi-Projekte für Young Makers



FRANZIS

POWERED BY  
**CONRAD**

## Liebe Kunden!

Dieses Produkt wurde in Übereinstimmung mit den geltenden europäischen Richtlinien hergestellt und trägt das CE-Zeichen. Der bestimmungsgemäße Gebrauch ist in der beiliegenden Anleitung beschrieben.



Bei jeder anderen Nutzung oder Veränderung des Produktes sind allein Sie für die Einhaltung der geltenden Regeln verantwortlich. Bauen Sie die Schaltungen deshalb genau so auf, wie es in der Anleitung beschrieben wird. Das Produkt darf nur zusammen mit dieser Anleitung weitergegeben werden.



Das Symbol der durchkreuzten Mülltonne bedeutet, dass dieses Produkt getrennt vom Hausmüll als Elektroschrott dem Recycling zugeführt werden muss. Wo Sie die nächstgelegene kostenlose Annahmestelle finden, sagt Ihnen Ihre kommunale Verwaltung.

## Achtung! Augenschutz und LEDs:

Blicken Sie nicht aus geringer Entfernung direkt in eine LED, denn ein direkter Blick kann Netzhautschäden verursachen! Dies gilt besonders für helle LEDs im klaren Gehäuse sowie in besonderem Maße für Power-LEDs. Bei weißen, blauen, violetten und ultravioletten LEDs gibt die scheinbare Helligkeit einen falschen Eindruck von der tatsächlichen Gefahr für Ihre Augen. Besondere Vorsicht ist bei der Verwendung von Sammellinsen geboten. Betreiben Sie die LEDs so wie in der Anleitung vorgesehen, nicht aber mit größeren Strömen.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

Alle in diesem Buch vorgestellten Schaltungen und Programme wurden mit der größtmöglichen Sorgfalt entwickelt, geprüft und getestet. Trotzdem können Fehler im Buch und in der Software nicht vollständig ausgeschlossen werden. Verlag und Autor haften in Fällen des Vorsatzes oder der groben Fahrlässigkeit nach den gesetzlichen Bestimmungen. Im Übrigen haften Verlag und Autor nur nach dem Produkthaftungsgesetz wegen der Verletzung des Lebens, des Körpers oder der Gesundheit oder wegen der schuldhaften Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht ein Fall der zwingenden Haftung nach dem Produkthaftungsgesetz gegeben ist.

**Autor:** Christian Immler

**Art & Design:** [www.ideehoch2.de](http://www.ideehoch2.de)

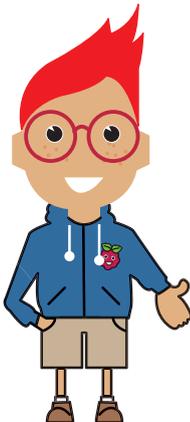
**Satz:** DTP-Satz A. Kugge, München



# Inhaltsverzeichnis



<b>Raspberry Pi vorbereiten.....</b>	<b>6</b>
Betriebssystem installieren.....	12
Der erste Start des Raspberry Pi.....	17
Die Teile im Paket.....	20
LEDs.....	24
RGB-LEDs.....	26
Knete, Draht und Krokodilklemmen-Kabel.....	30
<b>1 Elektronik steuern mit Knetkontakten.....</b>	<b>34</b>
Es geht los!.....	41
So funktioniert das Programm.....	49
<b>2 Fußgängerampel.....</b>	<b>56</b>
So funktioniert das Programm.....	59
<b>3 RGB-LED mit Knetkontakten steuern.....</b>	<b>62</b>
So funktioniert das Programm.....	65
<b>4 LED-Würfel.....</b>	<b>66</b>
So funktioniert das Programm.....	72
<b>5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern.....</b>	<b>78</b>
Der Raspberry Pi erzeugt Töne.....	81
So funktioniert das Programm.....	82



# Raspberry Pi vorbereiten

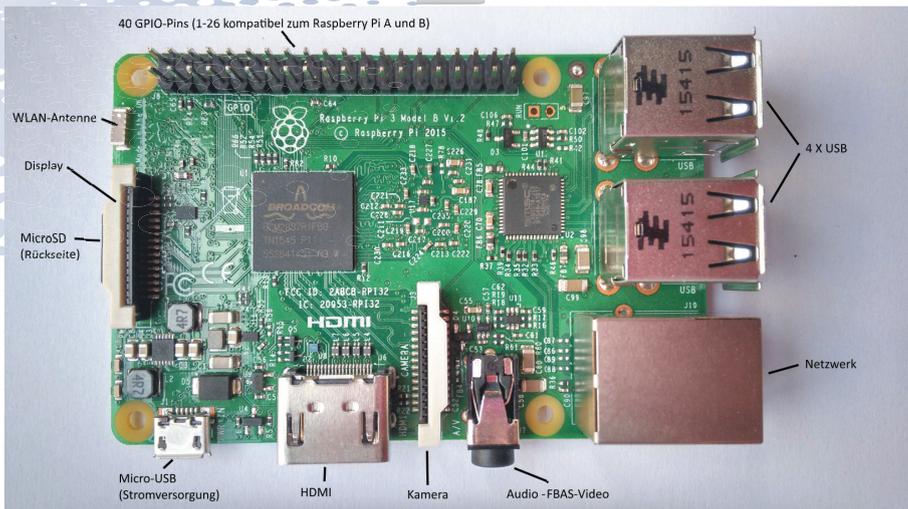
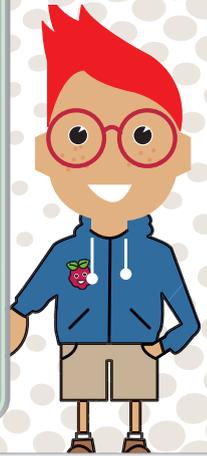


Der Raspberry Pi ist – auch wenn es auf den ersten Blick gar nicht so aussieht – ein vollwertiger Computer. Du kannst damit Texte schreiben, ins Internet gehen, Spiele spielen oder Filme ansehen. Vieles geht etwas langsamer, als man es von modernen PCs gewohnt ist, dafür ist der Raspberry Pi ja auch viel kleiner und vor allem billiger als ein PC.

Und du kannst den Raspberry Pi programmieren – das geht sogar viel leichter als auf dem PC, weil alle dafür notwendigen Werkzeuge bereits

## DER NAME RASPBERRY PI

*Raspberry* ist das englische Wort für Himbeere. Schon früher wurden Computer nach Früchten benannt, wie z. B. Apple, Apricot, Blackberry. *Pi* steht für Python Interpreter, eine wichtige Programmiersprache auf dem Raspberry Pi. Zusammen genommen ergibt sich ein Name, der wie das englische Wort für Himbeerkuchen *raspberry pie*, klingt.



Die Anschlüsse des Raspberry Pi 3



vorinstalliert sind und einfachere, verständliche Programmiersprachen verwendet werden. Python, das dem Raspberry Pi seinen Namen gegeben hat, und vor allem Scratch sind deutlich leichter zu erlernen als C++ und Java, die sich unter Profiprogrammierern auf dem PC durchgesetzt haben.

Im Laufe der noch sehr kurzen Geschichte des Raspberry Pi wurden verschiedene Modelle entwickelt. Für unsere Projekte kannst du alle aktuellen Raspberry-Pi-Modelle B+, Pi 2 und Pi 3 verwenden.

Wie für einen „großen“ PC braucht man auch für den Raspberry Pi einiges Zubehör, um ihn wirklich benutzen zu können.

## STROM ÜBER MICRO-USB-HANDYLADEGERÄT

Der Raspberry Pi benötigt nur wenig Strom und muss deshalb nicht wie ein PC direkt an eine Steckdose angeschlossen werden. Zur Stromversorgung reicht jedes moderne Handynetzteil.

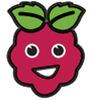
Ältere Ladegeräte aus den Anfangszeiten der USB-Ladetechnik sind noch zu schwach. Das Netzteil sollte mindestens 1.000 mA liefern, besser 2.000 mA. Der eingebaute Leistungsregler verhindert ein „Durchbrennen“ bei zu starken Netzteilen. Wie viel Strom ein Handyladegerät liefert, steht ganz klein gedruckt auf den Geräten.

### SO ERKENNT MAN EIN ZU SCHWACHES NETZTEIL

Wenn der Raspberry Pi zwar bootet, sich dann aber keine Mausbewegung erkennen lässt oder das System nicht auf Tastatureingaben reagiert, deutet das auf eine zu schwache Stromversorgung hin. Auch wenn kein Zugriff auf angeschlossene USB-Sticks oder Festplatten möglich ist, solltest du ein stärkeres Netzteil verwenden.



# Raspberry Pi vorbereiten



Dieses Netzteil liefert 2A = 2000 mA; das ist völlig ausreichend für aktuelle Raspberry-Pi-Modelle.

## MONITOR

Natürlich brauchst du einen Monitor, um auch zu sehen, was du tust. Der Raspberry Pi kann per HDMI-Kabel an moderne Monitore oder Fernseher angeschlossen werden. Zum Anschluss an Computermonitore mit DVI-Anschluss gibt es spezielle HDMI-Kabel oder Adapter. Es gibt auch Adapter zum Anschluss an VGA-Monitore, die allerdings vergleichsweise teuer sind.

Steht kein HDMI-Monitor zur Verfügung, kann der Raspberry Pi mit einem analogen FBAS-Videokabel an einen klassischen Fernseher angeschlossen werden, wobei die Bildschirmauflösung allerdings sehr niedrig ist. Die grafische Oberfläche lässt sich in analoger Fernsehauf-

lösung nur mit Einschränkungen bedienen, Scratch fast gar nicht. Deshalb ist diese Methode nicht zu empfehlen.

## TASTATUR UND MAUS

Am Raspberry Pi kann jede gängige Tastatur mit USB-Anschluss genutzt werden. Kabellose Tastaturen funktionieren manchmal nicht, da sie zu viel Strom oder gar spezielle Treiber benötigen.

Einige USB-Tastaturen haben einen weiteren USB-Anschluss für die Maus. Dadurch ersparst du dir am Raspberry Pi einen USB-Anschluss, was besonders beim Raspberry Pi A+ wichtig ist, der nur einen USB-Anschluss hat. Die anderen aktuellen Raspberry-Pi-Modelle haben vier USB-Anschlüsse, die immer ausreichen sollten.

## NETZWERKKABEL

Möchtest du mit dem Raspberry Pi ins Internet, brauchst du ein Netzwerkkabel, um ihn mit deinem Router zu verbinden. Für die Experimente in diesem Paket benötigst



Raspberry Pi mit angeschlossenen Kabeln: USB-Tastatur (transparent), Netzwerk (gelb), HDMI-Monitor (schwarz) und Netzteil (weiß)

du keine Internetverbindung. Der Raspberry Pi 3 hat ein eingebautes WLAN-Modul. Hier ist kein Netzkabel für eine Internetverbindung nötig.

## SPEICHERKARTE

Der Raspberry Pi hat keine Festplatte, stattdessen wird eine Speicherkarte verwendet. Sie enthält das Betriebssystem. Eigene Daten

und installierte Programme werden ebenfalls darauf gespeichert. Die Speicherkarte muss mindestens 4 GB groß sein. Auf der Rückseite befindet sich der Steckplatz für eine MicroSD-Karte. MicroSD-Karten sind deutlich stabiler als frühere Speicherkartenformate und rasten im Stecksocket fest ein. Um sie wieder herauszubekommen, drücke die Speicherkarte noch einmal leicht in den Stecksocket, dann löst sich die Halterung, und eine kleine Feder schiebt die Karte heraus.





### IMMER +3,3 V VERWENDEN

Verwende für elektronische Experimente immer die +3,3-V-Anschlüsse des Raspberry Pi und nicht die +5-V-Anschlüsse. Würde 5-V-Spannung durch eine elektronische Schaltung wieder zurück auf einen GPIO-Pin kommen, würde dies den Raspberry Pi beschädigen. Die +5-V-Anschlüsse sind nur für externe Geräte wie zum Beispiel Servomotoren gedacht und nicht für Logiksignale.

gung und andere Zwecke fest eingerichtet. Es gibt je zwei Pins mit +3,3 V und +5 V Spannung sowie insgesamt acht Pins mit Masseanschluss.

## DER RASPBERRY PI A+

Neben den bekannten Modellen B+, Pi 2 und Pi 3 gibt es auch den etwas kleineren Raspberry Pi A+ mit nur einem USB-Port, ohne Netzwerkanschluss und mit etwas vereinfachter Hardwareleistung. Dieses Modell verfügt aber über die gleiche GPIO-Schnittstelle und ist billiger als die anderen. Man kann es mit

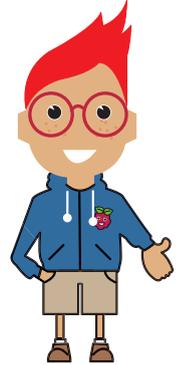
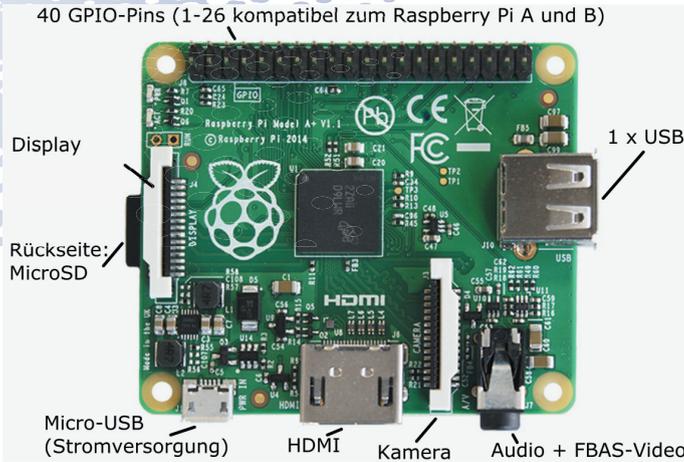
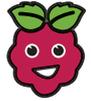
etwas Glück schon für unter 20 Euro bekommen.

Da der Raspberry Pi A+ keinen Netzwerkanschluss hat, kommst du nur mit zusätzlicher Hardware ins Internet, was aber für unsere Projekte gar nicht nötig ist. Allerdings benötigst du eine Tastatur mit USB-Anschluss für die Maus oder einen USB-Hub, um Tastatur und Maus gleichzeitig anschließen zu können.

Die aktuelle Version des Raspbian-Betriebssystems *Jessie* läuft auf dem B+, Pi 2, Pi 3 und dem A+ gleichermaßen.



# Raspberry Pi vorbereiten



Der Raspberry Pi A+

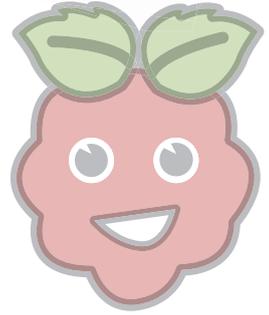
## BETRIEBSSYSTEM INSTALLIEREN

Der Raspberry Pi verwendet Linux und kein Windows, aber das sieht auch nicht viel anders aus. Linux hat den großen Vorteil, dass es längst nicht so hardwarehungrig ist wie Windows. Außerdem ist Linux „freie Software“. Das heißt, jeder darf Linux nutzen, für seine eigenen Zwecke anpassen und auch beliebig weitergeben, ohne Seriennummern verwenden oder sich mit Lizenzen herumärgern zu müssen.

Wir haben für dieses Paket ein spezielles Linux vorbereitet, das bereits alles Nötige enthält, damit man sofort mit den Experimenten loslegen kann. Du findest alle notwendigen Dateien auf der DVD.

## SPEICHERKARTE VORBEREITEN

Da der Raspberry Pi selbst noch nicht booten kann, bereiten wir die Speicherkarte auf dem PC vor. Dazu brauchst du einen Kartenleser am



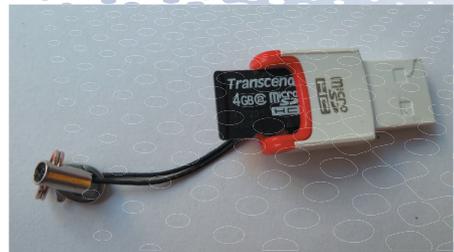
## INFO



Unser Linux für den Raspberry Pi basiert auf dem original Raspbian-Betriebssystem Jessie, das die Raspberry-Pi-Stiftung ([www.raspberrypi.org](http://www.raspberrypi.org)) anbietet. Bei dieser Linux-Version ist die Programmiersprache Scratch, die wir verwenden, auch schon dabei. Wir haben zusätzlich die verwendeten Programme für die Projekte vorinstalliert und alles komplett auf Deutsch eingestellt. Unsere Raspbian-Version ist nur etwa 3,7 GB groß und passt somit auf jede 4-GB- Speicherkarte – auf größere natürlich auch.

gehend ungebrauchlich sind. Smartphones und Tablets verwenden üblicherweise auch MicroSD-Karten. Bei im PC eingebauten Kartenlesern ist für MicroSD-Karten meistens ein Adapter erforderlich, der bei neuen MicroSD-Karten oft mitgeliefert wird.

PC. Dieser kann fest eingebaut oder per USB angeschlossen werden. Wenn dein PC noch keinen Kartenleser hat, besorge dir am besten einen kleinen Kartenleser in USB-Stick-Form. Die einfachen Modelle, die nur MicroSD-Karten lesen und bei manchen Speicherkarten sogar mitgeliefert werden, reichen völlig aus, da wir genau dieses Kartenformat für den Raspberry Pi brauchen und andere Kartentypen inzwischen weit-



Wenn du eine fabrikneue Speicherkarte verwendest oder eine, die bisher nur in Smartphones oder Tablets genutzt wurde, ist sie bereits vom Hersteller optimal vorformatiert,

# Raspberry Pi vorbereiten



und du brauchst dich um die Formatierung nicht zu kümmern.

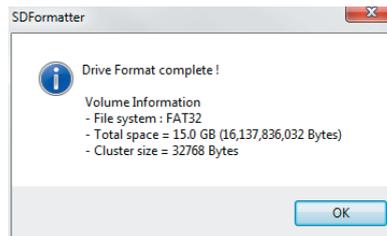
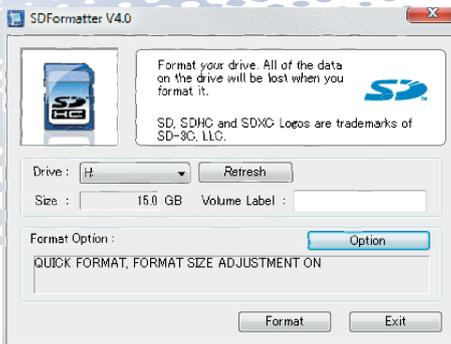
- Hast du die Speicherkarte früher schon einmal für einen Raspberry Pi oder einen anderen Linux-Computer verwendet, muss sie neu formatiert werden, damit der Kartenleser im PC die Größe der Speicherkarte richtig erkennt.

## ACHTUNG

Egal ob neu formatiert oder nicht: Bei der Installation des Betriebssystems werden alle Daten, die sich auf der Speicherkarte befinden, unwiderruflich gelöscht. Also vorher alles sichern, was wichtig ist!

Um die Speicherkarte neu zu formatieren, lade dir das kostenlose Programm **SD Formatter** der SD Association unter [www.sdcard.org/downloads/formatter\\_4](http://www.sdcard.org/downloads/formatter_4) herunter.

Das Programm erkennt die Speicherkarte im Kartenleser automatisch und zeigt ihre Größe an. Sollte



Das SD-Formatter-Tool unter Windows in Aktion



die Speicherkarte Partitionen aus einer früheren Linux-Installation enthalten, zeigt der SD Formatter nicht die vollständige Größe an. Klicke in diesem Fall auf **Option** und wähle die Formatierungsoption **ON** bei **Format Size Adjustment**. Nach einem Klick auf **Format** wird die Speicherkarte in ihrer vollen Größe formatiert.

## IMAGE-DATEI AUF DIE SPEICHERKARTE ÜBERTRAGEN

Auf der DVD findest du eine Image-Datei des Raspbian-Betriebssystems, die jetzt auf die Speicherkarte übertragen werden muss. Die Datei kann nicht einfach kopiert werden, sie enthält ein Abbild der Linux-Partition mit dem Bootblock und der kompletten Verzeichnisstruktur des Raspbian-Betriebssystems, das auf der Speicherkarte bootfähig installiert wird.



Der Startbildschirm der DVD

# Raspberry Pi vorbereiten

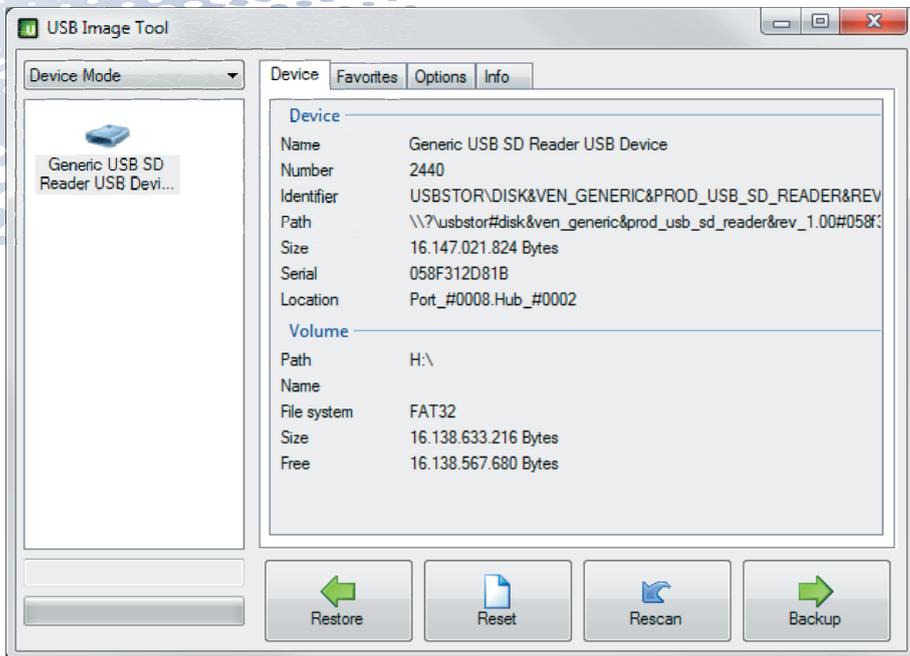


Zur Übertragung solcher Image-Dateien gibt es ein Spezialprogramm, das **USB-Image-Tool** von [www.alex-page.de](http://www.alex-page.de). Du brauchst es nicht herunterzuladen, es ist auf der DVD vorhanden.

