

CE

**CONRAD**

## Alle Versuche im Überblick

<b>Raspberry-Pi-Adventskalender 2018</b> .....	<b>3</b>	<b>13. Tag</b> .....	<b>27</b>
<b>1. Tag</b> .....	<b>4</b>	Heute im Adventskalender .....	27
Heute im Adventskalender .....	4	Widerstand .....	27
Den Raspberry Pi vorbereiten .....	4	Sensorkontakt aus Knete .....	27
Betriebssysteminstallation in Kürze .....	4	RGB-LED blinkt zufällig .....	28
Die wichtigsten Bauteile kurz erklärt .....	5	Das Programm .....	28
Steckbrett .....	5	<b>14. Tag</b> .....	<b>30</b>
LEDs .....	5	Heute im Adventskalender .....	30
GPIO-Verbindungskabel .....	5	RGB-LED-Lichteffekte .....	30
Eine LED leuchtet .....	6	Das Programm .....	30
<b>2. Tag</b> .....	<b>7</b>	<b>15. Tag</b> .....	<b>32</b>
Heute im Adventskalender .....	7	Heute im Adventskalender .....	32
Die erste LED leuchtet auf der Weihnachtskrippe .....	7	Lauflicht mit Knetekontakt steuern .....	32
Grundeinstellungen für Scratch .....	7	Das Programm .....	32
Das Programm .....	8	<b>16. Tag</b> .....	<b>34</b>
<b>3. Tag</b> .....	<b>9</b>	Heute im Adventskalender .....	34
Heute im Adventskalender .....	9	Der Raspberry Pi erzeugt Töne .....	34
Zwei LEDs blinken abwechselnd .....	9	Das Programm .....	34
Das Programm .....	9	<b>17. Tag</b> .....	<b>36</b>
<b>4. Tag</b> .....	<b>10</b>	Heute im Adventskalender .....	36
Heute im Adventskalender .....	10	Erweitertes Lauflicht .....	36
Ein Stern schaltet zwei LEDs um .....	10	Das Programm .....	36
Das Programm .....	10	<b>18. Tag</b> .....	<b>37</b>
<b>5. Tag</b> .....	<b>13</b>	Heute im Adventskalender .....	37
Heute im Adventskalender .....	13	Drei verschiedene Lauflichtmuster .....	37
LEDs mit Taster umschalten .....	13	Das Programm .....	37
Das Programm .....	13	<b>19. Tag</b> .....	<b>40</b>
<b>6. Tag</b> .....	<b>15</b>	Heute im Adventskalender .....	40
Heute im Adventskalender .....	15	Lauflicht oder Blinken .....	40
LEDs mit Scratch-Steuerpult umschalten .....	15	Das Programm .....	40
Das Programm .....	15	<b>20. Tag</b> .....	<b>42</b>
<b>7. Tag</b> .....	<b>17</b>	Heute im Adventskalender .....	42
Heute im Adventskalender .....	17	RGB-Farbspiele mit Knetesensor steuern .....	42
Lauflicht mit drei LEDs .....	17	Das Programm .....	43
Das Programm .....	17	<b>21. Tag</b> .....	<b>44</b>
<b>8. Tag</b> .....	<b>19</b>	Heute im Adventskalender .....	44
Heute im Adventskalender .....	19	Farbspiele auf dem Bildschirm .....	44
Schaltdraht .....	19	Das Programm .....	44
LEDs dimmen .....	19	<b>22. Tag</b> .....	<b>46</b>
Das Programm .....	19	Heute im Adventskalender .....	46
<b>9. Tag</b> .....	<b>21</b>	Zwei RGB-LEDs unabhängig steuern .....	46
Heute im Adventskalender .....	21	Das Programm .....	46
Lauflicht mit Taster .....	21	<b>23. Tag</b> .....	<b>48</b>
Das Programm .....	21	Heute im Adventskalender .....	48
<b>10. Tag</b> .....	<b>22</b>	Bunt beleuchtete Weihnachtskrippe .....	48
Heute im Adventskalender .....	22	Das Programm .....	48
LEDs mit zwei Tastern umschalten .....	22	<b>24. Tag</b> .....	<b>50</b>
Das Programm .....	22	Heute im Adventskalender .....	50
<b>11. Tag</b> .....	<b>24</b>	Weihnachtslieder auf dem Raspberry Pi .....	50
Heute im Adventskalender .....	24	Das Programm .....	50
RGB-LEDs .....	24	Übersicht .....	51
RGB-Farbspiele .....	24		
Das Programm .....	25		
<b>12. Tag</b> .....	<b>26</b>		
Heute im Adventskalender .....	26		
Farbmischung mit RGB-LEDs .....	26		
Das Programm .....	26		