1 Stecke die Speicherkarte in den Kartenleser und starte das USB-Image-Tool von der DVD. Dieses Programm braucht nicht vorher auf dem PC installiert zu werden.

2 Klicke oben links unter **Device Mode** auf das Symbol der Speicherkarte. Rechts tauchen dann jede Menge technischer Daten auf sowie der Laufwerksbuchstabe, unter dem Windows die Speicherkarte anspricht. Sollte oben links statt **Device Mode** der **Volume Mode** ausgewählt sein, schalte auf **Device Mode** um.

3 Klicke dann auf **Restore** und wähle die Image-Datei **raspberrypikids.img** von der DVD aus.



Das USB-Image-Tool von der DVD



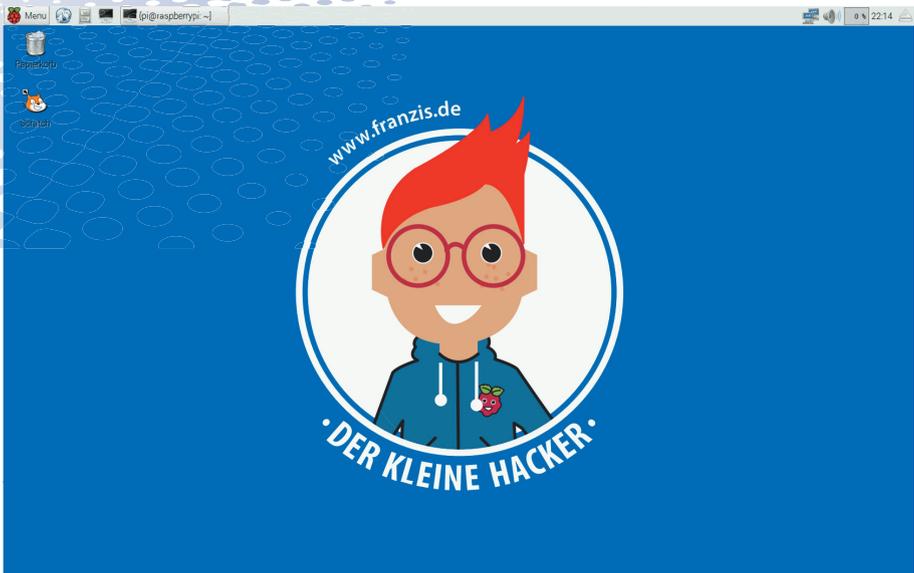
 Starte danach den Kopiervorgang, der einige Minuten dauern kann. Nach Abschluss ist die Speicherkarte fertig vorbereitet, und du kannst sie aus dem PC herausnehmen.

## DER ERSTE START DES RASPBERRY PI

Jetzt ist es so weit, und du kannst den Raspberry Pi zum ersten Mal booten.

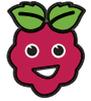
Stecke die Speicherkarte in den Steckplatz auf der Rückseite des Raspberry Pi und schließe Tastatur, Maus und Monitor an. Der USB-Stromanschluss kommt als Letztes. Damit wird der Raspberry Pi eingeschaltet. Einen extra Einschaltknopf gibt es nicht.

Der Raspberry Pi bootet und zeigt dabei auf einem schwarzen Bildschirm diverse Linux-Kommandos, die schnell durchrauschen und uns erst einmal nicht weiter interessieren müssen. Wundere dich nicht, wie



Die Benutzeroberfläche auf dem Raspberry Pi

# Raspberry Pi vorbereiten



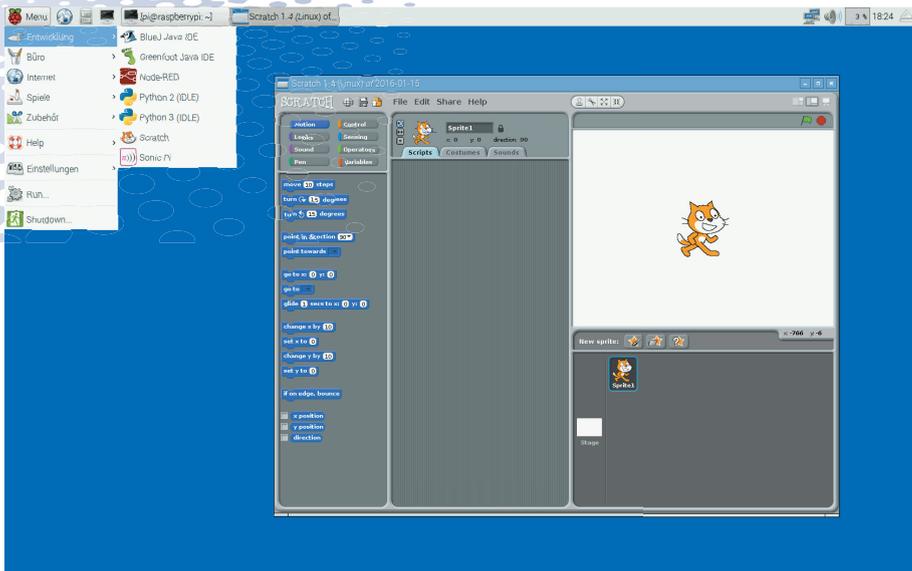
schnell das geht. Trotz der geringen Hardwareleistung bootet das komplette Linux-Betriebssystem deutlich schneller als ein Windows-PC. Am Ende erscheint ein Desktop, der Windows relativ ähnlich sieht. Die aus Windows bekannte Taskleiste befindet sich hier am oberen Bildschirmrand.

Das **Menü**-Symbol ganz links oben öffnet das Startmenü, die Symbole daneben den Webbrowser und den Dateimanager. Das Startmenü ist wie unter Windows mehrstufig aufge-

baut. Um die Experimente in diesem Kapitel nachzubauen und zu programmieren, brauchst du die Programme im Startmenü nicht alle. Wir verwenden nur Scratch.

Starte Scratch über das Desktop-Symbol oder aus dem Startmenü unter **Entwicklung**.

Wie unter Windows kannst du die Fenster über den Bildschirm schieben, indem du sie mit der Maus an der Titelleiste anfässt.





Mit dem zweiten Symbol rechts oben kannst du das Scratch-Fenster auf volle Bildschirmgröße vergrößern, um mehr Platz zum Programmieren zu haben.

### RASPBERRY PI AUSSCHALTEN

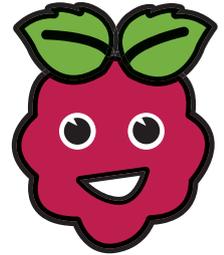
Theoretisch kann man bei dem Raspberry Pi einfach den Stecker ziehen, und er schaltet sich ab. Besser ist es jedoch, ihn wie einen PC sauber herunterzufahren. Wähle dazu im Startmenü *Shutdown* und klicke dann auf *Shutdown* und *OK*.



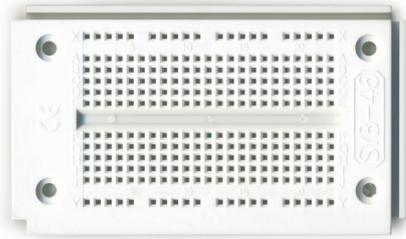
# Raspberry Pi vorbereiten



## DIE TEILE IM PAKET



1x Steckbrett



2x LEDs grün



2x LEDs rot



2x LED gelb



1x LED blau



1x RGB-LED



7x 220-Ohm-Widerstände



3x 20-MOhm-Widerstände





1x blanker Schaltdraht



8x Verbindungskabel



4x Krokodilklemmenkabel

2x Knete in verschiedenen  
Farben



In diesem Kapitel erhältst du das nötige Expertenwissen, um mit den Bauteilen in diesem Paket perfekt umgehen zu können.

# Raspberry Pi vorbereiten



## STECKBRETTER

Für den schnellen Aufbau elektronischer Schaltungen, ohne löten zu müssen, ist ein Steckbrett im Paket enthalten. Hier können elektronische Bauteile direkt in ein Lochraster gesteckt werden.

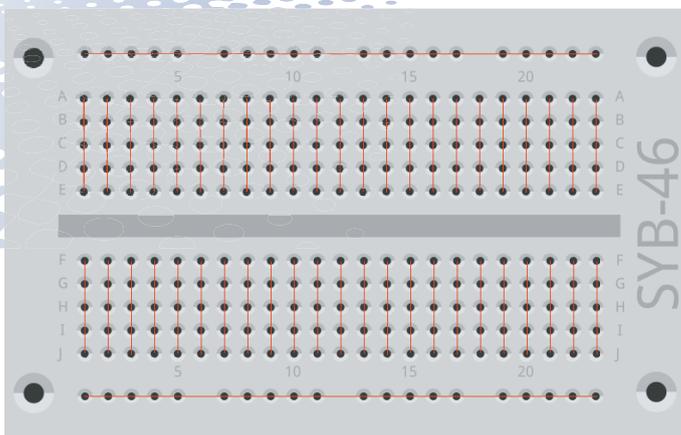
Bei diesen Steckbrettern sind die äußeren Längsreihen über Kontakte (X und Y) alle miteinander verbunden. Diese Kontaktreihen werden oft als Plus- und Minuspol zur Stromversorgung der Schaltungen genutzt. In den anderen Kontaktreihen sind jeweils fünf Kontakte (A bis E und F bis J) quer miteinander verbunden, wobei in der Mitte der Platine eine Lücke ist. So können in der Mitte größere Bauelemente gesteckt und nach außen hin verdrahtet werden.

## VERBINDUNGSKABEL

Die farbigen Verbindungskabel haben alle auf einer Seite einen dünnen Drahtstecker, mit dem sie sich auf das Steckbrett stecken lassen. Auf der anderen Seite ist eine Steckbuchse, die auf einen GPIO-Pin des Raspberry Pi passt.

## WIDERSTÄNDE

Ein Widerstand begrenzt den Strom, der durch eine Leitung fließt. Man kann ihn sich vorstellen wie ein dünnes Stück Gartenschlauch, das in ein Wasserrohr eingebaut wird. Durch die Rohrleitung fließt dann im Ganzen weniger Wasser, nämlich nur noch so viel, wie durch den Schlauch hindurchkommt.



Das Bild zeigt, welche Löcher auf dem Steckbrett miteinander verbunden sind.



Widerstände werden zur Strombegrenzung an empfindlichen elektronischen Bauteilen sowie als Vorwiderstände für LEDs verwendet. Die Maßeinheit für Widerstände ist Ohm. 1.000 Ohm entsprechen einem Kiloohm, abgekürzt kOhm. 1.000 kOhm entsprechen einem Megaohm, abgekürzt MOhm. Oft wird für die Einheit Ohm das Omega-Zeichen  $\Omega$  verwendet.

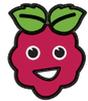
Die farbigen Ringe auf den Widerständen geben den Widerstandswert an. Mit etwas Übung sind sie deutlich leichter zu erkennen als die winzig

kleinen Zahlen, die man nur noch auf ganz alten Widerständen findet.

Die meisten Widerstände haben vier solcher Farbringe. Die ersten beiden Farbringe bezeichnen die Ziffern, der dritte einen Multiplikator und der vierte die Toleranz. Der Toleranzring ist meistens gold oder silbern, das heißt in Farben, die auf den ersten Ringen nicht vorkommen. Dadurch ist die Leserichtung immer eindeutig. Der Toleranzwert selbst spielt in der Digitalelektronik kaum eine Rolle. Die Tabelle zeigt die Bedeutung der farbigen Ringe auf Widerständen.

Farbe	Widerstandswert in Ohm			
	1. Ring (Zehner)	2. Ring (Einer)	3. Ring (Multiplikator)	4. Ring (Toleranz)
Silber			$10^{-2} = 0,01$	$\pm 10 \%$
Gold			$10^{-1} = 0,1$	$\pm 5 \%$
Schwarz		0	$10^0 = 1$	
Braun	1	1	$10^1 = 10$	$\pm 1 \%$
Rot	2	2	$10^2 = 100$	$\pm 2 \%$
Orange	3	3	$10^3 = 1.000$	
Gelb	4	4	$10^4 = 10.000$	
Grün	5	5	$10^5 = 100.000$	$\pm 0,5 \%$
Blau	6	6	$10^6 = 1.000.000$	$\pm 0,25 \%$
Violett	7	7	$10^7 = 10.000.000$	$\pm 0,1 \%$
Grau	8	8	$10^8 = 100.000.000$	$\pm 0,05 \%$
Weiß	9	9	$10^9 = 1.000.000.000$	

# Raspberry Pi vorbereiten



Im Paket sind Widerstände in nur zwei verschiedenen Werten enthalten:

Strom in einer Richtung durch sie hindurchfließt. In der anderen Rich-

Wert	1. Ring (Zehner)	2. Ring (Einer)	3. Ring (Multipl.)	4. Ring (Toleranz)	Verwendung
220 Ohm	Rot	Rot	Braun	Gold	Vorwiderstand für LEDs
20 MOhm	Rot	Schwarz	Blau	Gold	Widerstand für Knetkontakte

Achte darauf, die beiden Widerstandsarten nicht zu verwechseln. Die Schaltungen würden dann nicht funktionieren. Die Widerstände sehen zwar gleich groß aus, ein 20-MOhm-Widerstand ist aber etwa einhunderttausendmal stärker als ein 220-Ohm-Widerstand.

In welcher Richtung ein Widerstand eingebaut wird, ist egal. Bei LEDs dagegen spielt die Einbaurichtung eine wichtige Rolle.

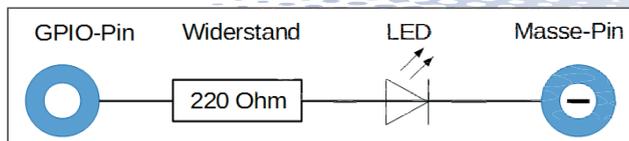
## LEDs

LED ist die Abkürzung für das englische Wort „light emitting diode“, was wörtlich übersetzt „Licht abstrahlende Diode“ oder einfach Leuchtdiode bedeutet. LEDs können schön bunt leuchten, wenn

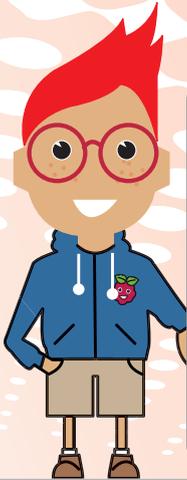
tung lassen sie nichts durch und leuchten auch nicht.

LEDs werden in Schaltungen mit einem pfeilförmigen Dreieckssymbol dargestellt, das die Flussrichtung vom Pluspol zum Minuspol oder zur Masseleitung angibt.

Eine LED lässt in der Durchflussrichtung nahezu beliebig viel Strom durch, sie hat nur einen sehr geringen Widerstand. Um den Durchflussstrom zu begrenzen und so ein Durchbrennen der LED zu verhindern, muss zwischen dem verwendeten GPIO-Pin und der Anode der LED ein 220-Ohm-Vorwiderstand (Rot-



Schaltplan einer LED mit Vorwiderstand



## LED – IN WELCHER RICHTUNG ANSCHLIESSEN?

Die beiden Anschlussdrähte einer LED sind unterschiedlich lang. Der längere ist die Anode (der Pluspol), der kürzere die Kathode (der Minuspol). Einfach zu merken: Das Pluszeichen hat einen Strich mehr als das Minuszeichen und macht quasi den Draht etwas länger. Außerdem sind die meisten LEDs auf der Minusseite abgeflacht, wie ein Minuszeichen. Leicht zu merken: Kathode = kurz = Kante.

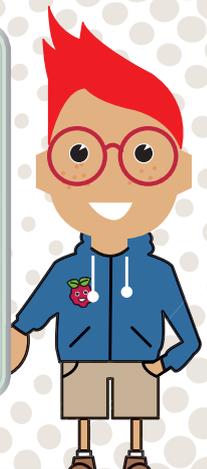


Rot-Braun) eingebaut werden. Dieser Vorwiderstand schützt auch den GPIO-Ausgang des Raspberry Pi vor zu hohen Stromstärken.

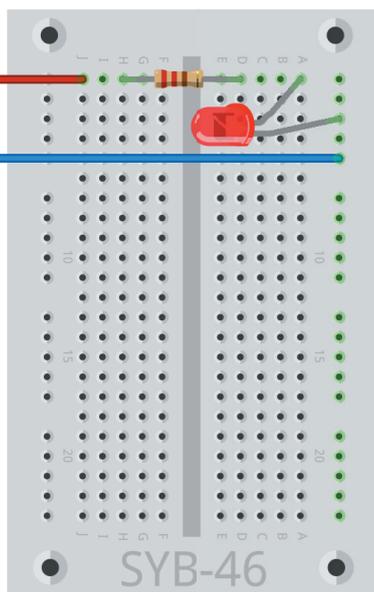
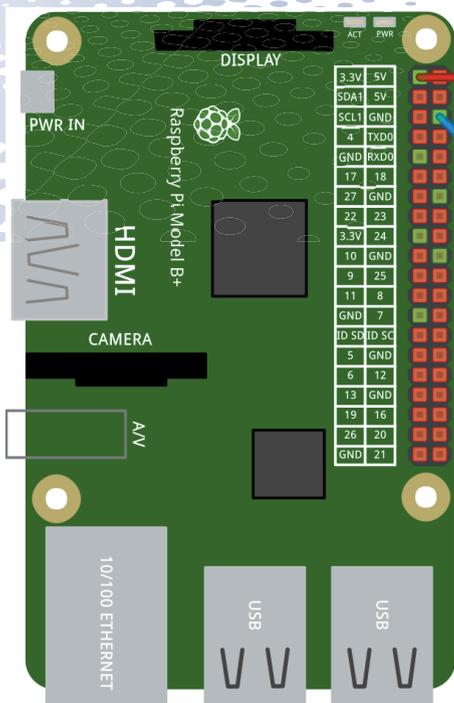
Probiere einfach einmal eine LED aus. Baue dazu die abgebildete Schaltung auf einem Steckbrett auf. In diesem ersten Experiment wird der Raspberry Pi nur als Stromversorgung für die LED genutzt. Die beiden verwendeten Pins am Raspberry Pi sind immer mit +3,3 V und Masse geschaltet. Die LED leuchtet immer, man braucht keinerlei Programm dazu.

## BENÖTIGTE BAUTEILE

- 1x Steckbrett
- 1x LED rot
- 1x 220-Ohm-Widerstand
- 2x Verbindungskabel



# Raspberry Pi vorbereiten



fritzing

Die erste LED am Raspberry Pi

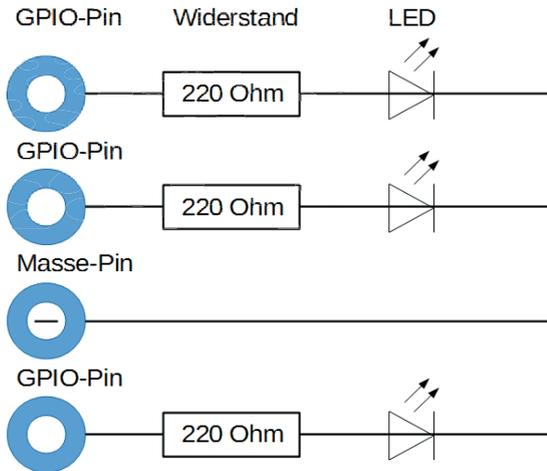
## RGB-LEDs

Eine normale LED leuchtet immer nur in einer Farbe. Bei den im Paket mitgelieferten LEDs kann man die Farbe auch dann erkennen, wenn die

LED aus ist. Es gibt aber auch LEDs, die transparent aussehen und ihre Farbe erst zeigen, wenn Strom fließt.



Anschlusspins einer RGB-LED



Schaltplan für eine RGB-LED mit drei Vorwiderständen

RGB-LEDs funktionieren wie drei einzelne LEDs und brauchen deshalb auch drei Vorwiderstände.

Mit der abgebildeten Schaltung kannst du die Funktionen einer RGB-LED ausprobieren. Wie beim letzten Experiment wird der Raspberry Pi nur als Stromversorgung für die LED genutzt.

RGB-LEDs können wahlweise in mehreren Farben leuchten. Bei ihnen sind im Prinzip drei LEDs mit verschiedenen Farben in ein transparentes Gehäuse eingebaut. Jede dieser drei LEDs hat eine eigene Anode, über die sie mit einem GPIO-Pin verbunden wird. Die Kathode, die mit der Masseleitung verbunden wird, ist nur einmal vorhanden. Deshalb hat eine RGB-LED vier Anschlussdrähte.

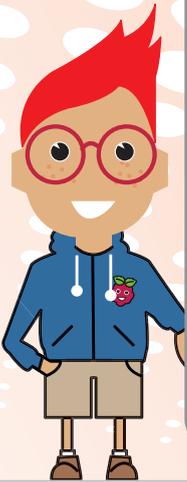
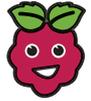
Die Anschlussdrähte der RGB-LEDs sind unterschiedlich lang, damit sie eindeutig zu erkennen sind. Im Gegensatz zu normalen LEDs ist die Kathode bei ihnen der längste Draht.