## Raspberry-Pi-Adventskalender 2018

Dieser Adventskalender enthält für jeden Tag ein Hardwareexperiment mit dem Raspberry Pi. Nach und nach entsteht dabei eine Weihnachtskrippe, die am 24. Dezember in voller Pracht erstrahlt. Die Experimente werden mit Scratch programmiert. Diese Programmiersprache ist auf dem Raspberry Pi vorinstalliert. Alle Experimente funktionieren mit dem Raspberry Pi 3 und dem Raspberry Pi 3 B+.

Mit einem normalen PC oder gar einem Notebook einfache Elektronik zu steuern, ist - auch wenn es nur ein paar LEDs sind - für Hobbyprogrammierer mit kaum vertretbarem Aufwand verbunden. Dem PC fehlen einfach die dafür notwendigen Schnittstellen. Außerdem ist das Windows-Betriebssystem denkbar ungeeignet dafür, mit Elektronik zu kommunizieren.

Obwohl es auf den ersten Blick gar nicht so aussieht, ist der Raspberry Pi ein vollwertiger Computer. Vieles geht etwas langsamer, als man es von modernen PCs gewohnt ist, dafür ist der Raspberry Pi aber auch viel kleiner und vor allem billiger als ein PC.

## 1. Tag

### Heute im Adventskalender

- 1 Steckbrett
- 1 LED gelb mit Vorwiderstand
- 2 GPIO-Verbindungskabel (kurz)

### Den Raspberry Pi vorbereiten

Um den Raspberry Pi in Betrieb zu nehmen, braucht man:

- USB-Tastatur und Maus
- HDMI-Kabel für Monitor
- Netzkabel
- MicroSD-Karte mit Betriebssystem Raspbian Stretch
- Micro-USB-Handyladegerät als Netzteil (mindestens 1.500 mA)

Das Netzteil muss als Letztes angeschlossen werden, denn damit schaltet sich der Raspberry Pi automatisch ein. Es gibt keinen eigenen Ein-/Ausschalter.

### Betriebssysteminstallation in Kürze

Für alle, die ihren Raspberry Pi noch nicht mit der aktuellen Raspbian-Version betriebsbereit haben, folgt hier die Systeminstallation in elf Schritten:

NOOBS (mindestens Version 2.4.5, für Raspberry Pi 3 B+ mindestens 2.7.0) von [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads) auf den PC herunterladen und Zip-Archiv auf die Festplatte entpacken.

Wurde die SD-Karte bereits benutzt, mit SD-Formatter im PC neu formatieren: [www.sdcard.org/downloads/formatter\\_4](http://www.sdcard.org/downloads/formatter_4). Dabei **Format Size Adjustment** einschalten (die SD-Karte muss mindestens 8 GB groß sein).

Alle Dateien und Unterverzeichnisse von NOOBS auf die SD-Karte kopieren.

SD-Karte aus dem PC nehmen, in den Raspberry Pi stecken und booten. Ganz unten **Deutsch** als Installationssprache wählen. Damit wird automatisch auch die deutsche Tastatur ausgewählt.

Das Häkchen beim vorausgewählten Raspbian-Betriebssystem setzen und oben links auf **Install** klicken. Nach Bestätigung einer Sicherheitsabfrage, dass die Speicherkarte überschrieben wird, startet die Installation, die einige Minuten dauert.

Nach abgeschlossener Installation bootet der Raspberry Pi neu.

Im Menü unter **Einstellungen** das Tool **Raspberry Pi Configuration** starten.

Auf der Registerkarte **Lokalisierung** im Feld **Zeitzone festlegen** die Optionen **Europe** und **Berlin** auswählen. **Sprachumgebung** und **Tastatur** sollten automatisch auf Deutsch gesetzt sein. Falls nicht, deutsche Lokalisierung und Tastatur auswählen.