### BENÖTIGTE BAUTEILE

- 1x Steckbrett
- 1x RGB-LED
- 3x 220-Ohm-Widerstand
- 2x Verbindungskabel
- 4x Drahtbrücke
- 1x Krokodilklemmen-Kabel

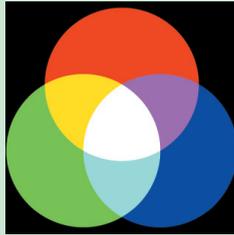


# Raspberry Pi vorbereiten



## ADDITIVE FARBMISSHUNG

RGB-LEDs verwenden die sogenannte additive Farbmischung. Dabei werden die drei Lichtfarben Rot, Grün und Blau addiert und ergeben reines Weiß. Im Gegensatz dazu verwendet ein Farbdrucker die subtraktive Farbmischung. Jede Farbe wirkt auf einem weißen Blatt wie ein Filter, der einen Teil des weiß reflektierenden Lichts wegnimmt (= subtrahiert). Drückt man alle drei Druckerfarben übereinander, erhält man Schwarz, das gar kein Licht mehr reflektiert.



Additive Farbmischung

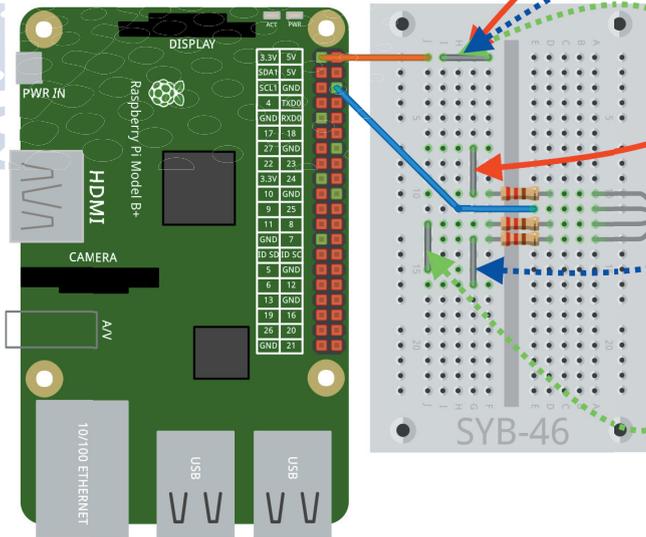
Die grauen Linien auf dem Steckbrett bezeichnen etwa 2 cm lange Drahtbrücken, die du aus dem mitgelieferten Draht zurechtbiegen kannst.

Beachte beim Einbau der RGB-LED, dass die flache Seite oben ist.

Nimm ein Krokodilklemmen-Kabel und klemme die eine Klemme an die Drahtbrücke oben am +3,3-V-Anschluss. Berühre mit der anderen Klemme nacheinander die anderen

drei Drahtbrücken. Die Pfeile zeigen die Farben, in denen die RGB-LED dann leuchten wird.





RGB-LED  
ausprobieren – dafür  
ist kein  
Programm  
nötig.

fritzing

### DRAHT SCHRÄG ABSCHNEIDEN

Der mitgelieferte Draht ist relativ dick, damit er sich nicht zu leicht verbiegt und sich mit den Krokodilklemmen gut greifen lässt. Es kann passieren, dass er sich besonders am Anfang, wenn die Steckbretter noch neu sind, schwer in die Kontaktlöcher stecken lässt. Schneide deshalb den Draht für die Anschlüsse am Steckbrett mit der Zange schräg ab. Dann entsteht eine leichte Spitze, die sich einfacher in das Steckbrett stecken lässt.



# Raspberry Pi vorbereiten



## KROKODILKLEMMME

Warum die Krokodilklemmen-Kabel so heißen, wie sie heißen, kann sich jeder leicht vorstellen. Wir haben uns diesen Namen übrigens nicht ausgedacht. Es ist die unter Elektrikern wirklich gebräuchliche Bezeichnung für diese Art von Kabel.



## KNETE, DRAHT UND KROKODILKLEMMEN-KABEL

Im Paket befinden sich Knete, Draht und Krokodilklemmen-Kabel. Damit bauen wir Kontaktschalter, um die Elektronikexperimente zu steuern. Die Knete kann Tasten oder bei einem Spiel auch ein ganzes Gamepad ersetzen.

## KNETEKONTAKTE BAUEN

- 1 Für einen Knetekontakt brauchst du als Erstes ein Stück Knete. Forme aus zwei Stücken Knete (rot und blau) zwei etwa 2 bis 3 cm große Kugeln oder eine andere Form, die gut in der Hand liegt.
- 2 Schneide mit einer Zange zwei etwa 4 bis 5 cm lange Stücke vom Draht ab, biege sie zu je einem





## WARUM EIGENTLICH KNETE?

Knete leitet Strom ähnlich gut wie deine Haut. Sie lässt sich leicht in jede beliebige Form bringen, und ein Knetekontakt fasst sich viel besser an als ein einfaches Stück Draht. Die Fläche, mit der deine Hand den Kontakt berührt, ist deutlich größer. So kommt es nicht so leicht zu einem „Wackelkontakt“. Natürlich kannst du auch andere leitfähige Dinge, wie zum Beispiel einen Löffel oder eine Münze, an ein Krokodilklemmen-Kabel klemmen und als Kontakt nutzen. Die neuartige leichte „Kugelnknete“ mit ihren eingebauten Styroporkügelchen eignet sich nicht. Sie leitet den Strom nicht gut genug.



„U“ und stecke die beiden Enden so weit in die Knetekugeln, dass der Draht nicht mehr von allein herausrutschen kann.

**3** Letztlich sollte er noch etwa  $\frac{1}{2}$  cm aus der Knete herausstehen. Klemme hier

eines der Krokodilklemmen-Kabel an.

**4** Das andere Ende des Krokodilklemmen-Kabels soll mit der Elektronik auf dem Steckbrett verbunden werden. Schneide dazu noch



# Raspberry Pi vorbereiten



## FÜR JEDEN KNETEKONTAKT BRAUCHST DU:

- 1 Stück Knete
- 1 U-förmig gebogenen Draht von etwa 4 bis 5 cm Länge für den Anschluss der Krokodilklemme an der Knete
- 1 U-förmig gebogenen Draht von etwa 2 bis 3 cm Länge für den Anschluss der Krokodilklemme am Steckbrett
- 1 Krokodilklemmen-Kabel



zwei kürzere, nur etwa 2 cm lange Stücke vom Draht ab und biege daraus auch je ein „U“. Stecke diese dann in das Steckbrett. In den Abbildungen sind diese Kabelanschlüsse grau dargestellt, um die blanken Drähte von den farbigen, isolierten Verbindungskabeln zu unterscheiden.

**5** Klemme zuletzt die Krokodilklemmen-Kabel der beiden Knetekontakte an die gerade gebauten Anschlüsse auf dem Steckbrett.

Da du jetzt weißt, was ein Knetekontakt ist und wie man ihn bastelt, werden wir diesen Schritt bei den folgenden Experimenten nicht mehr jedes Mal erklären.

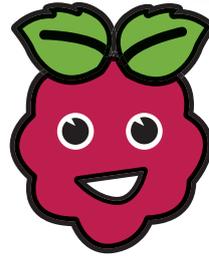
## WENN DIE KNETE HART WIRD

Die Knete ist so weich, weil sie ein bisschen Wasser enthält. Bei trockener Luft, wenn die Knete zum Beispiel in beheizten Räumen gelagert wird, kann sie austrocknen, nachdem sie ein paar Tage offen herumgelegen hat, und dann lässt sie sich nicht mehr so gut kneten. Außerdem kann die Leitfähigkeit nachlassen, und die Knetekontakte funktionieren dann nicht mehr so gut wie am ersten Tag.

Packe die Knete, wenn du sie ein paar Stunden nicht brauchst, deshalb immer in eine geschlossene Plastikdose. Die Dosen, die man zum



Lagern von Lebensmitteln im Kühlschrank verwendet, eignen sich gut. Du kannst auch einfach den Plastikbeutel verwenden, in dem die elektronischen Bauteile in diesem Paket verpackt sind.



Wenn die Knete doch einmal zu hart geworden ist, hilft dieser Trick:

### KNETE WIEDER WEICH MACHEN

Nimm ein Stofftuch, das nicht fusselt (sonst hast du die Fusseln später an der Knete kleben), und mache es mit Wasser nass. Drücke es dann so weit aus, dass es sich nur noch leicht feucht anfühlt. Wickle die Knete in das Tuch ein und lass sie über Nacht darin liegen. Am nächsten Morgen wird sich die Oberfläche leicht glitschig anfühlen. Knete die Masse gut durch, dann verteilt sich die Feuchtigkeit gleichmäßig, und die Knete kann wieder ganz normal benutzt werden. Damit die Feuchtigkeit besser in die Knete hineinkriechen kann, forme keine Kugeln, sondern drücke die Knete platt, bevor du sie in das Tuch packst. Die Kugel ist die Form, die im Verhältnis zum Volumen die geringste Oberfläche hat, es kann also wenig Feuchtigkeit eindringen. Umgekehrt kommt über die geringe Oberfläche auch kaum Feuchtigkeit heraus. Deshalb sind die meisten Früchte annähernd kugelförmig – damit sie im Sommer nicht so schnell austrocknen.



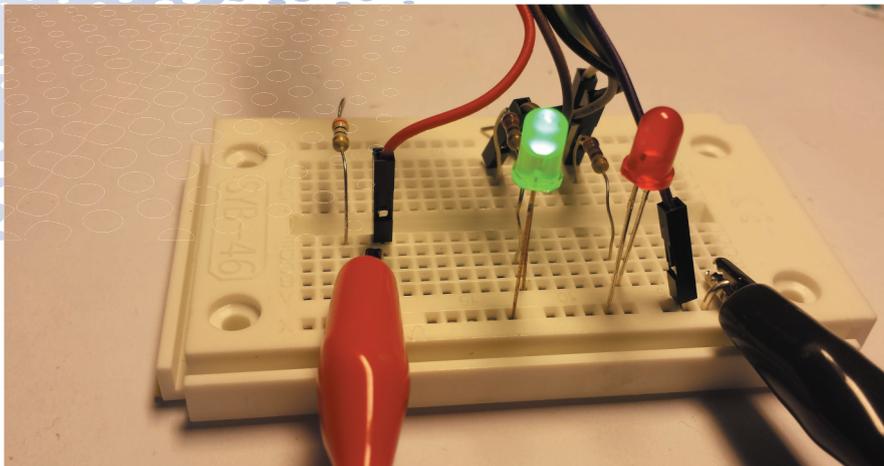
# 1 Elektronik steuern mit Knetekontakten



Jetzt geht es los! Nachdem der Raspberry Pi läuft, zeigt unser erstes Scratch-Programm, wie du den Raspberry Pi mit selbstgebaute Kontakten aus Knete steuern kannst. Probiere dieses Experiment in aller Ruhe aus, bevor du dich an die komplizierteren Versuche heranwagst. Denn hier erfährst du eine Menge Tricks, die du später noch brauchen wirst. Vielleicht klingt das Experiment erst mal nicht weiter aufregend, aber es liefert das gesamte wichtige Fachwissen, um später weitere Elektronik anzuschließen. Im Vergleich zu dem Versuch, mit einem

## BENÖTIGTE BAUTEILE

- 1x Steckbrett
- 1x LED rot
- 1x LED grün
- 2x 220-Ohm-Widerstand (rot-rot-braun)
- 1x 20-MOhm-Widerstand (rot-schwarz-blau)
- 5x Verbindungskabel
- 2x Drahtbrücke
- 2x Knetekontakt

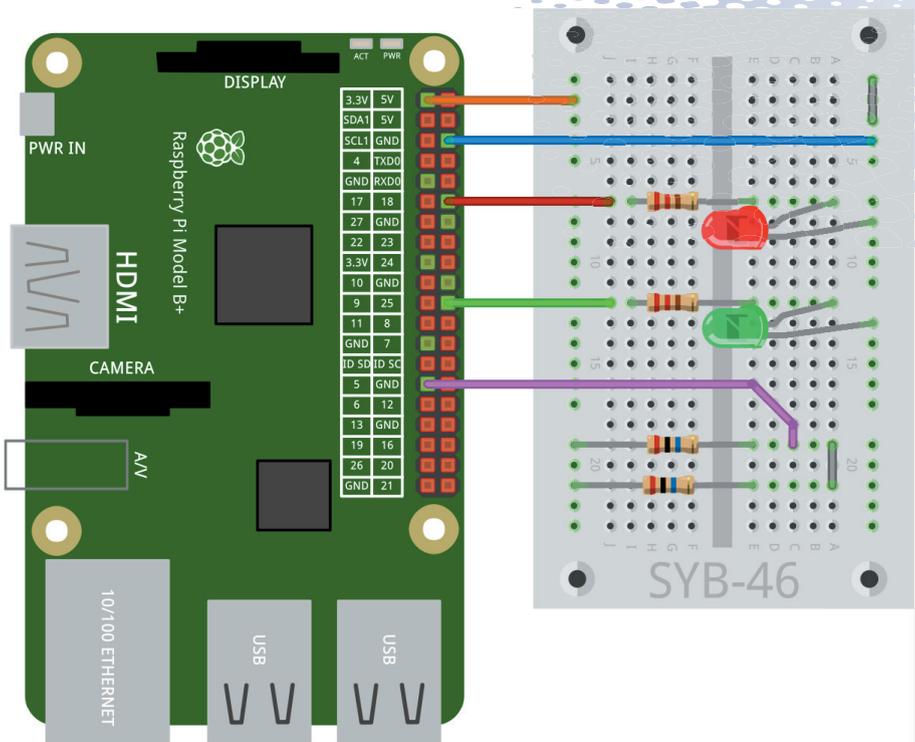




Windows PC eine extern angeschlossene LED zum Blinken zu bringen, geht es mit dem Raspberry Pi ganz einfach.

Baue zunächst die abgebildete Schaltung auf einem Steckbrett auf.

Die beiden LEDs werden mit 220-Ohm-Vorwiderständen an den GPIO-Pins 18 und 25 am Raspberry Pi angeschlossen. Das gleiche Prinzip werden wir beim Anschließen von LEDs immer anwenden.



fritzing

Die Schaltung auf dem Steckbrett

# 1 Elektronik steuern mit Knetkontakten



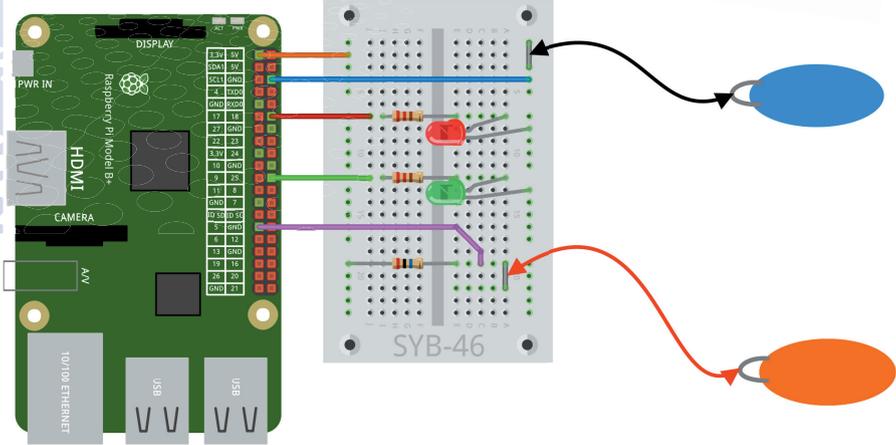
## WICHTIG



Eine LED bietet dem Strom fast keinen Widerstand. Es könnte also beliebig viel Strom fließen – gerade so, als ob man die beiden Pole einer Batterie miteinander verbindet. Die Batterie wäre nach Sekunden kaputt. Das Gleiche gilt für die GPIO-Ports des Raspberry Pi, die „durchbrennen“, wenn man sie mit einem Draht oder einer LED verbindet. Zwischen einem GPIO-Pin und der Anode der LED muss immer ein 220-Ohm-Vorwiderstand eingebaut sein. Die Kathode der LED verbindet man mit der Masseleitung auf Pin 6.

Eine der Kontaktleisten an den langen Seiten des Steckbretts ist mit der Masseleitung auf Pin 6 verbunden, die andere Seite mit der +3,3-V-Leitung auf Pin 1 des Raspberry Pi. Wir werden die gleiche Technik noch für andere Experimente verwenden, denn genau dafür – zur Stromversorgung – sind diese beiden langen Kontaktleisten auf den Steckbrettern schließlich gedacht.

Der 20-MOhm-Widerstand und die Leitung zum GPIO-Pin 5 sind für den Knetkontakt. Wir verwenden in diesem Experiment zwei Knetkontakte: Einer kommt an die Masseleitung, der andere an den GPIO-Pin 5.



fritzing

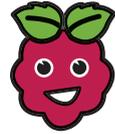
Die Schaltung mit angeschlossenen Knetekontakten

### KABELFARBEN

Üblicherweise werden in der Elektrotechnik rote Kabel für die Verbindung mit dem Pluspol der Schaltung und schwarze und blaue Kabel für die Verbindung mit der Masseleitung verwendet. Um leichter erklären zu können, welcher Knetekontakt gemeint ist, zeigen wir in der Abbildung rote und blaue Knete.



# 0 Elektronik steuern mit Knetkontakten



## HIGH ODER LOW?

Ein GPIO-Pin, der vom Programm ein- oder ausgeschaltet werden kann, wird wie im ersten Experiment als Ausgang bezeichnet, da der Raspberry Pi hier ein Signal ausgibt, nämlich eine 0 für ausgeschaltet oder eine 1 für eingeschaltet.

Umgekehrt kann ein GPIO-Pin auch als Eingang benutzt werden, der dem Programm eine 0 oder eine 1 schickt. Das hängt davon ab, welches Signal am Eingang anliegt. Man unterscheidet in der Digitalelektronik zwischen **Low**- (= niedrig) und **High**-Signalen (= hoch). Dabei gilt:

- **Low**: Der Eingang ist mit der Masse verbunden.
- **High**: Am Eingang liegt eine positive Spannung an.

### ACHTUNG!



Man darf auf keinen Fall einfach einen GPIO-Pin mit der +3,3-V-Leitung oder gar der +5-V-Leitung verbinden. Dies kann zu einem Kurzschluss führen und den Raspberry Pi beschädigen. Und das, obwohl es sich nur um eine geringe Spannung handelt, die man mit dem Finger nicht einmal bemerkt.

In elektronischen Geräten sind an solchen Stellen immer Schutzwiderstände eingebaut. Wir brauchen uns damit nicht herumzuzügeln, beim Schalten mit Knetkontakten spielst du selbst den Widerstand. Das funktioniert, weil der Mensch Strom leitet, zwar nicht so gut wie ein Draht, aber gut genug. Das kann bei den geringen Spannungen, die wir für Knetkontakte verwenden, sehr nützlich, bei hohen Spannungen aber auch sehr gefährlich sein.



## ACHTUNG!

Berühre nie eine Steckdose oder ein nicht isoliertes Kabel, das ans 230-V-Stromnetz angeschlossen ist! Auch diesen Strom würdest du, weil du über deine Füße immer mit der Masseleitung Erde verbunden bist, direkt dorthin leiten – und das ist lebensgefährlich! Das wäre in etwa so, als würde ein Blitz in deinen Körper einschlagen. Ein Gewitter ist nichts anderes als ein Kurzschluss zwischen positiv geladenen Wolken und der Erde, der durch die feuchte Luft herbeigeführt wird.

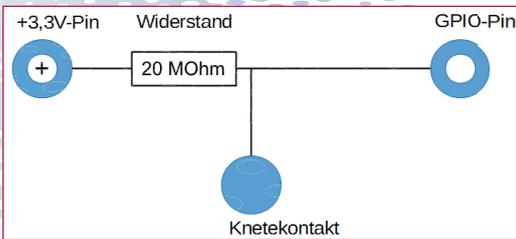


# Elektronik steuern mit Knetekontakten



## DER TRICK MIT DEN KNETEKONTAKTEN

Das Prinzip der Knetekontakte ist einfach: Der verwendete GPIO-Pin ist

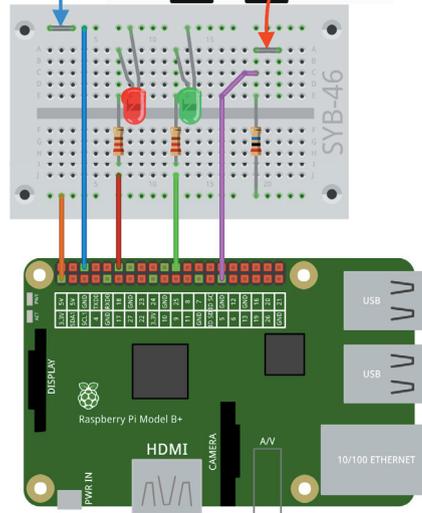


Schaltbild für einen Knetekontakt

über einen extrem hohen Widerstand (20 MOhm) mit +3,3 V verbunden, sodass ein schwaches, aber eindeutig als **High** definiertes Signal am GPIO-Pin anliegt.

Wenn du mit einer Hand den blauen Knetekontakt an der Masseleitung anfasst, ist dein Körper „geerdet“, also mit der Masse verbunden.

Wenn du gleichzeitig mit der anderen Hand den roten Knetekontakt am GPIO-Pin anfasst, wird das schwache **High**-Signal dort von dem deutlich stärkeren **Low**-Signal der Hand überlagert und zieht den entsprechenden GPIO-Pin auf **Low**.