Möchten Sie auf einem Raspberry Pi 3 B+ WLAN nutzen, muss über den Button **WiFi Land festlegen** die Option **DE - Germany** ausgewählt werden. Das WLAN auf dem Raspberry Pi 3 funktioniert noch unabhängig von dieser Einstellung.

Auf der Registerkarte **Schnittstellen** den Schalter **SSH** auf **Aktiviert** setzen, wenn Sie vom PC über das Netzwerk Daten auf den Raspberry Pi übertragen wollen.

Auf **OK** klicken und Raspberry Pi über den Menüpunkt **Shutdown** neu booten. Eine eventuell auftauchende Warnung wegen eines unsicheren Passworts kann ignoriert werden.

### Num-Lock-Taste

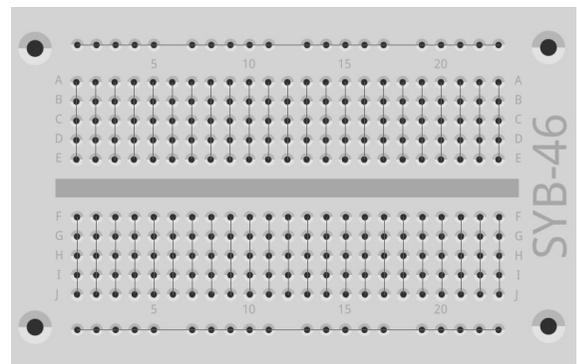
Wie bei fast allen Linux-Systemen ist auch beim Start von Raspbian der Ziffernblock standardmäßig ausgeschaltet. Drücken Sie die Num-Lock-Taste auf der Tastatur, um ihn einzuschalten.

## Die wichtigsten Bauteile kurz erklärt

### Steckbrett

Für den schnellen Aufbau elektronischer Schaltungen, ohne dass man löten muss, enthält der Adventskalender ein Steckbrett. Hier können elektronische Bauteile direkt in ein Lochraster gesteckt werden.

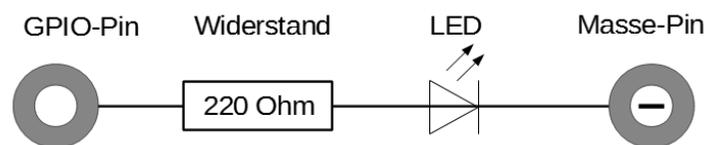
Bei diesem Steckbrett sind alle äußeren Längsreihen über Kontakte (X und Y) miteinander verbunden. Diese Kontaktreihen werden oft als Plus- und Minuspol zur Stromversorgung der Schaltungen genutzt. In den anderen Kontaktreihen sind jeweils fünf Kontakte (A bis E und F bis J) quer miteinander verbunden,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



Die Verbindungen auf dem Steckbrett.

### LEDs

LEDs (zu Deutsch: Leuchtdioden) leuchten, wenn Strom in Durchflussrichtung durch sie fließt. LEDs werden in Schaltungen mit einem pfeilförmigen Dreieckssymbol dargestellt, das die Flussrichtung vom Pluspol zum Minuspol oder zur Masseleitung angibt. Eine LED lässt in Durchflussrichtung nahezu beliebig viel Strom durch, sie hat nur einen sehr geringen Widerstand. Um den Durchflussstrom zu begrenzen und damit ein Durchbrennen der LED zu verhindern, wird üblicherweise zwischen dem verwendeten GPIO-Pin und der Anode der LED oder zwischen der Kathode und dem Massepin ein 220-Ohm-Vorwiderstand eingebaut. Dieser Vorwiderstand schützt auch den GPIO-Ausgang des Raspberry Pi vor zu hohen Stromstärken. Die LEDs im Adventskalender haben den Vorwiderstand bereits eingebaut und können daher direkt an die GPIO-Pins angeschlossen werden.