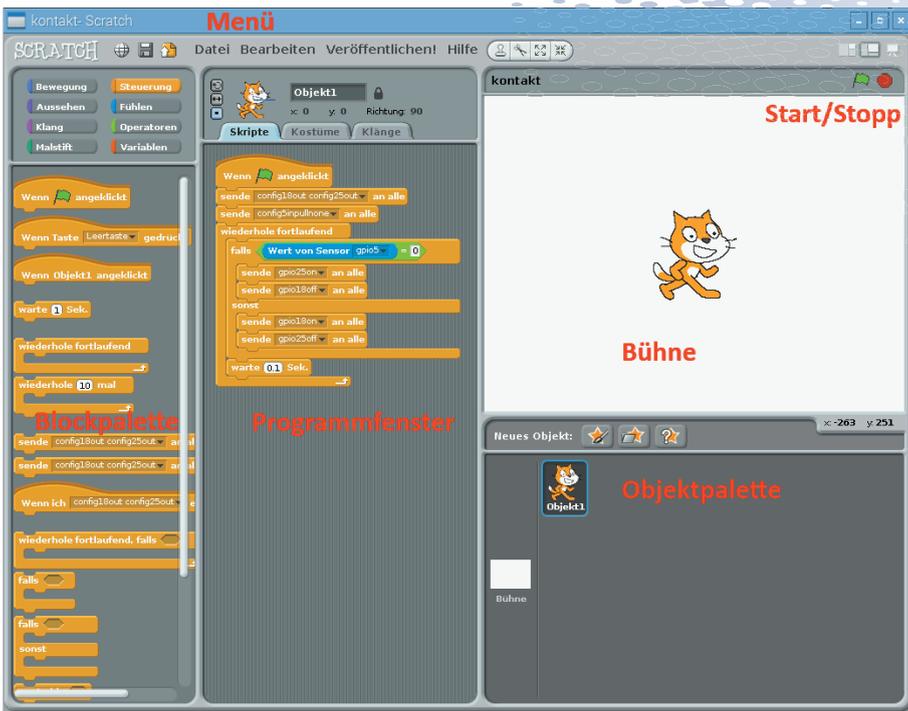




## ES GEHT LOS!

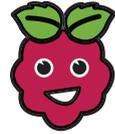


Klicke in Scratch oben links auf das gelbe Symbol **Steuerung**. Nun werden in der Blockpalette links die Blöcke zur Steuerung angezeigt. Ziehe die Blöcke, die du brauchst, einfach aus der Blockpalette links in das Skriptfenster in der Mitte von Scratch.



Dieses Scratch-Programm sorgt dafür, dass die LEDs bei Berührung des Knetkontakts die Farbe wechseln. Die Beschriftungen in Rot geben an, wie die verschiedenen Bereiche des Scratch-Fensters genannt werden.

# 1 Elektronik steuern mit Knetkontakten



## GPIO-FUNKTIONEN STARTEN

Um die GPIO-Unterstützung in Scratch nutzen zu können, musst du sie über den Menüpunkt *Bearbeiten/Start GPIO server* starten. Denke bei jedem neuen Scratch-Programm daran, diesen Schritt zu überprüfen. Wenn die GPIO-Funktionen bereits aktiv sind, ändert sich dieser Menüpunkt automatisch in *Stop GPIO server*.



Mit diesen Symbolen wird ein Programm gestartet und wieder angehalten.

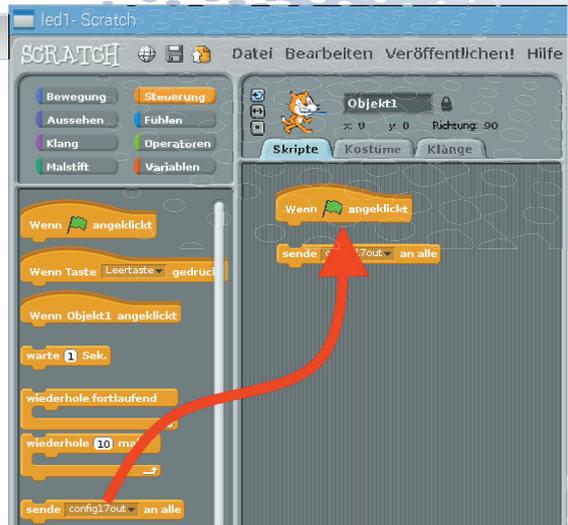
Mit einem Klick auf das rote Stopp-Symbol rechts oben kannst du das Programm wieder anhalten.



Nachdem du die Schaltung aufgebaut und die Knetkontakte

Einen nicht mehr benötigten Block kannst du jederzeit von den anderen Blöcken abtrennen und wieder zurück auf die Blockpalette ziehen.

Wenn du das Skript aus den Scratch-Blöcken fertig zusammengebaut hast und die Schaltung aufgebaut ist, klicke auf das grüne Fähnchen oben rechts in Scratch. Damit wird das Programm gestartet.



Wenn du einen neuen Block dicht unter einen anderen schiebst, rastet er ein und verbindet sich automatisch.



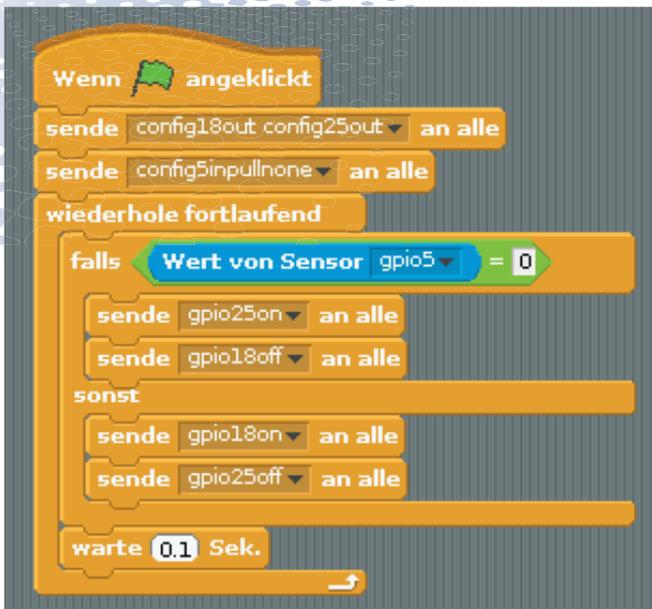
angeschlossen hast, starte das Programm und klicke auf das grüne Fähnchen. Du kannst das Programm auf dem Bildschirm selbst zusammenbauen oder das Programm *kontakt* von der DVD verwenden.

Die rote LED leuchtet. Nimm den blauen Knetekontakt an der Masseleitung in die eine Hand und gleichzeitig den roten Knetekontakt in die andere. Jetzt leuchtet die grüne LED. Wenn du loslässt, leuchtet wieder die rote LED.

● Grün = Kontakte verbunden, **Low** am GPIO-Pin

● Rot = Kontakte nicht verbunden, **High** am GPIO-Pin

Es mag vielleicht verwirrend klingen, dass ein **Low**-Signal, also eigentlich 0 V, am GPIO-Pin anliegt, wenn der Stromkreis geschlossen ist. Das kommt daher, dass du einen Pin, der im Ruhezustand einen schwachen **High**-Pegel hat, über die Knetkontakte und deine Hände mit der Masseleitung verbindest.

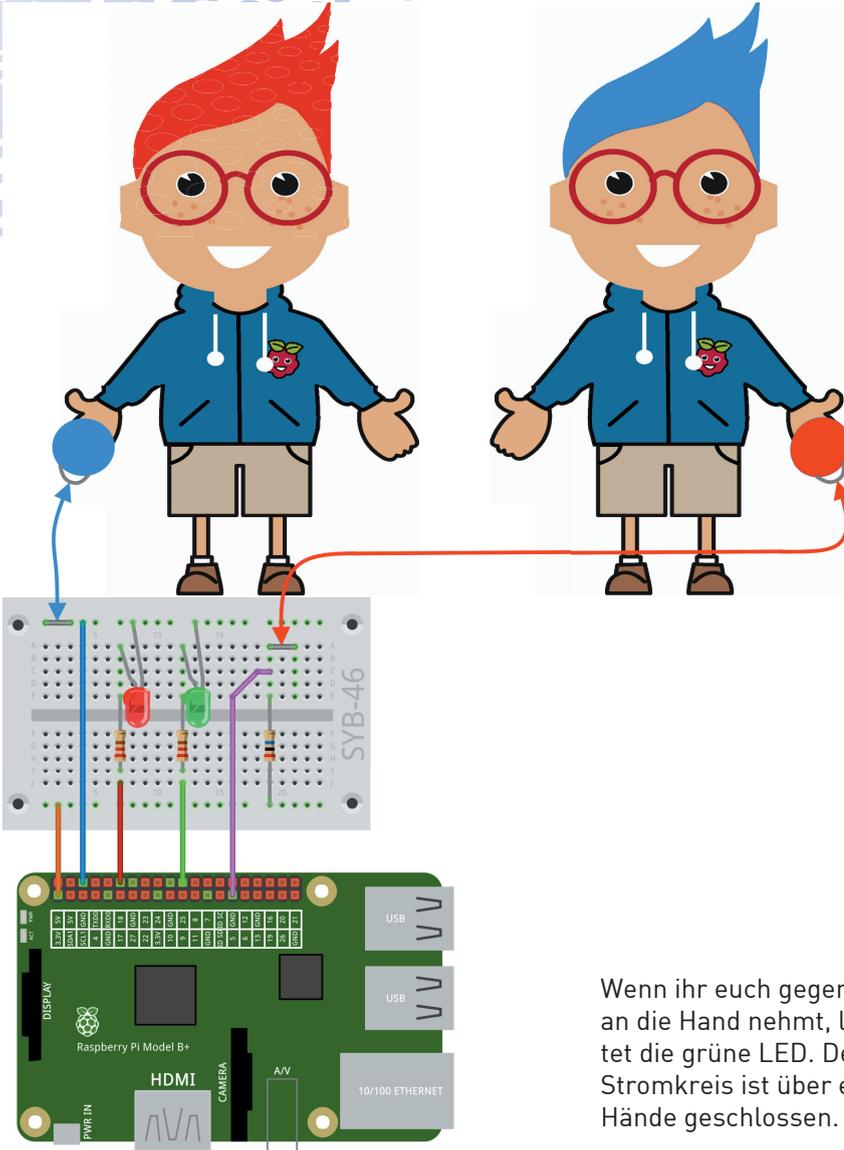
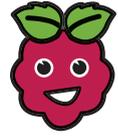


Das Programm *kontakt* zeigt, wie Knetkontakte funktionieren.

## ZWEI PERSONEN

Du kannst den Stromkreis auch über mehrere Personen schließen. Nimm den blauen Knetekontakt in eine Hand und bitte eine andere Person, den roten Knetekontakt in die Hand zu nehmen. Die rote LED leuchtet.

# 1 Elektronik steuern mit Knetkontakten



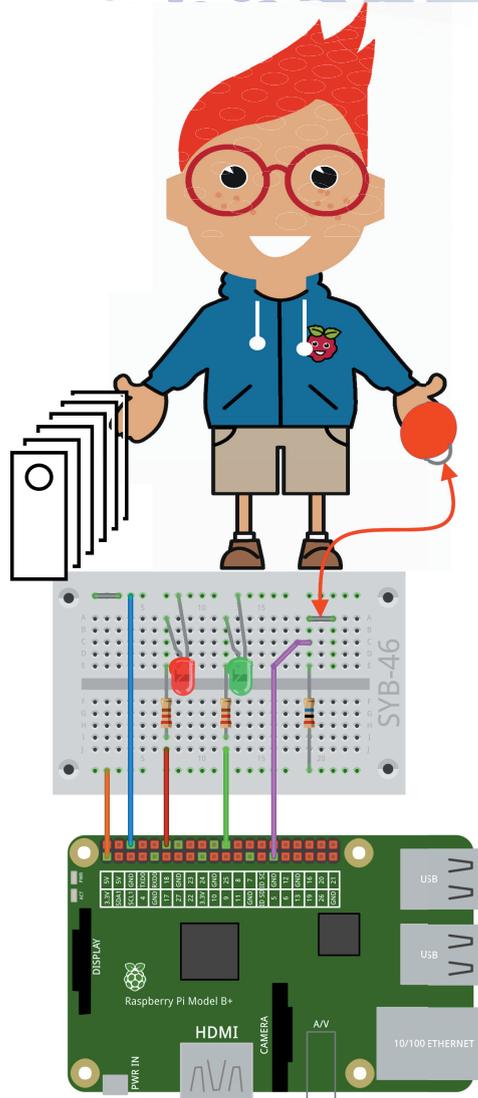
Wenn ihr euch gegenseitig an die Hand nimmt, leuchtet die grüne LED. Der Stromkreis ist über eure Hände geschlossen.



## ES GEHT AUCH OHNE MASSEKONTAKT

Nimm jetzt nur den roten Knetkontakt in die Hand, ohne über den blauen Knetkontakt mit der Masseleitung verbunden zu sein. Bewege dich etwas, stehe auf, setz dich hin oder berühre mit der anderen Hand ein Möbelstück. Die grüne LED wird immer wieder mal mehr, mal weniger lang leuchten.

Das bedeutet, dass der GPIO-Pin mit Masse verbunden ist, aber wie? Ein Mensch ist über seine Füße fast immer mit der Masse der Erde verbunden. Wie hoch allerdings der Widerstand zwischen deiner Hand und der Erde ist, hängt von vielen Dingen ab, vor allem davon, was für Schuhe du anhast und auf welchem Fußboden du stehst. Barfuß im nassen Gras ist die Verbindung zur Masse der Erde am besten, aber auch auf Steinfußboden funktioniert es meistens gut. Du kannst auch mit der freien Hand ein geerdetes Metallteil wie zum Beispiel einen Heizkörper oder einen Wasserhahn berühren, um guten Kontakt zur Erde zu bekommen und die grüne LED zum Leuchten zu bringen.



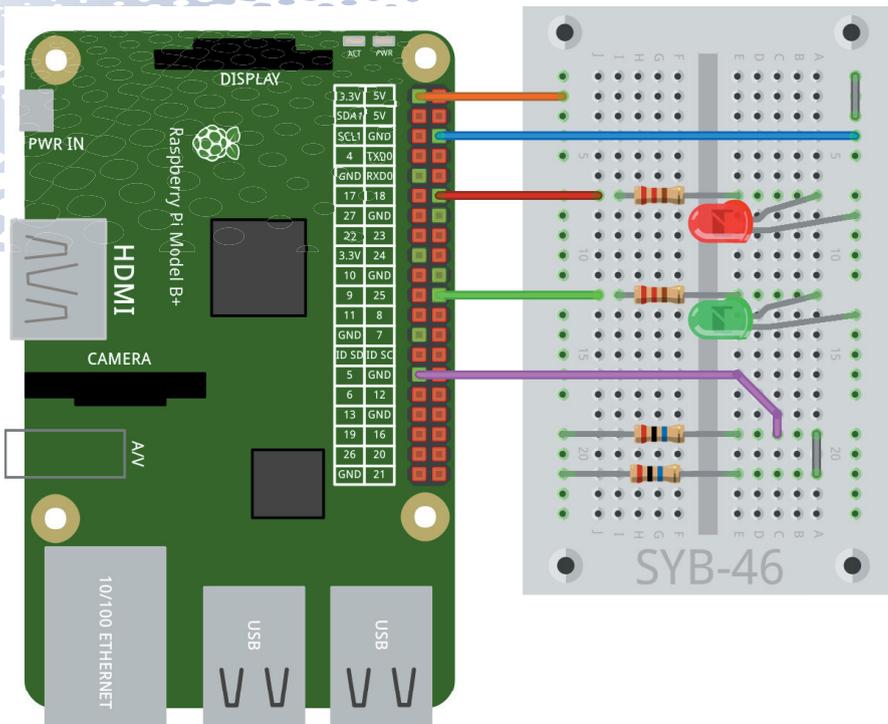
# 0 Elektronik steuern mit Knetkontakten



## WENN DIE LEDS BLINKEN

In manchen Fällen kann es passieren, dass die LEDs abwechselnd blinken, auch wenn du gar keinen der Knetkontakte berührst. Dies kann verschiedene Ursachen haben, zum Beispiel, dass die Tischplatte

oder eine sonstige Unterlage, auf der die Knetkontakte liegen, leitfähig ist und so den Stromkreis schließt. Wenn du den roten Knetkontakt am isolierten Kabel hochhebst, ohne die Knete oder eine blanke Drahtstelle zu berühren, sollte die rote LED wieder durchgehend leuchten.



fritzing

Die beiden 20-MOhm-Widerstände sind an beiden Enden miteinander verbunden: links über die Kontaktleiste im Steckbrett, rechts über die Drahtbrücke, an die die Krokodilklammer angeschlossen wird.



Selbst geringe statische Elektrizitätsfelder, wie sie zum Beispiel durch kunststoffbeschichtete Tischplatten oder auch Teppichböden entstehen, können schon zum Blinken führen. Die Verbindung des GPIO-Pins über den 20-MOhm-Widerstand mit dem Pluspol der Schaltung ist zu schwach, um diese Störfelder auszugleichen.

Hier hilft ein Trick: Baue einfach wie in der Abbildung einen zweiten 20-MOhm-Widerstand parallel zum vorhandenen Widerstand ein. Das löst solche Probleme meistens.

Wichtig zu wissen:

- Widerstände hintereinander erhöhen den Gesamtwiderstand.
- Widerstände parallel nebeneinander verringern den Gesamtwiderstand.

Diese Schaltungen kann man, etwas vereinfacht, mit Gartenschläuchen vergleichen. Wenn du einen Becher Wasser an einem Ende in einen Gartenschlauch schüttest, kommt es am anderen Ende nur noch langsam an. Steckst du zwei Gartenschläuche hintereinander, wird am Ende nur noch wenig Wasser sprudeln. Hältst

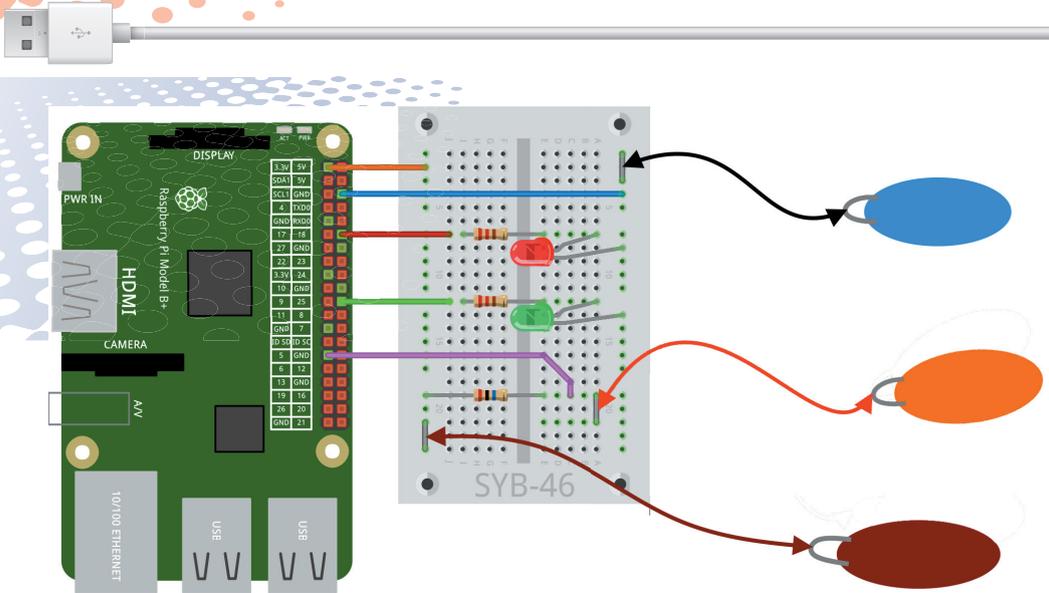
du hingegen zwei Gartenschläuche nebeneinander in der Hand und kippst das Wasser gleichzeitig in beide Öffnungen, kommt an den beiden Enden insgesamt mehr heraus.

Den Trick mit dem doppelten Widerstand kannst du in allen Experimenten mit Knetekontakten anwenden.

## WENN ES GAR NICHT FUNKTIONIERT

In seltenen Fällen kann es vorkommen, dass die grüne LED ab und zu aufblinkt, selbst wenn du den roten Knetekontakt frei in die Luft hältst oder ihn nicht einmal angeschlossen hast. In solchen Fällen hat der GPIO-Pin keinen eindeutig definierten Zustand, er ist also weder **High** noch **Low**. So etwas darf in der Digitalelektronik eigentlich nicht vorkommen, kann aber durch Störfelder – oft auch als Elektrosmog bezeichnet – passieren. Solche elektrischen Felder entstehen durch Handys, Computer, Fernseher, Mikrowellen und zahlreiche andere elektrische Geräte in der unmittelbaren Umgebung, können aber bei bestimmten Wetterlagen auch natürlich entstehen. Sogar besondere Gesteinsformationen im Erdboden unter dir können

# Elektronik steuern mit Knetkontakten



fritzing

Der dritte Knetkontakt am Pluspol

elektromagnetische Unregelmäßigkeiten verursachen, die sich bei den geringen Strömen, mit denen unsere Knetkontakte arbeiten, schon bemerkbar machen können.

Aber auch in diesen Fällen kannst du mit einem Trick nachhelfen. Baue dazu einen dritten Knetkontakt aus Knete und Draht und schließe ihn, wie in der Abbildung gezeigt, an eine neue Drahtbrücke an der +3,3-V-Kontaktleiste an.

Wenn du jetzt gleichzeitig den (dunkelrot dargestellten) Knetkontakt am Pluspol und den (orange dargestellten) Knetkontakt am GPIO-Pin berührst, setzt du den GPIO-Pin auf ein deutliches **High**-Signal. Berührst du nun den blauen Knetkontakt und lässt dafür den dunkelroten los, setzt du den GPIO-Pin wieder auf **Low**.

Auch dieser Trick funktioniert natürlich bei allen Experimenten mit den Knetkontakten, sollte aber nur in Ausnahmefällen nötig sein.