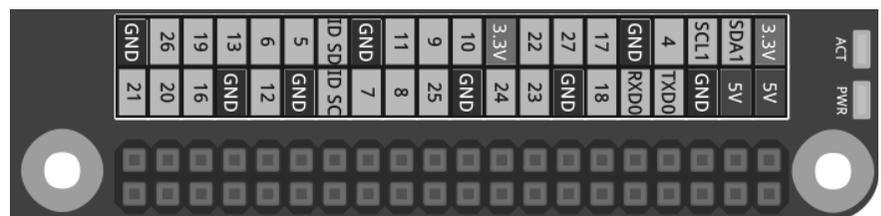
Schaltplan einer LED mit Vorwiderstand.

### LED in welcher Richtung anschließen?

Die beiden Anschlussdrähte einer LED sind unterschiedlich lang. Der längere ist der Pluspol, die Anode, der kürzere der Minuspol, die Kathode. Einfach zu merken: Das Pluszeichen hat einen Strich mehr als das Minuszeichen und macht damit den Draht optisch etwas länger. Außerdem sind die meisten LEDs auf der Minusseite abgeflacht, vergleichbar mit einem Minuszeichen. Auch leicht zu merken: Kathode = kurz = Kante.

### GPIO-Verbindungskabel

Die farbigen Verbindungskabel haben alle auf einer Seite einen Stecker, auf der anderen Seite eine Steckbuchse, die auf einen GPIO-Pin des Raspberry Pi passt. Die LEDs auf der Weihnachtskrippe werden ebenfalls direkt in diese Steckbuchsen gesteckt. Die Stecker werden in das Steckbrett gesteckt. Die programmierbaren GPIO-Pins auf dem Raspberry Pi haben Nummern, die Massepins sind in der Abbildung mit GND gekennzeichnet.



Belegung der GPIO-Pins.

### Vorsichtsmaßnahmen

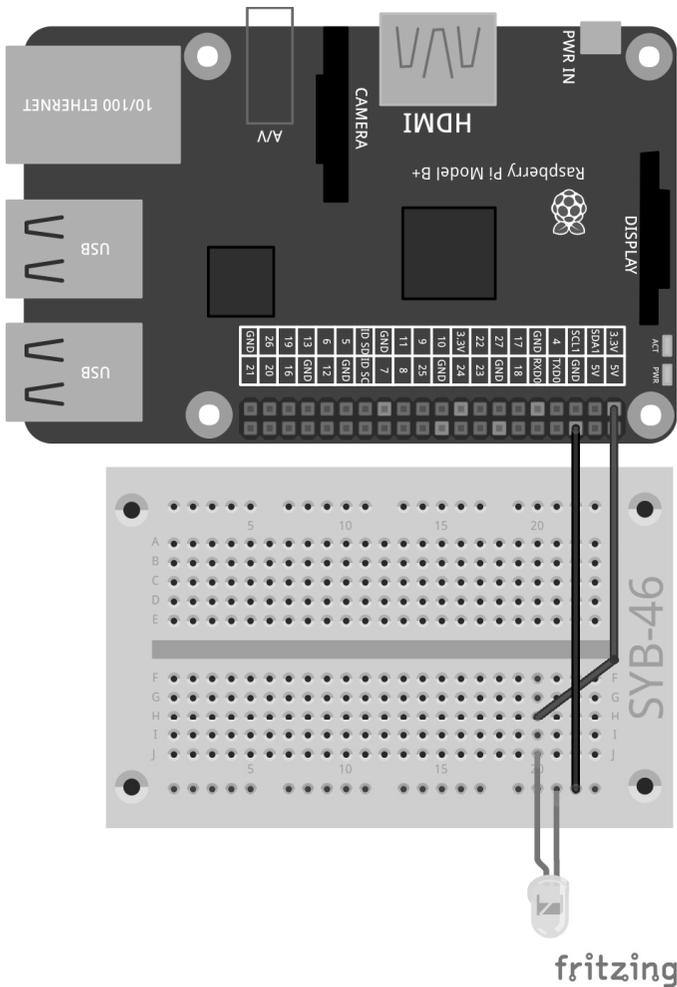
Auf keinen Fall sollte man irgendwelche GPIO-Pins miteinander verbinden und abwarten ab, was passiert.

Nicht alle GPIO-Pins lassen sich frei programmieren. Ein paar sind für die Stromversorgung und andere Zwecke fest eingerichtet.