## SO FUNKTIONIERT DAS PROGRAMM

Das Programm startet wie die meisten Scratch-Programme mit dem Block **Wenn grünes Fähnchen angeklickt**, der in Scratch auf der Blockpalette **Steuerung** zu finden ist. Der Block ist oben rund, passt also unter keinen anderen Block. Er muss immer als Erstes gesetzt werden.



Scratch verwendet für die GPIO-Funktionen den Steuerungsblock **sende... an alle**.



Dieser Block enthält ein Feld, in dem freier Text eingegeben werden kann. Klickst du darauf, erscheint eine Liste der zuletzt verwendeten Eingabe-

ben. Ein Klick auf **Neu/edit...** öffnet ein Eingabefeld für neuen Text.



Schreibe in dieses Feld **config18out config25out** und bestätige die Eingabe mit **OK**.

Am Anfang werden die beiden GPIO-Pins 18 und 25 als Ausgänge (**out**) für die LEDs initialisiert. Mehrere Pins können in einem **sende...an alle**-Block initialisiert werden. Zwischen den beiden Wörtern **config18out** und **config25out** muss ein Leerzeichen stehen. Jeder GPIO-Pin muss, bevor er verwendet werden kann, als Ausgang (z. B. für LEDs) oder als Eingang (z. B. für Taster oder Knete-Kontakte) definiert werden.

# 0 Elektronik steuern mit Knetekontakten



Anschließend wird der GPIO-Pin 5 als Eingang initialisiert. Damit die Sensorkontakte funktionieren, müssen zuerst die internen Pull-up-Widerstände an den GPIO-Pins ausgeschaltet werden, die Scratch auf dem Raspberry Pi standardmäßig immer einschaltet. Das erledigt der Zusatz **pullnone** am GPIO-Befehl **config5in**.

```
Wenn [Knetekontakt] angeklickt
  sende config18out config25out an alle
  sende config5inpullnone an alle
```

zwei verschiedene Dinge tun, nämlich die grüne oder die rote LED leuchten lassen – je nachdem, ob der Knetekontakt berührt wird oder

Danach startet eine Endlosschleife, die sich automatisch so lange wiederholt, bis du auf das rote Stopp-Symbol klickst.

Innerhalb der Schleife muss das Programm

```
Wenn [Knetekontakt] angeklickt
  sende config18out config25out an alle
  sende config5inpullnone an alle
  wiederhole fortlaufend
    falls [Wert von Sensor gpio5] = 0
    sonst
```

```
Wenn [Knetekontakt] angeklickt
  sende config18out config25out an alle
  sende config5inpullnone an alle
  wiederhole fortlaufend
```

nicht. Dazu verwenden wir den Steuerungsblock **falls...sonst**.

Innerhalb der beiden Klammern des **falls...sonst**-Blocks werden die Blöcke eingefügt, die ausgeführt werden sollen, wenn die Abfrage wahr oder falsch ergibt.



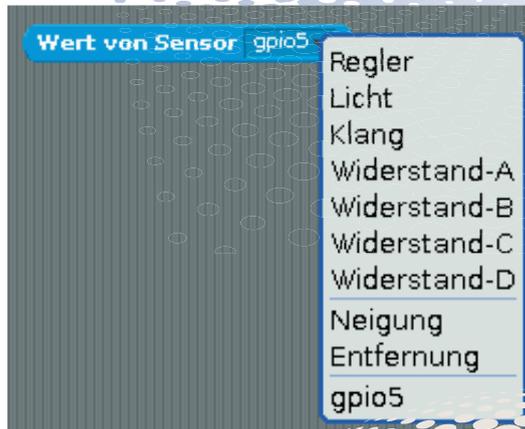
Für die Abfrage selbst ist im **falls**-Block ein längliches Feld mit spitzen Enden vorgesehen. Hier muss ein Block aus der grünen Blockpalette **Operatoren** eingefügt werden. Ziehe den Block mit dem Gleichheitszeichen auf das Platzhalterfeld im Block **falls**.



Sensoren werden alle GPIO-Pins zur Auswahl angeboten, die als Eingang definiert sind. Deshalb musste das Programm einmal kurz gestartet werden.

Dieser Operator ist immer dann wahr, wenn die beiden Werte links und rechts des Gleichheitszeichens identisch sind.

In unserem Fall soll der Wert des GPIO-Pins 5 dem Wert **0** entsprechen. Die Ziffer **0** steht für **Low**. Schreibe also eine **0** in das rechte der beiden Textfelder im grünen Block.



Klicke einmal auf das grüne Fähnchen oben rechts, um das noch unfertige Programm zu starten. Damit werden die GPIO-Pins definiert. Klicke dann wieder auf das rote Stopp-Schild.

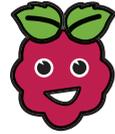
Zur Abfrage von GPIO-Eingängen wird der Block **Wert von Sensor...** aus der blauen Blockpalette **Fühlen** verwendet. Wähle im Listenfeld des blauen Blocks den Sensor **gpio5** aus. Neben einigen vordefinierten

### GPIO-EINGANG ERSCHEINT NICHT

Sollte der GPIO-Eingang auch nach einigen Sekunden nicht von allein in der Auswahlliste auftauchen, überprüfe im Menü **Bearbeiten**, ob der GPIO-Server gestartet worden ist.

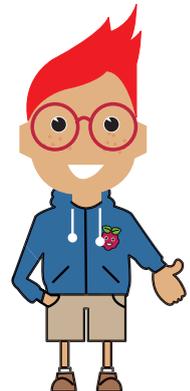


# 0 Elektronik steuern mit Knetekontakten



Ziehe den blauen Block **Wert von Sensor...** in das linke Feld des grünen Blocks und diesen in das Feld im Titel des **falls...sonst**-Blocks.

Wenn diese Abfrage wahr ist, am GPIO-Pin 5 also ein **Low**-Signal (0) anliegt, berührt jemand den roten Knetekontakt. Jetzt soll die grüne LED am GPIO-Pin 25 leuchten und die rote LED am GPIO-Pin 18 ausgehen.





```

Wenn [flagge] angeklickt
  sende config18out config25out an alle
  sende config5inpullnone an alle
  wiederhole fortlaufend
    falls Wert von Sensor gpio5 = 0
      sende gpio25on an alle
      sende gpio18off an alle
    sonst
  ende wiederhole

```

**gpio18on** und **gpio25off**, denn in diesem zweiten Fall (**sonst**) soll die rote LED am GPIO-Pin 18 leuchten und die grüne LED am GPIO-Pin 25 ausgehen. Dieser Fall tritt immer dann ein, wenn die Abfrage am Anfang des Blocks das Ergebnis **falsch** liefert, der Wert von Sensor GPIO 5 also

Du kannst diese beiden Zeilen einfach duplizieren und in den unteren Teil der Abfrage unter **sonst** einfügen. Klicke zum Duplizieren mit der rechten Maustaste auf den ersten Block, der dupliziert werden soll. Wähle dann im Menü **Duplizieren**. Der darunter hängende **sende...an alle**-Block wird automatisch mit dupliziert.

nicht **0**, sondern **1** ist. In diesem Fall besteht keine Verbindung zur Mas-

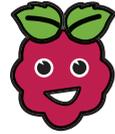
Ändere im kopierten Block die beiden Textfelder auf

```

Wenn [flagge] angeklickt
  sende config18out config25out an alle
  sende config5inpullnone an alle
  wiederhole fortlaufend
    falls Wert von Sensor gpio5 = 0
      sende gpio25on an alle
      sende gpio18off an alle
    sonst
      sende gpio18on an alle
      sende gpio25off an alle
  ende wiederhole

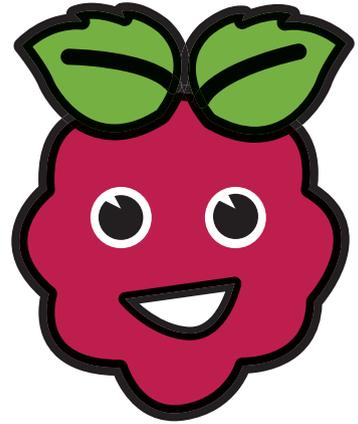
```

# 0 Elektronik steuern mit Knetekontakten



selektion, der Knetekontakt wird also nicht berührt.

Nach der Abfrage wartet das Programm eine Zehntelsekunde lang. Solche sogenannten „Timeouts“ oder auf Deutsch einfach „Auszeiten“ baut man immer dann ein, wenn Programme direkt mit Hardware kommunizieren. Vereinfacht ausgedrückt, verhindern sie, dass sich ein Programm „überschlägt“ und irgendein Hardwareereignis nicht mehr mitbekommt.



```
Wenn  angeklickt
  sende config18out config25out an alle
  sende config5inpullnone an alle
  wiederhole fortlaufend
    falls Wert von Sensor gpio5 = 0
      sende gpio25on an alle
      sende gpio18off an alle
    sonst
      sende gpio18on an alle
      sende gpio25off an alle
  warte 0.1 Sek.
```

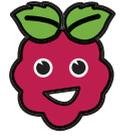


## PROGRAMM SPEICHERN

Vergiss nicht, das fertige Programm mithilfe des Menüpunkts *Datei/Speichern als* zu speichern, um es später wiederverwenden zu können.



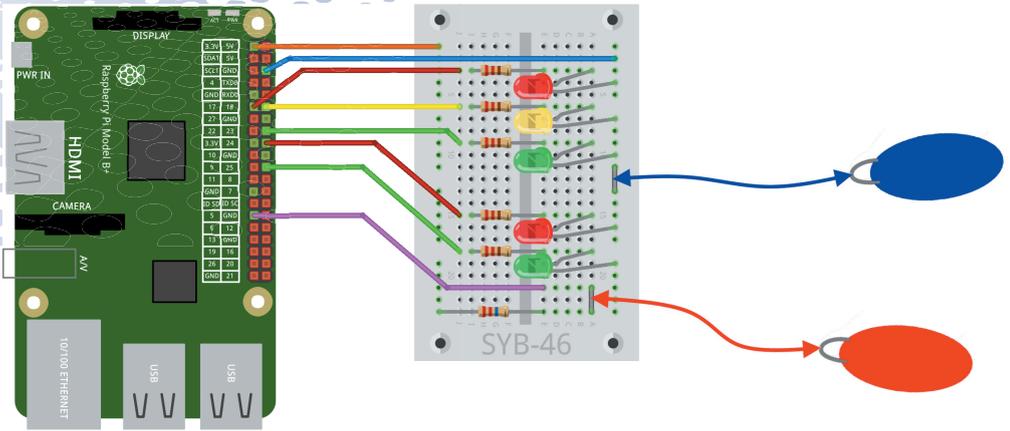
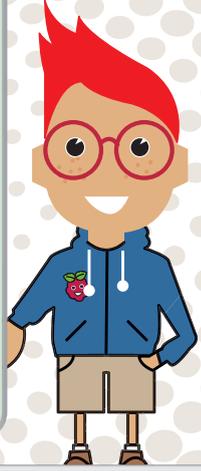
# 2 Fußgängerampel



Eine einzelne LED ein- und wieder auszuschalten mag im ersten Moment ganz spannend sein, aber dafür braucht man eigentlich keinen Computer. Eine Ampel mit ihrem typischen Lichtwechsel von Grün über Gelb nach Rot und dann über eine Lichtkombination Rot-Gelb wieder zu Grün ist schon viel spannender.

Das nächste Experiment stellt eine einfache Ampelschaltung dar, mit einer Ampel für Autos und einer für Fußgänger: Während der Rotphase der Verkehrsampel soll die Fußgängerampel Grün anzeigen. Umgesetzt wird das auf

- BENÖTIGTE BAUTEILE**
- 1x Steckbrett
  - 2x LED rot
  - 2x LED grün
  - 1x LED gelb
  - 5x 220-Ohm-Widerstand (rot-rot-braun)
  - 1x 20-MOhm-Widerstand (rot-schwarz-blau)
  - 8x Verbindungskabel
  - 2x Drahtbrücke
  - 2x Knetkontakt



Verkehrsampel und Fußgängerampel



einem Steckbrett mit fünf LEDs, einem Knetekontakt für den Drucktaster sowie bei Bedarf noch einem Massekontakt.

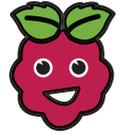
Baue die abgebildete Schaltung auf einem Steckbrett auf und schließe die Knetekontakte an.

Nachdem du die Schaltung aufgebaut und die Knetekontakte angeschlossen hast, starte das Programm **ampel**.

Um mehr Platz für das Programm zu haben, kannst du mit den Symbolen oben rechts im Scratch-Fenster

```
Wenn angeklickt
  sende config17out config18out config23out config24out config25out an alle
  sende config5inpullnone an alle
  sende gpio17off gpio18off gpio23on gpio24on gpio25off an alle
  wiederhole fortlaufend
    falls Wert von Sensor gpio5 = 0
      sende gpio18on an alle
      sende gpio23off an alle
      warte 0.6 Sek.
      sende gpio17on an alle
      sende gpio18off an alle
      warte 0.6 Sek.
      sende gpio25on an alle
      sende gpio24off an alle
      warte 2 Sek.
      sende gpio24on an alle
      sende gpio25off an alle
      warte 0.6 Sek.
      sende gpio18on an alle
      warte 0.6 Sek.
      sende gpio23on an alle
      sende gpio17off an alle
      sende gpio18off an alle
      warte 2 Sek.
```

# 2 Fußgängerampel



den Skriptbereich verbreitern und die Bühne mit der Katze, die wir im Moment nicht brauchen, verkleinern.

Mit dem letzten Lichtmuster – Fußgängerampel rot, Verkehrsampel grün – erreicht die Ampel wieder den

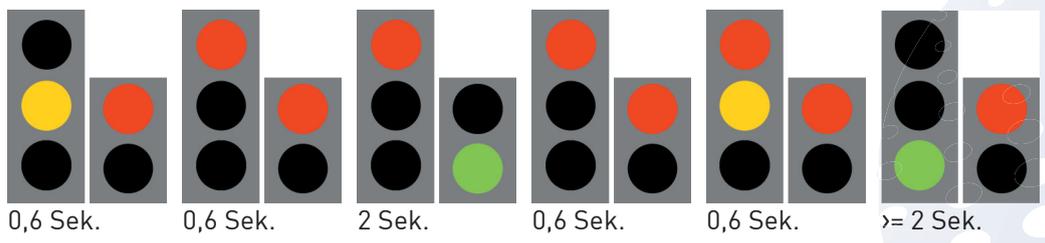


Klicke auf das grüne Fähnchen, um das Programm zu starten. Die Verkehrsampel leuchtet grün, die Fußgängerampel rot, genauso, wie es echte Ampeln stundenlang tun, solange kein Fußgänger kommt und auf den Knopf drückt.



Standardzustand. Das Programm muss allerdings dafür sorgen, dass auch dieser immer eine Mindestzeit lang eingehalten wird. Selbst wenn ständig Fußgänger auf den Knopf drücken, müssen die Autos auch mal fahren dürfen. In unserer Modellampel sind das 2 Sekunden, bei einer wirklichen Ampel dauert diese Phase natürlich deutlich länger.

Berühre den roten Knetkontakt oder – je nachdem, wie gut deine Erdung ist – gleichzeitig den blauen Knetkontakt an der Masseleitung. Jetzt startet der Ampelzyklus, der in unserem Programm wie auch bei einer echten Ampel aus sechs unterschiedlichen Lichtmustern besteht, die unterschiedlich lange leuchten.



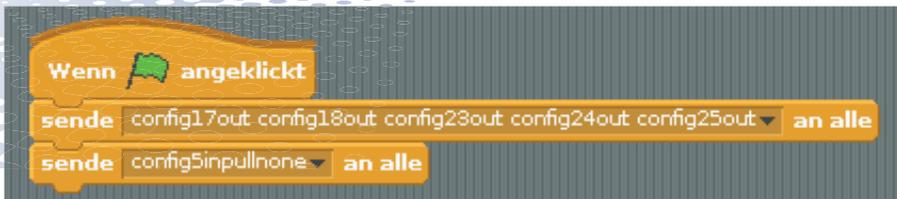


## SO FUNKTIONIERT DAS PROGRAMM

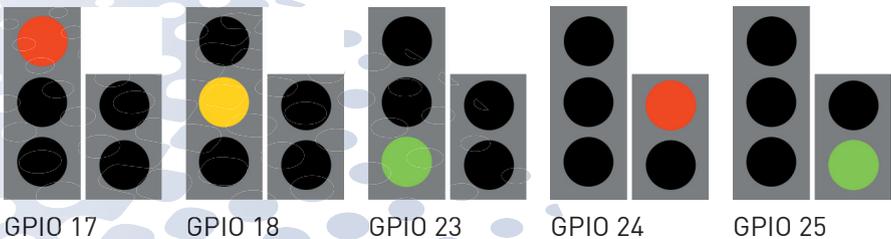
Wenn du auf das grüne Fähnchen klickst, werden als Erstes die fünf als Ausgänge verwendeten GPIO-Pins initialisiert. Hier verwenden wir wieder einen einzigen **sende ... an alle**-Block für alle GPIO-Pins. Danach wird noch der GPIO-Pin 5 als Eingang definiert.



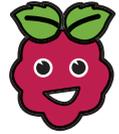
Nach einer kurzen Wartezeit wird die Grundstellung der Ampel eingeschaltet, d. h. Grün für die Autos und Rot für die Fußgänger. Die anderen drei LEDs werden ausgeschaltet. Das Ausschalten ist hier am Anfang eigentlich nicht nötig. Es dient nur dazu, das Programm auf jeden Fall mit einem klar definierten Zustand zu starten.



Die Tabelle zeigt, welcher GPIO-Port welche LED steuert.

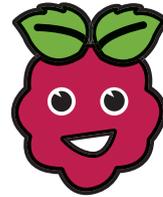


## 2 Fußgängerampel



```
Wenn  angeklickt
  sende config17out config18out config23out config24out config25out ▼ an alle
  sende config5inputnone ▼ an alle
  sende gpio17off gpio18off gpio23on gpio24on gpio25off ▼ an alle
```

Jetzt beginnt wie im letzten Programm eine **wiederhole fortlaufend**-Schleife, die den Knetekontakt am GPIO-Pin 5 abfragt. Wenn dieser den Wert **Low** liefert, werden nacheinander die unterschiedlichen Lichtmuster der Ampel geschaltet. Danach wartet die Ampel 0,6 Sekunden bis zum nächsten Lichtmuster. Während die Fußgängerampel grün leuchtet, wird 2 Sekunden gewartet.





Wenn  angeklickt

sende config17out config18out config23out config24out config25out ▼ an alle

sende config5inpullnone ▼ an alle

sende gpio17off gpio18off gpio23on gpio24on gpio25off ▼ an alle

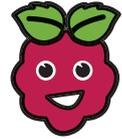
wiederhole fortlaufend

falls Wert von Sensor gpio5 ▼ = 0

- sende gpio18on ▼ an alle
- sende gpio23off ▼ an alle
- warte 0.6 Sek.
- sende gpio17on ▼ an alle
- sende gpio18off ▼ an alle
- warte 0.6 Sek.
- sende gpio25on ▼ an alle
- sende gpio24off ▼ an alle
- warte 2 Sek.
- sende gpio24on ▼ an alle
- sende gpio25off ▼ an alle
- warte 0.6 Sek.
- sende gpio18on ▼ an alle
- warte 0.6 Sek.
- sende gpio23on ▼ an alle
- sende gpio17off ▼ an alle
- sende gpio18off ▼ an alle
- warte 2 Sek.

↪

# 3 RGB-LED mit Knetkontakten steuern

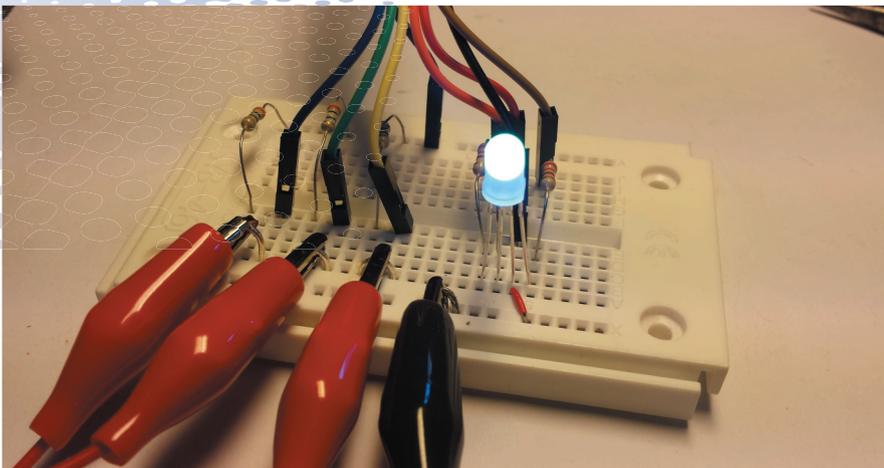
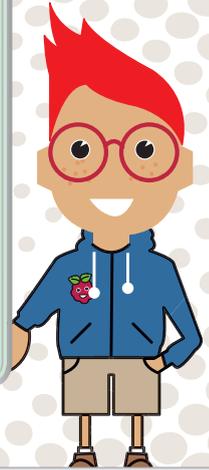


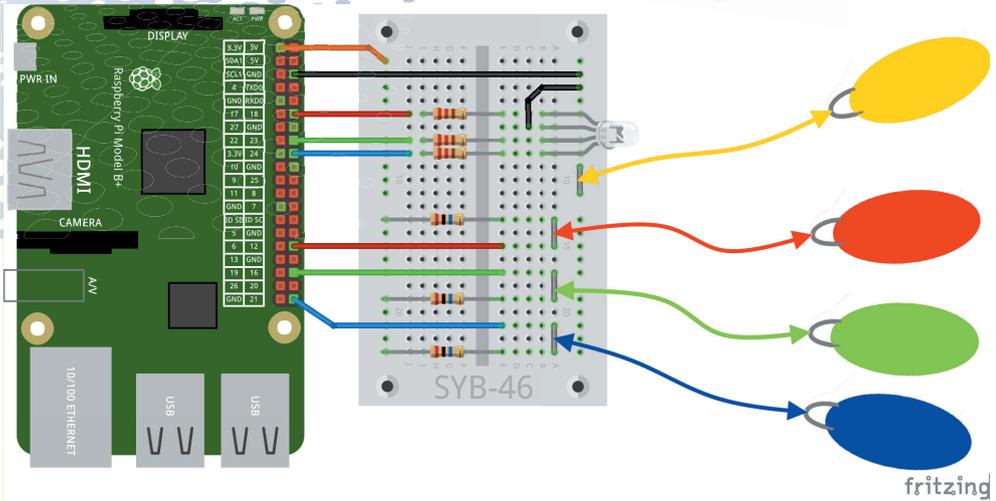
Die drei Farben einer RGB-LED lassen sich einzeln oder gemeinsam ansteuern. So können, wie bereits zu Beginn dieses Buchs beschrieben, verschiedene Farben erzeugt werden. In diesem Experiment werden wir mit drei Knetkontakten eine RGB-LED steuern.

Baue die abgebildete Schaltung auf einem Steckbrett auf. Achte darauf, dass die RGB-LED richtig herum eingebaut ist. Die flache Seite ist oben. Für die Verbindung der Kathode mit der Masseleiste des Steckbretts brauchst du eine Drahtbrücke. Achte dar-

## BENÖTIGTE BAUTEILE

- 1x Steckbrett
- 1x RGB-LED
- 3x 220-Ohm-Widerstand (rot-rot-braun)
- 3x 20-MOhm-Widerstand (rot-schwarz-blau)
- 8x Verbindungskabel
- 5x Drahtbrücke
- 4x Knetkontakt





Die Schaltung mit angeschlossenen Knetkontakten

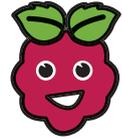
auf, dass der blanke Draht keines der anderen Bauteile berührt, sonst kommt es zu einem Kurzschluss.

Für dieses Experiment brauchst du vier Knetkontakte. Wenn du über den Fußboden gut geerdet bist, reichen auch drei. In der Abbildung sind die Knetkontakte in den Farben dargestellt, die die entsprechenden Farben der RGB-LED steuern. Der Kne-

tekontakt für die Masseleitung ist hier gelb, was in der RGB-LED nicht vorkommt. Grüne Knete kannst du übrigens gut aus gelber und blauer mischen.

Nachdem du die Schaltung aufgebaut und die Knetkontakte angeschlossen hast, starte das Programm **rgb-led**.

# 3 RGB-LED mit Knete-kontakten steuern



```
Wenn angeklickt
  sende config24out config23out config18out an alle
  sende config21inpullnone config16inpullnone config12inpullnone an alle
  wiederhole fortlaufend
    falls Wert von Sensor gpio21 = 0
      sende gpio24on an alle
    sonst
      sende gpio24off an alle
    falls Wert von Sensor gpio16 = 0
      sende gpio23on an alle
    sonst
      sende gpio23off an alle
    falls Wert von Sensor gpio12 = 0
      sende gpio18on an alle
    sonst
      sende gpio18off an alle
    warte 0.1 Sek.
```

Halte mit einer Hand den Knete-kontakt an der Masseleitung und berühre dann mit der anderen Hand nacheinander je einmal den roten, den grünen und den blauen Knete-

kontakt. Die RGB-LED leuchtet in der entsprechenden Farbe. Wenn du mehrere Kontakte gleichzeitig berührst, erzeugst du Mischfarben.



## SO FUNKTIONIERT DAS PROGRAMM

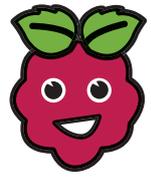
Wie in jedem Scratch-Programm, das Hardware ansteuert, werden zuerst die GPIO-Pins eingerichtet. Wir brauchen dieses Mal drei Ausgänge für die drei Farben der RGB-LED und drei Eingänge für die Knetekontakte.

Berührst du einen Knetekontakt, bekommt der entsprechende Sensor den Wert **0**, und der GPIO-Pin der zugehörigen Farbe für die RGB-LED wird eingeschaltet. Berührst du den Knetekontakt nicht, wird der GPIO-Pin ausgeschaltet.



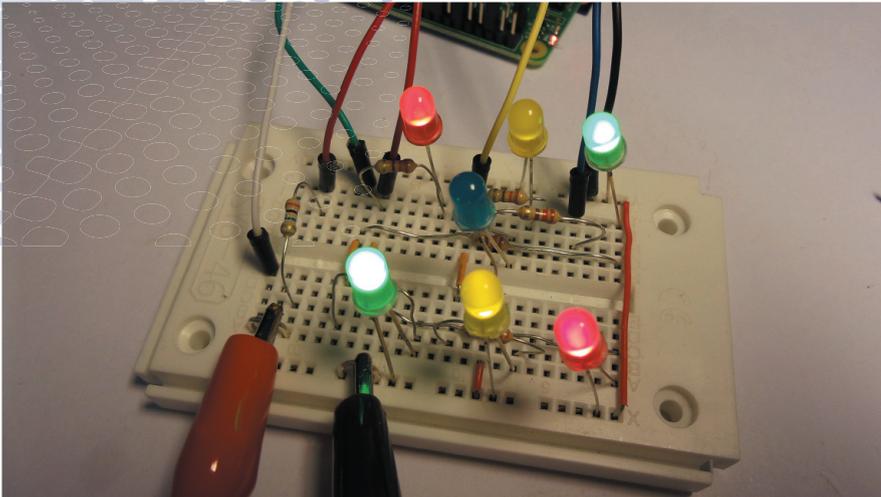
Das eigentliche Programm startet wieder mit einer **wiederhole fortlaufend**-Schleife. Sie enthält drei gleichartige **falls...sonst**-Blöcke, die jeweils einen der Knetekontakte abfragen.





Die typischen Spielwürfel mit einem bis sechs Augen kennt jeder, und vermutlich hat jeder welche zu Hause. Wesentlich cooler ist ein elektronisch gesteuerter Würfel, der mit einer Steuerung über einen Knehtkontakt die Augen leuchten lässt – aber nicht einfach eine bis sechs LEDs in einer Reihe, sondern in der Anordnung eines Spielwürfels.

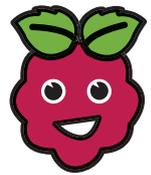
Spielwürfel haben Augen in der typischen quadratischen Anordnung. Dazu braucht man sieben LEDs, die in vier Gruppen gesteuert werden, wie die Abbildung zeigt. Für die Ansteuerung der LEDs brauchen wir nur vier statt sieben GPIO-Pins, da ein Würfel zur Darstellung gerader Zahlen die Augen paarweise nutzt.





Würfelzahl	GPIO 18	GPIO 25	GPIO 24	GPIO 23

Die Würfelaugensymbole mit LED-Gruppen dargestellt



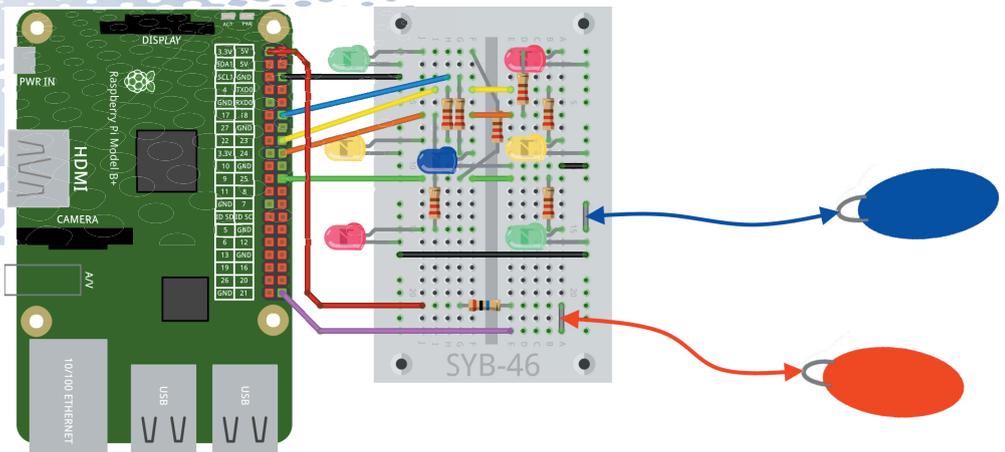
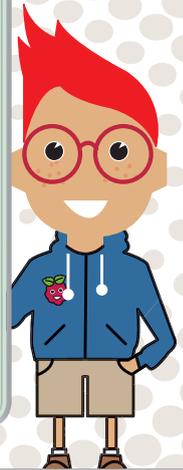
Baue die Schaltung aus sieben LEDs, Vorwiderständen und einem Knetekontakt auf.

Die Schaltung sieht auf den ersten Blick deutlich komplizierter aus als die vorherigen. Immerhin werden 15 elektronische Bauteile und zusätzlich diverse Drähte auf dem Steckbrett aufgebaut.

Mache es wie die Elektronikprofis, die Hunderte von Bauteilen auf Platinen löten: Fange mit den untersten Bauteilen an und arbeite dich

## BENÖTIGTE BAUTEILE

- 1x Steckbrett
- 2x LED rot
- 2x LED grün
- 2x LED gelb
- 1x LED blau
- 7x 220-Ohm-Widerstand (rot-rot-braun)
- 1x 20-MΩm-Widerstand (rot-schwarz-blau)
- 7x Verbindungskabel
- 7x Drahtbrücken (unterschiedliche Längen)
- 2x Knetekontakt



fritzing

LED-Würfel am Raspberry Pi mit Knetekontakten



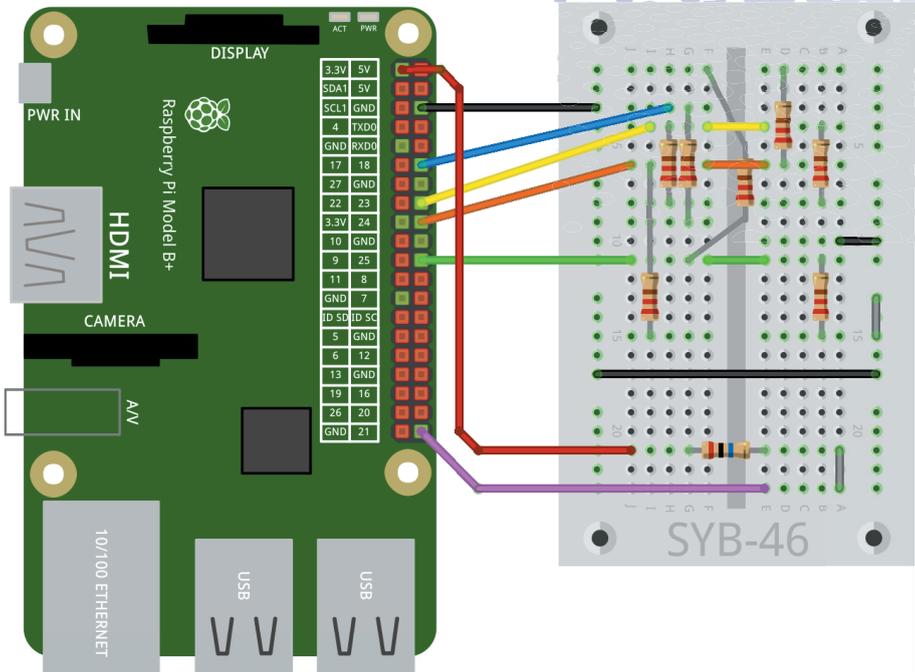
nach oben hoch. Das bedeutet in diesem Fall Folgendes:

**1** Stecke zuerst die Widerstände und die kurzen Drahtbrücken. Achte dabei darauf, dass die blanken Drahtbrücken keine Bauteile berühren. Wenn du in deiner Bastelkiste isolierten Schaltdraht hast, kannst du natürlich auch diesen für die farbig dargestellten Drahtbrücken verwenden. Nur die grauen Drähte müssen

blank sein, weil hier die Krokodilklemmen angeklemt werden.

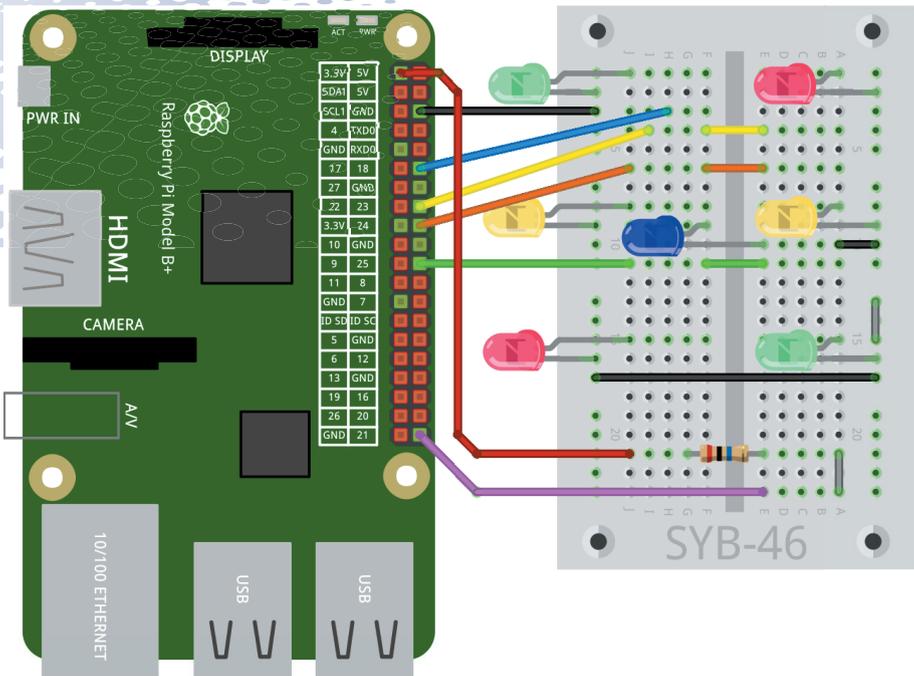
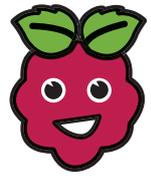
**2** Stecke erst danach die LEDs. Achte darauf, dass die Drähte der LEDs keine Widerstände berühren.

**3** Schließe als Letztes die Verbindungskabel zwischen Raspberry Pi und Steckbrett sowie die Knetkontakte an.



fritzing

Diese Abbildung zeigt nur die Widerstände und Drahtbrücken.



fritzing

Diese Abbildung zeigt zur Übersicht die LEDs ohne die Vorwiderstände, die jetzt schon auf dem Steckbrett aufgebaut sind.

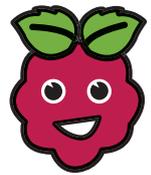
Nachdem du die Schaltung aufgebaut und die Knetkontakte angeschlossen hast, starte das Programm **led-wuerfel**.

Alle LEDs sind zunächst ausgeschaltet. Berühre kurz die beiden Knete-

kontakte, um zu würfeln. Wenn du wieder loslässt, wird die gewürfelte Zahl so lange angezeigt, bis du die Knetkontakte erneut berührst.



```
Wenn  angeklickt
  sende config18out config23out config24out config25out an alle
  sende config21inputnone an alle
  wiederhole fortlaufend
    falls Wert von Sensor gpio21 = 0
      sende gpio18off gpio23off gpio24off gpio25off an alle
      setze w auf Zufallszahl von 1 bis 6
      falls w = 1
        sende gpio18on an alle
      falls w = 2
        sende gpio25on an alle
      falls w = 3
        sende gpio18on gpio24on an alle
      falls w = 4
        sende gpio24on gpio25on an alle
      falls w = 5
        sende gpio18on gpio24on gpio25on an alle
      falls w = 6
        sende gpio23on gpio24on gpio25on an alle
      warte 0.5 Sek.
```



## SO FUNKTIONIERT DAS PROGRAMM

Wenn du auf das grüne Fähnchen klickst, werden als Erstes die fünf verwendeten GPIO-Pins initialisiert. Wir verwenden die Pins 18, 23, 24 und 25 als Ausgänge für die LEDs und den Pin 21 als Eingang für den Knetekontakt.



Wenn dieser GPIO-Pin den Wert **0** hat, werden die Anweisungen innerhalb des **falls**-Blocks ausgeführt. Als Erstes schalten wir die vier LEDs aus. Zu Beginn sind die LEDs alle ausgeschaltet, später bleibt ein angezeigtes Würfelergebnis so lange bestehen, bis du wieder die Knetkontakte berührst.

Danach wird eine zufällige Zahl zwischen 1 und 6 erzeugt und in der

```

Wenn  angeklickt
  sende config18out config23out config24out config25out ▼ an alle
  sende config21inputnone ▼ an alle
  
```

Danach beginnt die Endlosschleife, die darauf wartet, dass du die Knetkontakte berührst.

```

sende gpio18off gpio23off gpio24off gpio25off ▼ an alle
  
```

```

Wenn  angeklickt
  sende config18out config23out config24out config25out ▼ an alle
  sende config21inputnone ▼ an alle
  wiederhole fortlaufend
    falls Wert von Sensor gpio21 ▼ = 0
      
```



Variablen **w** gespeichert. Variablen sind kleine Speicherplätze, in denen man sich, während ein Programm läuft, eine Zahl oder irgendetwas anders merken kann. Wenn das Programm beendet wird, werden diese Variablenspeicher automatisch wieder geleert.

gewürfelte Zahl und kannst leicht kontrollieren, ob die LEDs funktionieren. Dieses Zahlenfeld ist allerdings sehr klein. Klicke mit der rechten Maustaste darauf und wähle aus dem Menü **Großanzeige** aus. Dann ist die Zahl besser zu erkennen.

setze **w** auf Zufallszahl von **1** bis **6**

Variablen müssen in Scratch zunächst einmal angelegt werden, bevor man sie benutzen kann. Klicke in der Blockpalette oben auf das orangefarbene Symbol **Variablen** und dann auf **Neue Variable**.

Neue Variable

Gib der Variablen einen Namen. In unserem Beispiel verwenden wir einfach **w** (wie Würfel). In der Blockpalette werden jetzt verschiedene Blöcke zur Arbeit mit Variablen angezeigt.

Wenn du den Schalter links neben den Variablen **w** einschaltest, wird diese Variable automatisch auf der Bühne bei der Katze in einem kleinen orangefarbenen Feld angezeigt. So siehst du hier immer die

Bewegung Steuerung  
Aussehen Fühlen  
Klang Operatoren  
Malstift Variablen

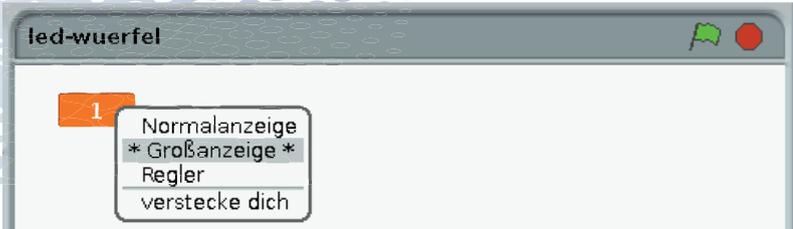
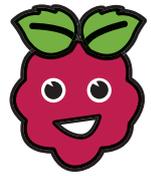
Neue Variable  
Variable löschen

**w**

setze **w** auf **0**  
ändere **w** um **1**  
zeige Variable **w**  
verstecke Variable **w**

Neue Liste

# 4 LED-Würfel



Ziehe den **setze...auf**-Block in das Skriptfenster. Solange nur eine Variable definiert ist, ist sie im Block automatisch ausgewählt. Wenn du später in einem Programm mehrere Variablen verwendest, musst du im Listenfeld die richtige auswählen.

Ziehe dann aus der grünen Blockpalette der Operatoren den Block **Zufallszahl von...bis...** auf das Zahlenfeld im orangefarbenen **setze...auf**-Block. Trage in die beiden Zahlenfelder eine **1** und eine **6** ein, da die Zufallszahl in diesem Bereich liegen soll.



Baue jetzt den Block in das Programm ein. Dies sollte so aussehen:





Nachdem die Zahl gewürfelt wurde, folgen sechs **falls**-Blöcke für jeden möglichen Würfelwert.

Mit einem Rechtsklick auf den **falls**-Block kannst du diesen duplizieren. Nun brauchst du nur noch die



Wenn eine bestimmte Zahl gewürfelt wurde, schaltet jeder dieser Blöcke die entsprechende Kombination von LEDs ein.

Ziehe den grünen **=**-Block in das Abfragefeld des **falls**-Blocks.



Ziehe dann den Block der Variablen **w** aus der Blockpalette **Variablen** in das erste der beiden kleinen weißen Felder im grünen Block. In das zweite Feld schreibe eine **1**.

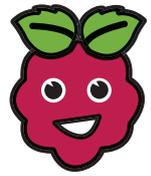


Jetzt wird der Block innerhalb der Klammer immer dann abgearbeitet, wenn das Würfelergebnis eine **1** war. Setze innerhalb des **falls**-Blocks einen Block **sende...an alle** und gib dort den GPIO-Befehl **gpio18on** ein, um die mittlere LED einzuschalten, also die 1 auf dem Würfel.

Würfelergebnisse und die passenden LEDs zu ändern.