Einige GPIO-Pins sind direkt mit Anschlüssen des Prozessors verbunden, ein Kurzschluss kann den Raspberry Pi komplett zerstören. Verbindet man über einen Schalter oder eine LED zwei Pins miteinander, muss immer ein Schutzwiderstand dazwischengeschaltet werden. Eine Ausnahme bilden die LEDs mit eingebautem Vorwiderstand.

Für Logiksignale muss immer Pin 1 verwendet werden, der +3,3 V liefert und bis 50 mA belastet werden kann. Pin 6 ist die Masseleitung für Logiksignale.

Pin 2 und 4 liefern +5 V zur Stromversorgung externer Hardware. Hier kann so viel Strom entnommen werden, wie das USB-Netzteil des Raspberry Pi liefert. Diese Pins dürfen aber nicht mit einem GPIO-Eingang verbunden werden.



fritzing

Die erste LED leuchtet am Raspberry Pi.

### Eine LED leuchtet

Für das erste Experiment wird kein Programm benötigt. Der Raspberry Pi dient hier nur als Stromversorgung für die RGB-LED. Das Experiment zeigt, wie RGB-LEDs angeschlossen werden. Achten Sie darauf, dass die RGB-LED richtig herum eingebaut ist. Die flache Seite ist in der Abbildung rechts.

Die meisten Schaltungsaufbauten verwenden die Kontaktleiste an der einen Längsseite des Steckbretts als Massekontakt. Hier werden die Kathoden aller LEDs eingesteckt und mithilfe eines Kabels mit einem GND-Pin auf dem Raspberry Pi verbunden.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED gelb mit Vorwiderstand; 2 GPIO-Verbindungskabel kurz (Raspberry Pi – Steckbrett)

### Programme zum Download

Die im Adventskalender verwendeten Programme gibt es hier zum Download:  
<http://bit.ly/c-adventskalender-raspberry-pi-18>



Öffnen Sie die Webseite direkt mit dem vorinstallierten Browser auf dem Raspberry Pi und laden Sie die Zip-Datei in das Home-Verzeichnis `/home/pi` herunter.



Starten Sie den Dateimanager auf dem Raspberry Pi. Dieser zeigt beim Start automatisch das Home-Verzeichnis an. Klicken Sie mit der rechten Maustaste auf die heruntergeladene Zip-Datei und wählen Sie im Kontextmenü **Hier entpacken**.

Das Downloadarchiv zum Adventskalender enthält dieses Handbuch als PDF in Farbe, damit Sie auf den Schaltplänen die einzelnen Leitungen besser erkennen können.

## 2. Tag

### Heute im Adventskalender

• 3 GPIO-Verbindungskabel (lang)

Der Adventskalender enthält GPIO-Verbindungskabel in zwei verschiedenen Längen. Die kurzen Kabel dienen der Verbindung zwischen Raspberry Pi und Steckbrett, die langen Kabel verbinden die LEDs auf der Weihnachtskrippe mit dem Steckbrett.

### Die erste LED leuchtet auf der Weihnachtskrippe

Die LED wird in die gekennzeichneten Löcher auf dem Schweiß des Weihnachtssterns über der Weihnachtskrippe gesteckt. Schneiden Sie diese dazu entlang der vorgezeichneten Linien auf der Rückseitenpappe des Adventskalenders aus.