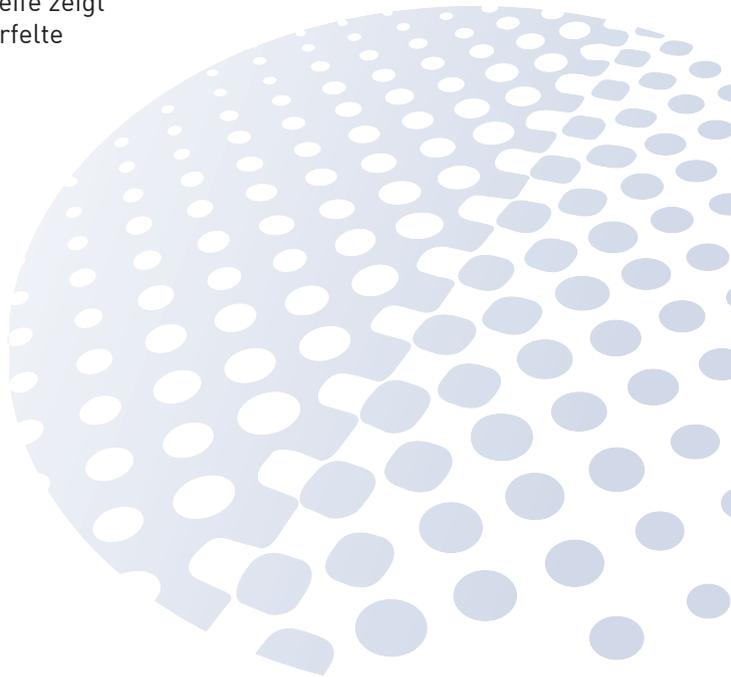


Unabhängig vom Würfelergebnis, soll das Programm immer eine halbe Sekunde lang warten, nachdem mit dem Knetekontakt neu gewürfelt wurde, um zu vermeiden, dass durch sogenanntes Tastenprellen kurz hintereinander zwei verschiedene Würfelaktionen ausgelöst werden.



```
falls w = 6
  sende gpio23on gpio24on gpio25on an alle
  warte 0.5 Sek.
  ↪
```

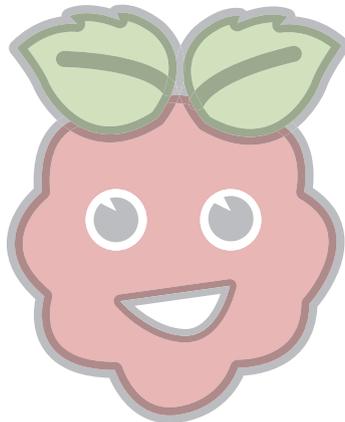
Danach startet die Endlosschleife einen neuen Durchlauf. Je nach ermitteltem Sensorwert wird die Würfelaktion gestartet oder nicht. Solange der Knetekontakt nicht berührt wird, hat der Sensor den Wert **1**, und die Endlosschleife zeigt weiterhin das zuletzt gewürfelte Ergebnis an.



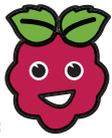


## WIE ENTSTEHEN EIGENTLICH ZUFALLSZAHLN?

Man könnte annehmen, in einem Programm passiere nichts zufällig, und alles sei geplant. Wie aber kann ein Programm dann in der Lage sein, zufällige Zahlen zu generieren? Teilt man eine große Primzahl durch irgendeinen Wert, ergeben sich ab der x-ten Nachkommastelle Zahlen, die kaum noch vorhersehbar sind. Sie ändern sich auch dann ohne jede Regelmäßigkeit, wenn man den Divisor regelmäßig erhöht. Dieses Ergebnis ist zwar scheinbar zufällig, lässt sich aber durch ein identisches Programm oder mehrfachen Aufruf des gleichen Programms jederzeit reproduzieren. Nimmt man aber eine aus mehreren dieser Ziffern zusammengesetzte Zahl und teilt sie wiederum durch eine Zahl, die sich aus der aktuellen Uhrzeitsekunde oder dem Inhalt einer beliebigen Speicherstelle des Rechners ergibt, kommt ein Ergebnis heraus, das sich nicht reproduzieren lässt. Es wird deshalb als „Zufallszahl“ bezeichnet.



# 5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern

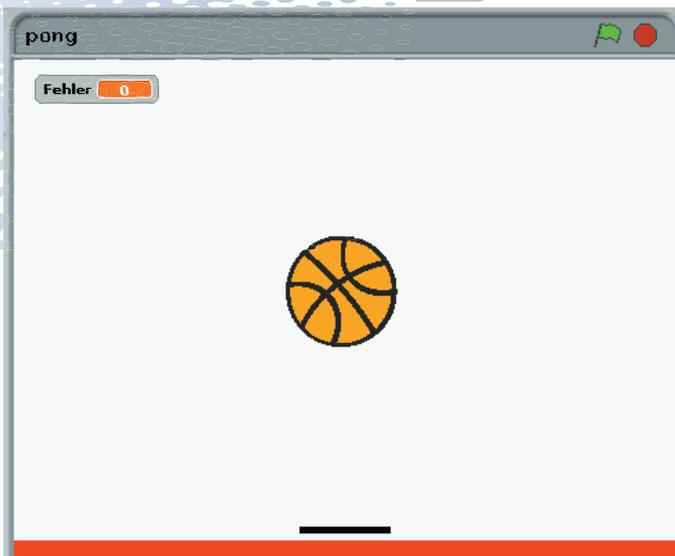


Natürlich kann man mit Scratch nicht nur LEDs programmieren. Ursprünglich war die Programmiersprache dazu gedacht, einfache Bildschirm-Spiele zu programmieren. Ein solches Spiel auf dem Bildschirm einfach mit Knetekontakten steuern – mit unserem nächsten Projekt wird das möglich.

Wir werden ein klassisches Pong-Spiel programmieren, wie man es von diversen Retro-Konsolen kennt, und es über zwei Knetekontakte steuern. In die-

## BENÖTIGTE BAUTEILE

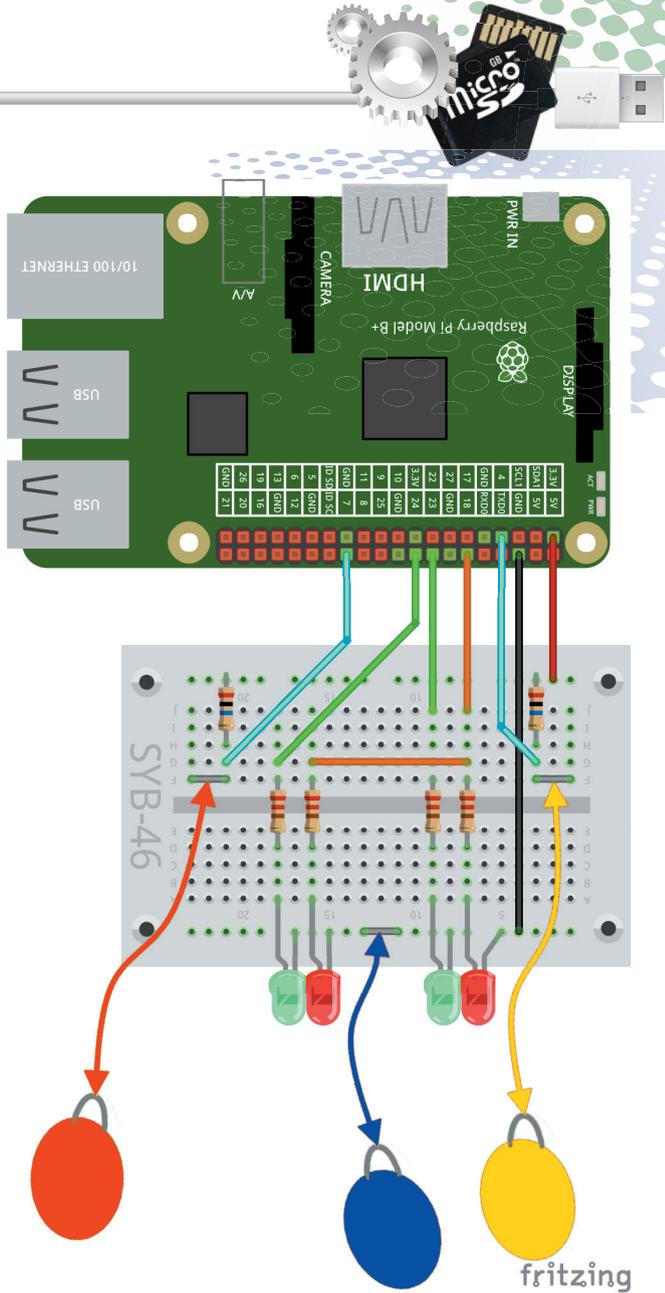
- 1x Steckbrett
- 2x LED rot
- 2x LED grün
- 4x 220-Ohm-Widerstand (rot-rot-braun)
- 2x 20-MOhm-Widerstand (rot-schwarz-blau)
- 7x Verbindungskabel
- 4x Drahtbrücken (unterschiedliche Längen)
- 3x Knetekontakt



Das Pong-Spiel in Scratch

sem Spiel versuchst du, einen Ball, der im Raum umherfliegt, mit einem Paddle (einer Art Schläger) zurückzuschlagen. Wenn der Ball die farbige Linie am unteren Rand berührt, bekommst du einen Minuspunkt, und der Ball startet in der Spielfeldmitte wieder neu. Mit zwei Knetkontakten kannst du das Paddle nach links und rechts bewegen.

Baue die abgebildete Schaltung auf einem Steckbrett auf. Achte darauf, dass die LEDs richtig herum eingebaut sind. Die flachen Seiten sind rechts. Die beiden grünen LEDs zeigen an, wenn einer der Knetkontakte berührt wird. Bei einem Fehler leuchten die beiden roten LEDs kurz auf. Dazu sind sie mit dem gleichen GPIO-Pin 18 verbunden. Zusätzlich werden die Töne abgespielt. All das ist für das Spiel



Gamepad aus Knete, am Raspberry Pi angeschlossen

# 5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern



nicht wichtig, zeigt dir aber in der Hektik des Spiels, ob die Knetkontakte wirklich funktionieren. Außerdem sieht es natürlich cooler aus.

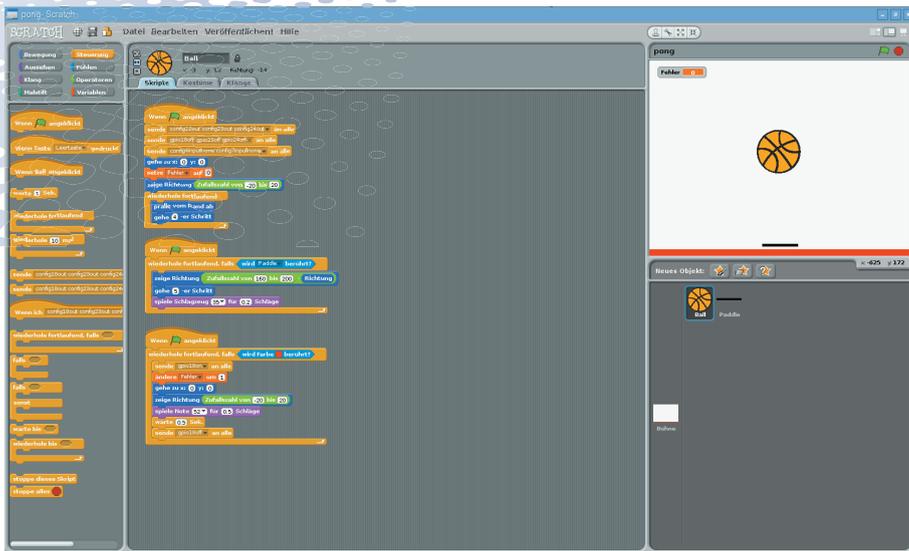
Forme drei Knetkontakte für die beiden Richtungen nach links und rechts sowie für den Massekontakt. Nachdem du die Schaltung aufgebaut und die Knetkontakte angeschlossen hast, starte das Programm **pong**.

In diesem Programm verwenden wir nicht die Katze, sondern eigene Objekte, denen bestimmte Funktionsblöcke zugeordnet werden. Dabei

wirst du auch einige neue Programmier-techniken kennenlernen.

Wenn du auf das grüne Fähnchen klickst, fliegt der Ball los, und du kannst mit den Knetkontakten das Paddle nach links und rechts bewegen. Jedes Mal, wenn der Ball gegen die rote untere Kante stößt, gibt es einen Minuspunkt, und der Ball fliegt wieder von der Mitte los.

Was du jetzt auf dem Bildschirm siehst, ist noch nicht das ganze Programm. Selbst bei genauerem Hin-



Die Programmblöcke für den Ball im Scratch-Programm **pong**



# 5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern



cher angeschlossen werden, da das Audiosignal nicht über das DVI-Kabel übertragen wird. Klicke mit der rechten Maustaste rechts oben auf das Lautsprechersymbol, um den gewünschten Audioausgang auszuwählen.



## SO FUNKTIONIERT DAS PROGRAMM

Da dieses Programm außer interessanten, neuen Programmblöcken auch einen eigenen Hintergrund und zwei eigene Objekte enthält, werden wir es Schritt für Schritt, sozusagen aus dem Nichts, aufbauen.



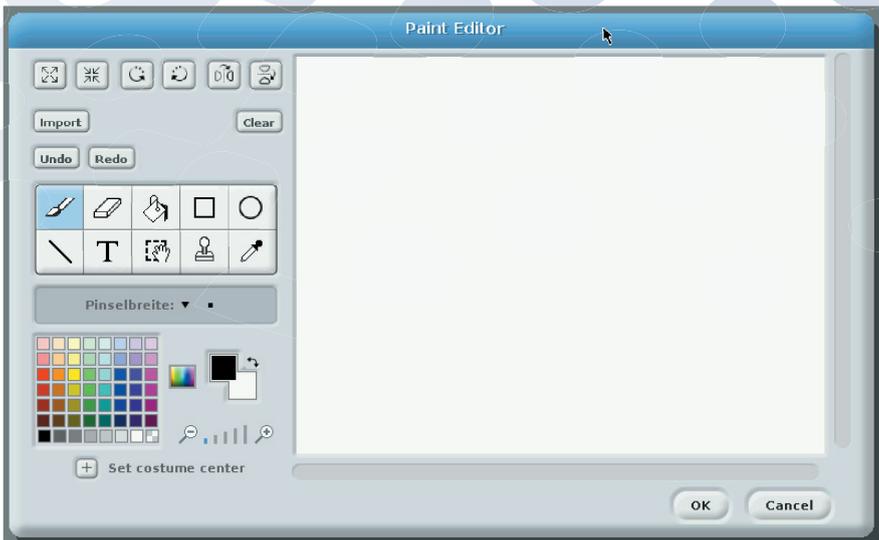
## DER HINTERGRUND

1 Klicke in Scratch oben auf das Menü **Datei** und wähle dort **Neu** aus, um mit einem ganz neuen Programm anzufangen.

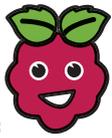
2 Unten rechts findest du die Liste aller Objekte im Programm. Hier sind zunächst nur die Katze und ein weißes Objekt **Bühne** zu sehen. Es bezeichnet das weiße Fenster oben rechts, auf dem sich die Katze oder ein anderes Objekt bewegt. Klicke in der Objektliste auf das Feld **Bühne**. Die obere Leiste des Programmfensters verändert sich und zeigt die **Bühne** an. Alle Programmblöcke in diesem Fenster gelten immer für das hier angezeigte Objekt.

3 Klicke auf die Registerkarte **Hintergründe**, um den Hintergrund der Bühne zu bearbeiten.

4 Ein Scratch-Programm kann mehrere Hintergründe haben, die durch das Programm ausge-



# 5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern



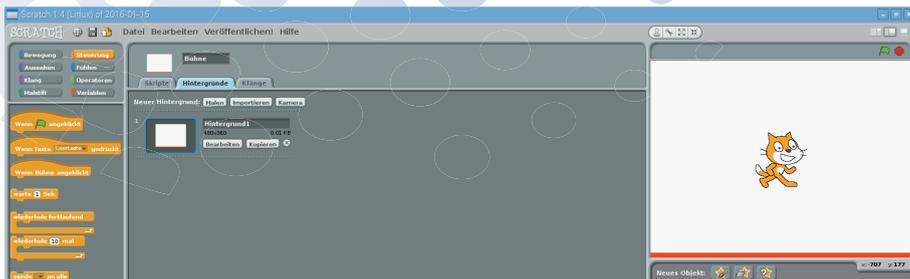
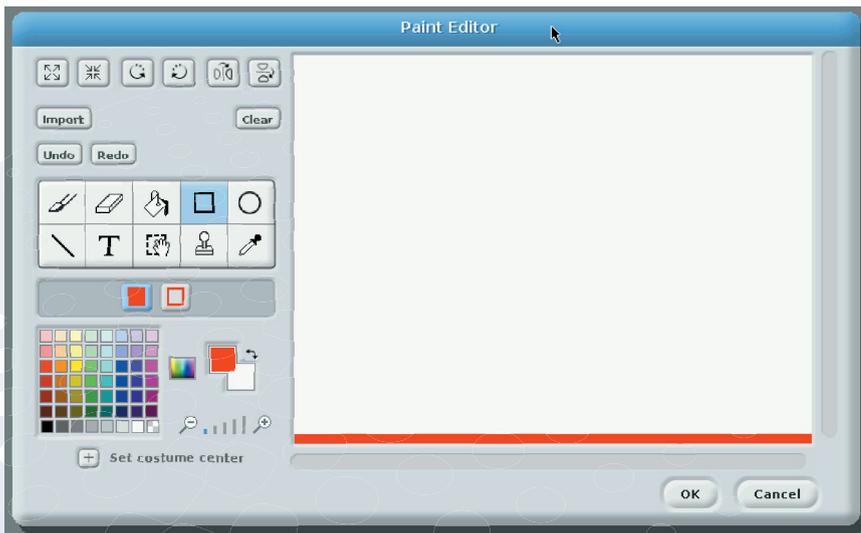
tauscht werden. Das brauchen wir hier nicht. Klicke einfach unter **Hintergrund1** auf **Bearbeiten**, um den vorhandenen weißen Hintergrund zu verändern.

**5** Scratch beinhaltet ein einfaches Malprogramm, mit dem du den Hintergrund sowie andere Objekte malen kannst. Wähle das

Rechteck-Werkzeug aus, und zwar die gefüllte Variante.

**6** Wähle dann eine auffällige Farbe aus, z. B. Rot, und male damit am unteren Rand über die gesamte Breite ein schmales Rechteck.

**7** Verlasse das Malprogramm mit **OK**, und der farbige Balken





erscheint auf dem Hintergrund der Bühne.

## DER BALL

1 Als Nächstes brauchen wir den Ball. Natürlich könntest du auch die Scratch-Katze durch den Raum werfen, aber ein Ball sieht echter aus. Lösche also als Erstes die Scratch-Katze, indem du mit der rechten Maustaste auf das Objekt in der Objektliste klickst und dann im Menü **Löschen** auswählst.

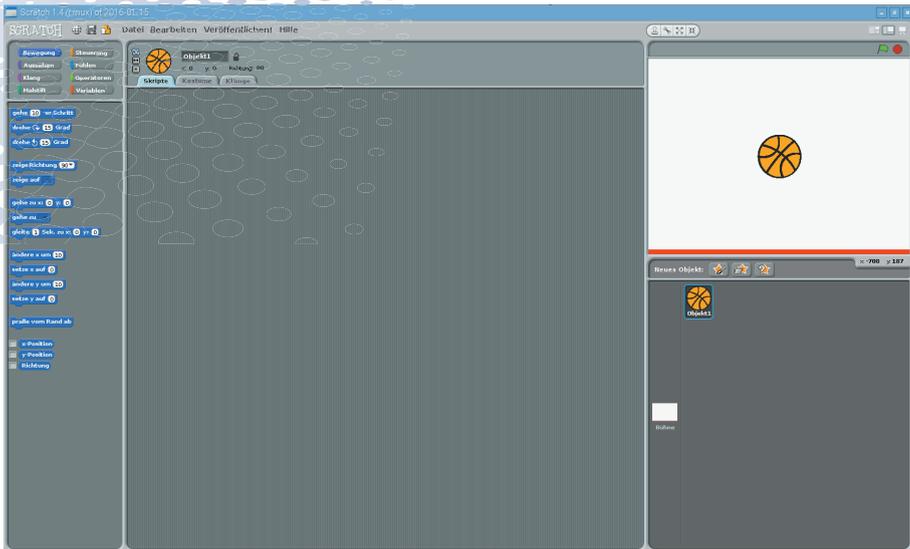
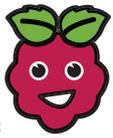
2 Klicke auf den Button **Neues Objekt aus Datei laden**, das ist der mittlere der drei Buttons bei **Neues Objekt**.

3 Wähle im nächsten Fenster unter **Things** den **Basketball** aus und klicke auf **OK**.

4 Der Ball erscheint in der Objektliste, auf der Bühne und, da er gerade ausgewählt ist, auch oben über dem Programmfenster. Alle neu zusammgebauten Programmblöcke gelten also jetzt für den Ball.



# 5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern



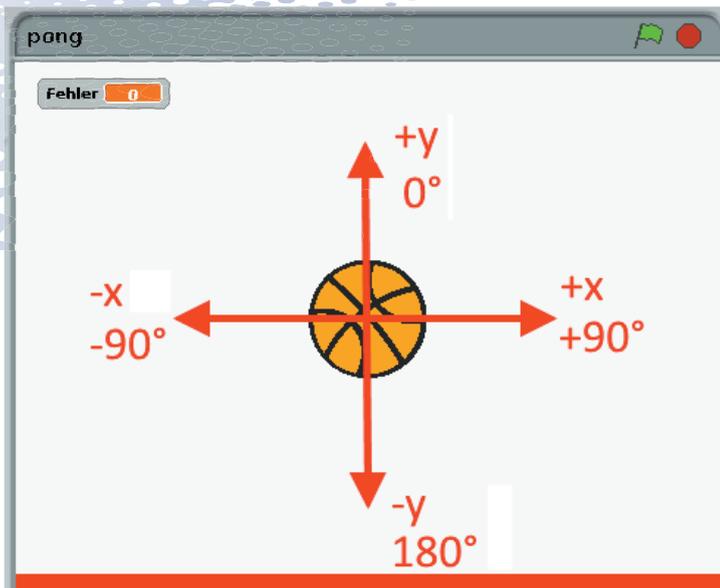


5 Als Nächstes erhält der Ball die Programmblöcke, nach denen er sich bewegen soll. Schalte dazu auf die gelbe Blockpalette **Steuerung** und baue die Blöcke zur Initialisierung der fünf verwendeten GPIO-Pins zusammen, wie schon aus früheren Programmen bekannt.

6 Beim Start des Programms soll der Ball immer in der Mitte liegen. Die Blöcke auf der blauen Blockpalette **Bewegung** steuern die Bewegung von Scratch-Objekten. Wähle hier den Block **gehe zu x:...y:...** und trage in beide Zahlenfelder eine **0** ein.



7 Der Punkt **x:0, y:0** ist der Mittelpunkt des Koordinaten-



Koordinatenachsen und Winkelangaben in Scratch

# 5 Ein Scratch-Spiel mit einem Gamepad auf Knete steuern



systems auf der Scratch-Bühne. Die positive x-Achse läuft nach rechts, die negative nach links. Die positive y-Achse läuft nach oben, die negative nach unten.

8 Außerdem sollen beim Programmstart die Fehler aus dem letzten Spiel auf **0** zurückgesetzt werden. Zum Zählen der Fehler brauchen wir eine Variable. Lege dazu auf der Blockpalette **Variablen** eine neue Variable mit dem Namen **Fehler** an und füge einen Block **setze Fehler auf 0** an das Programm an.



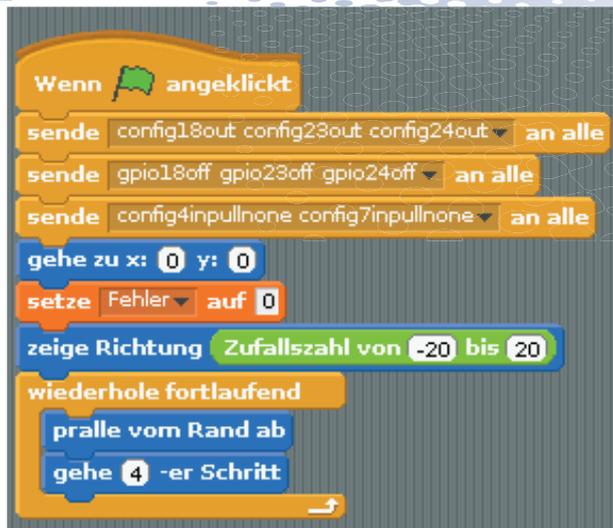
9 Aktiviere das Kästchen neben der Variablen **Fehler** auf der Blockpalette **Variablen**, damit die Fehler im Spiel angezeigt werden.

10 Der Ball soll in einem zufälligen Winkel nach schräg oben losfliegen. Dazu wird die Richtung mit

einem blauen Block **zeige Richtung...** auf einen zufälligen Wert zwischen **-20** und **20** gesetzt.



11 Anschließend wird die Bewegung des Balls fortlaufend wiederholt. Er prallt vom Rand ab, sollte er ihn berühren. Andernfalls fliegt er einen Vierer-Schritt in die eingestellte Richtung. Diese Bewegung wiederholt sich theoretisch endlos.



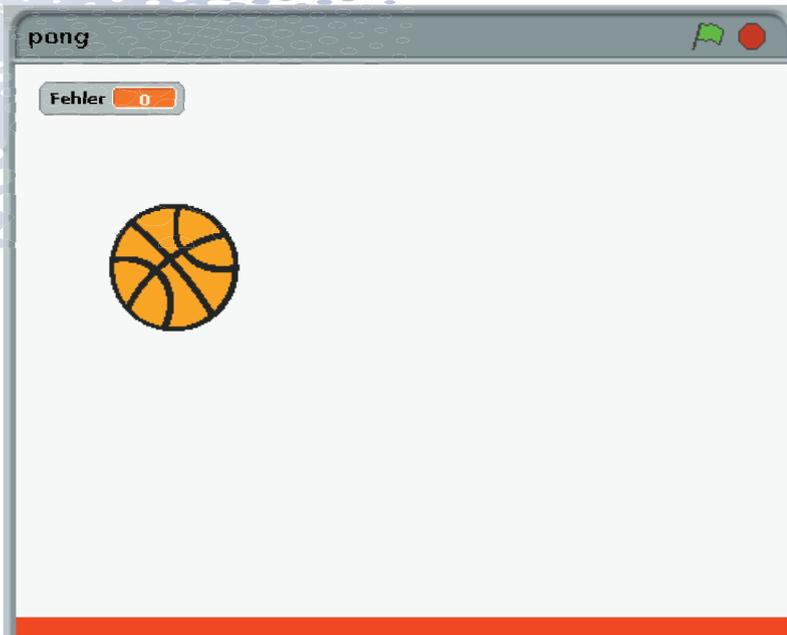


**12** Probiere es aus und starte das Programm mit dem grünen Fähnchen. Der Ball fliegt gleichmäßig durch den Raum und prallt an allen vier Rändern ab. Die rote Kante unten unterscheidet sich nicht von den anderen Rändern.



## DAS PADDLE

**1** Als Nächstes zeichnen wir das Paddle, mit dem du den Ball zurückspielst. Klicke dazu in der Objektpalette unten auf den Button *Neues Objekt malen*. Das Paddle werden wir nicht als fertiges Objekt

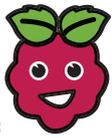


**13** Stoppe den Ball, um an dem Programm weiterzuarbeiten.



importieren, sondern schnell selbst zeichnen.

# 5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern

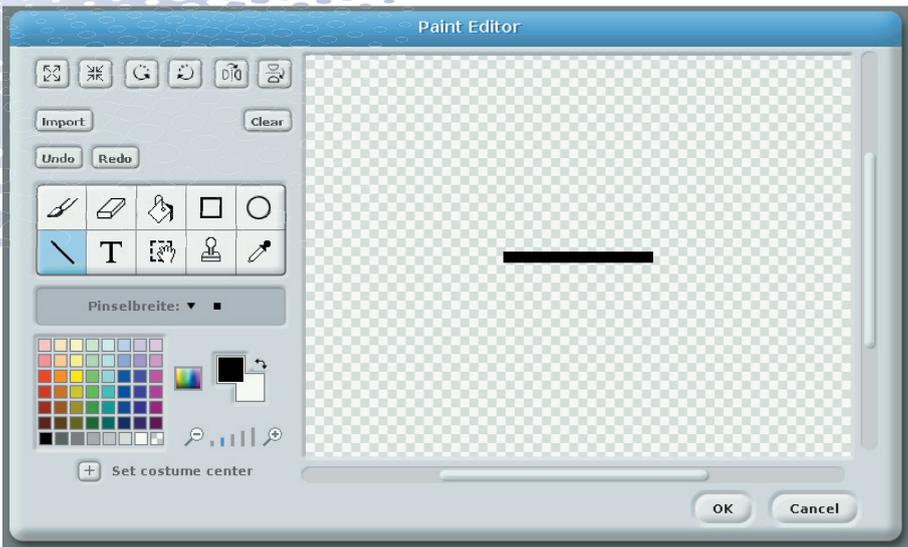


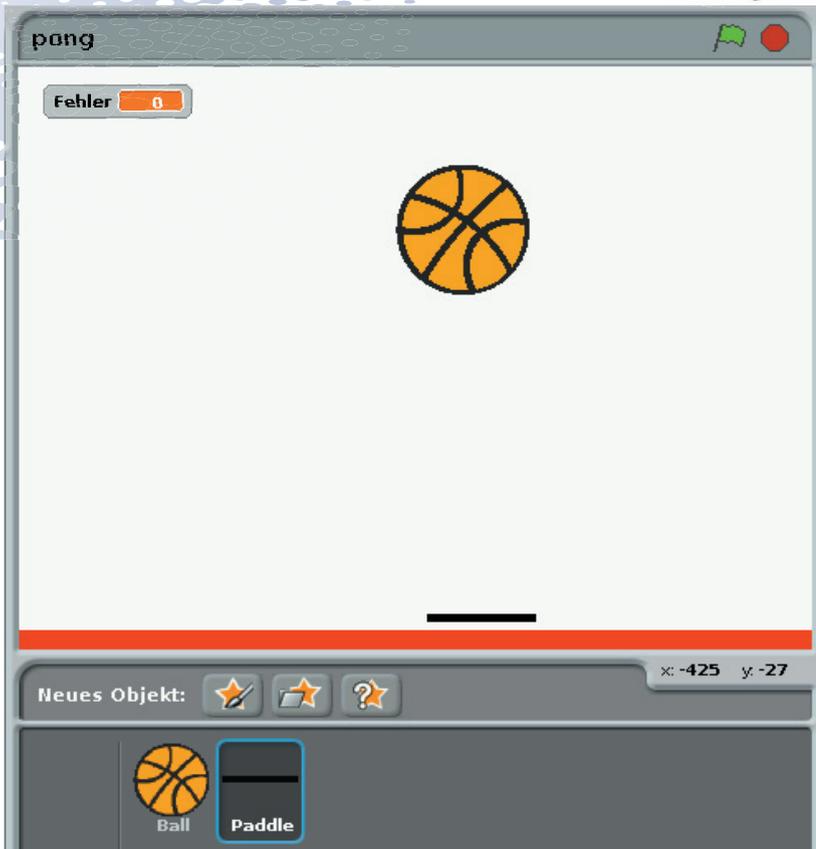
2 Hier wird das gleiche Malprogramm wie für den Hintergrund verwendet. Wähle das Werkzeug **Linie** und eine mittlere Pinselstärke aus. Zeichne dann eine kurze waagerechte Linie, wie in der Abbildung unten zu sehen.

3 Nach einem Klick auf **OK** erscheint das neue Objekt in

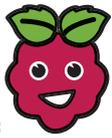
der Mitte der Bühne. Schiebe es an den unteren Rand, kurz über den roten Balken. In der Objektpalette unten taucht es als **Objekt2** auf.

4 Gib den Objekten sinnvolle Namen. Du kannst den Namen des ausgewählten Objekts oberhalb des Programmfensters ändern. Das Paddle soll **Paddle** heißen und der Ball **Ball**.





# 5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern



5 Das Paddle wird über die Knetkontakte gesteuert und braucht dazu ein Programm. Wenn du das Paddle in der Objektpalette auswählst, ist das Programmfenster noch leer. Baue ein kleines Programm, das automatisch startet, wenn du auf das grüne Fähnchen klickst. Dieses Programm setzt als erstes das Paddle auf die Position: **0,-160**. Dadurch steht das Paddle zu Beginn genau in der Mitte zwischen dem linken und dem rechten Rand der Bühne.



6 Anschließend beginnt eine Endlosschleife, die die Knetkontakte abfragt.



7 Baue jetzt eine **falls**-Abfrage, die prüft, ob der Wert des Sensors **gpio7 = 0** ist, also ob der Knetkontakt für die Bewegung nach links berührt wird.



8 Wenn du diesen Knetkontakt berührst, soll die grüne LED am GPIO-Pin 24 einmal kurz aufblin-ken. Dazu schalten wir sie mit einem **sende...an alle**-Block ein und 0,1 Sekunden später wieder aus.



9 Viel wichtiger, als dass die LED kurz leuchtet, ist die Bewegung des Paddles um 20 Koordinateneinheiten nach links. Dazu gibt es auf der blauen Blockpalette **Bewegung** einen Block **ändere x um...**



```

ändere x um -20
falls x-Position < -240
  setze x auf -240
  
```

10 Falls das Paddle am linken Rand angekommen ist, soll es sich nicht weiter bewegen. Hier verwenden wir eine **falls**-Abfrage, die prüft, ob die x-Position kleiner ist als -240. Ist dies der Fall, wird die x-Position einfach auf -240 gesetzt.

11 Schiebe diese Blöcke alle in die **falls**-Abfrage mit dem Block **Wert von Sensor...**. Jetzt kannst du die komplette Abfrage duplizieren, um auch den anderen Knetekontakt am GPIO-Pin 4 abzufragen.

12 Die Werte für die GPIO-Pins des zweiten Knetekontakts sowie der zweiten grünen LED kannst du direkt in den duplizierten Blöcken ändern. Wenn du diesen Knetekontakt berührst, wird das Paddle nach rechts bewegt. Im Block **ändere x um...** muss jetzt also **20** stehen, und die Abfrage danach muss prüfen, ob die y-Position größer als 240 ist. Klicke dazu mit der rechten Maustaste auf den grünen **<**-Operator. Schalte dann im Kontextmenü auf **>** um.

13 Baue nun diese beiden Abfragen in die Hauptschleife des Programms ein.

```

falls Wert von Sensor gpio7 = 0
  sende gpio24on an alle
  warte 0.1 Sek.
  sende gpio24off an alle
  ändere x um -20
  falls x-Position < -240
    setze x auf -240
  
```

```

falls Wert von Sensor gpio4 = 0
  sende gpio23on an alle
  warte 0.1 Sek.
  sende gpio23off an alle
  ändere x um 20
  falls x-Position > 240
    setze x auf 240
  
```



# 5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern



```
Wenn  angeklickt
  gehe zu x: 0 y: -160
  wiederhole fortlaufend
    falls Wert von Sensor gpio7 = 0
      sende gpio24on an alle
      warte 0.1 Sek.
      sende gpio24off an alle
      ändere x um -20
      falls x-Position < -240
        setze x auf -240
    falls Wert von Sensor gpio4 = 0
      sende gpio23on an alle
      warte 0.1 Sek.
      sende gpio23off an alle
      ändere x um 20
      falls x-Position > 240
        setze x auf 240
```

 **Starte jetzt das Programm** mit einem Klick auf das grüne Fähnchen. Du kannst das Paddle mit den Knetekontakten steuern. Der Ball fliegt aber davon völlig unberührt weiterhin durch den Raum.

## DIE SPIELREGELN

Nachdem die Spielmechanik funktioniert, braucht das Spiel noch seine Regeln. In diesem einfachen Spiel gibt es nur zwei Regeln:

 **Schlägt der Ball gegen das Paddle**, wird er zurückgeschlagen, wobei es eine leichte, zufällige Veränderung der Flugbahn geben soll, damit der Ball nicht immer völlig gleichförmig durch den Raum fliegt. Sonst könnte man als



Spieler das Paddle einfach an einer bestimmten Stelle stehen lassen.

1 Schlägt der Ball gegen den lilafarbenen Balken am rechten Rand (weil der Spieler ihn mit dem Paddle verfehlt hat), gibt es einen Minuspunkt, und der Ball startet in der Mitte des Spielfelds neu.

Diese beiden Regeln werden in zwei zusätzlichen Programmblöcken für den Ball definiert, die beide automatisch mit dem Spiel starten.

1 Wähle auf der Objektpalette rechts unten den Ball aus. Die neuen Programmblöcke betreffen dieses Objekt. Im Programmfenster ist nun wieder das Programm zu sehen, das die Ballbewegung steuert.

2 Baue hier eine neue Gruppe von Programmblöcken, die starten soll, sobald das grüne Fähnchen angeklickt wird. Danach folgt ein **wiederhole fortlaufend falls**-Block, der nur ausgeführt wird, wenn das aktuelle Objekt, nämlich der Ball, das Paddle berührt. Für solche Fälle gibt es auf der Blockpalette **Fühlen** den Block



**wird...berührt.** Wähle hier im Listenfeld das Objekt **Paddle** aus.

3 Wenn der Ball das Paddle berührt, wird die Bewegungsrichtung ins Negative umgekehrt. Der Ball fliegt im gleichen Winkel nach links unten weiter, in dem er von links oben kam – oder umgekehrt, wenn er von links unten kam, fliegt er nach links oben weiter. Um die Bewegung etwas unvorhersehbarer zu gestalten, wird die Flugrichtung des Balls gegenüber der bisherigen Richtung leicht verändert. Anstatt die bisherige Richtung von 180 Grad zu subtrahieren, subtrahieren wir sie von einem Zufallswert zwischen 160 und 200. Auf diese Weise ändert sich die Richtung für den Rückflug um einen zufälligen Wert zwischen -20 und 20 Grad.

4 Anschließend wird der Ball einen Fünfer-Schritt bewegt,



# 5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern



damit er das Paddle auf keinen Fall mehr berührt.

gehe 5 -er Schritt

5 Dann wird ein kurzer Ton abge-  
spielt. Ziehe dazu einen Block  
*spiele Schlagzeug ... für ... Schläge*  
von der Blockpalette *Klang* in das  
Programm. Wähle das Instrument **35**  
*Große Trommel (akustisch)* und eine  
Länge von **0.2** Schlägen.

spiele Schlagzeug 35 für 0.2 Schläge

6 Zusammen ergibt das die  
abgebildete Gruppe von Pro-  
grammblocken, die von Anfang an  
mit gestartet wird und nur darauf  
wartet, dass der Ball das Paddle  
berührt. Nur dann werden die Blöcke  
ausgeführt.

7 Nun geht es darum, die zweite  
Spielregel umzusetzen und  
festzulegen, was passiert, wenn der  
Ball den roten Balken am unteren  
Rand der Bühne berührt. Baue dazu  
eine weitere Gruppe von Programm-  
blöcken, die ebenfalls starten soll,  
wenn das grüne Fähnchen angeklickt  
wird.

8 Auch hier kommt wieder ein  
*wiederhole fortlaufend falls...*-  
Block zum Einsatz, der dieses Mal  
nur durchlaufen werden soll, wenn  
der Ball den roten Balken  
berührt. Dafür gibt es auf  
der Blockpalette *Fühlen*  
den *wird Farbe... berührt?*-

Block. Wenn du in diesem Block auf  
das Farbfeld tippst, erscheint eine  
Pipette, mit der du die rote Farbe auf  
der Bühne auswählen kannst.

Wenn  angeklickt

wiederhole fortlaufend, falls  wird berührt?

zeige Richtung Zufallszahl von 160 bis 200 - Richtung

gehe 5 -er Schritt

spiele Schlagzeug 35 für 0.2 Schläge



```
Wenn  angeklickt
wiederhole fortlaufend, falls  wird Farbe  berührt?
```

damit der Ball nicht exakt vom Rand abbrillt, aber trotzdem ungefähr nach oben fliegt.

9 Berührt der Ball den roten Balken, leuchten beide roten LEDs am GPIO-Pin 18 auf, und der Spieler bekommt einen Fehlerpunkt. Die Variable **Fehler** wird um **1** erhöht. Diese Variable wird auf der Bühne angezeigt, d. h., du kannst dort im Spiel jederzeit deine Punkte sehen.

```
sende gpio18on an alle
ändere Fehler um 1
```

10 Anschließend wird der Ball wieder in die Spielfeldmitte auf die Position **x: 0, y: 0** gesetzt.

```
gehe zu x: 0 y: 0
```

11 Auch in diesem Fall wird die neue Richtung zufällig zwischen **-20** und **20** Grad gewählt,

```
zeige Richtung Zufallszahl von -20 bis 20
```

12 Dann wird ein dumpfer Fehler-ton abgespielt. Ziehe dazu einen Block **spiele Note ... für ... Schläge** von der Blockpalette **Klang** in das Programm. Wähle den Ton **52** und eine Länge von **0.5** Schlägen.

```
spiele Note 52 für 0.5 Schläge
```

13 Nach einer Wartezeit von 0,5 Sekunden werden auch die rot leuchtenden LEDs wieder ausgeschaltet, und das Spiel läuft normal weiter.

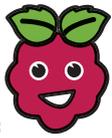
```
warte 0.5 Sek.
sende gpio18off an alle
```

Damit ist das Programm fertig. Wenn du auf das grüne Fähnchen klickst, starten drei



Gruppen von Programmblöcken für den Ball und eine für das Paddle.

# 5 Ein Scratch-Spiel mit einem Gamepad aus Knete steuern



**Ball**  
x: -3 y: 12 Richtung: -14

**Wenn angeklickt**

- sende config18out config23out config24out an alle
- sende gpio18off gpio23off gpio24off an alle
- sende config4inpullnone config7inpullnone an alle
- gehe zu x: 0 y: 0
- setze Fehler auf 0
- zeige Richtung Zufallszahl von -20 bis 20
- wiederhole fortlaufend
  - pralle vom Rand ab
  - gehe 4 -er Schritt

**Wenn angeklickt**

- wiederhole fortlaufend, falls wird Paddle berührt?
  - zeige Richtung Zufallszahl von 160 bis 200 - Richtung
  - gehe 5 -er Schritt
  - spiele Schlagzeug 85 für 0.2 Schläge

**Wenn angeklickt**

- wiederhole fortlaufend, falls wird Farbe berührt?
  - sende gpio18on an alle
  - ändere Fehler um 1
  - gehe zu x: 0 y: 0
  - zeige Richtung Zufallszahl von -20 bis 20
  - spiele Note 52 für 0.5 Schläge
  - warte 0.5 Sek.
  - sende gpio18off an alle

**Paddle**  
x: 0 y: -160 Richtung: 90

**Wenn angeklickt**

- gehe zu x: 0 y: -160
- wiederhole fortlaufend
  - falls Wert von Sensor gpio7 = 0
    - sende gpio24on an alle
    - warte 0.1 Sek.
    - sende gpio24off an alle
    - ändere x um -20
    - falls x-Position < -240
      - setze x auf -240
  - falls Wert von Sensor gpio4 = 0
    - sende gpio23on an alle
    - warte 0.1 Sek.
    - sende gpio23off an alle
    - ändere x um 20
    - falls x-Position > 240
      - setze x auf 240

Das fertige Programm für das Pong-Spiel