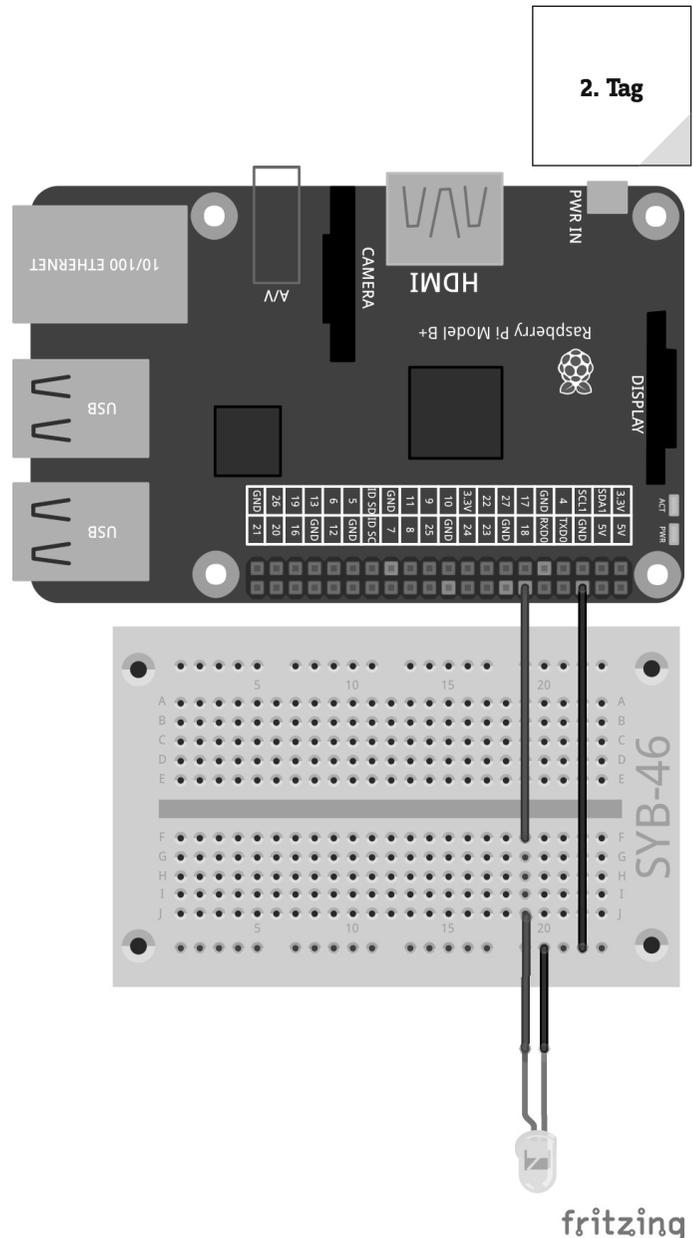
Die Buchsen der GPIO-Verbindungskabel kommen von der Rückseite auf die Anschlussdrähte der LED. Stecken Sie die LED so weit wie möglich in die Kabelbuchsen. Sollte sie trotzdem herausrutschen, knicken Sie die Anschlussdrähte der LED wenige Millimeter vor den Enden in einem flachen Winkel ab.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED gelb mit Vorwiderstand; 2 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 2 GPIO-Verbindungskabel lang (Steckbrett - LEDs)

Stecken Sie die LED oben an den Stern auf der Weihnachtskrippe.

Diesmal leuchtet die LED nicht permanent, sondern wird von einem Programm in der Programmiersprache Scratch für eine halbe Sekunde eingeschaltet.

Scratch ist auf dem Raspberry Pi im Menü unter **Entwicklung** vorinstalliert und gilt als eine der am leichtesten zu erlernenden Programmiersprachen.



Die erste LED leuchtet auf der Weihnachtskrippe.

### Das neue Scratch 2

Seit der ersten Raspbian-Version ist die Programmiersprache Scratch in der Version 1.x vorinstalliert. Für PCs gibt es seit einigen Jahren die neue Version Scratch 2 mit deutlich mehr Möglichkeiten. Unter anderem lassen sich damit eigene Funktionsblöcke erstellen.

Scratch 2 läuft auf dem PC online im Browser. Dafür ist allerdings mehr Rechenleistung erforderlich, als ein Raspberry Pi zurzeit bieten kann. Seit der Version NOOBS 2.4.0 ist im Raspbian-Betriebssystem eine Version von Scratch 2 vorinstalliert, die offline ohne Browser läuft und so mit der Leistung eines Raspberry Pi 3 problemlos auskommt. Mit Scratch 2 ist die Hardwaresteuerung über die GPIO-Schnittstelle wesentlich einfacher geworden. Allerdings werden einige wichtige Funktionen zur GPIO-Steuerung noch nicht unterstützt. Deshalb verwenden wir für alle Projekte in diesem Adventskalender weiterhin das bewährte Scratch 1.4.

### Grundeinstellungen für Scratch

Klicken Sie oben links neben dem Scratch-Logo auf die Weltkugel und wählen Sie **Deutsch**. Die ausgewählte Sprache bleibt gespeichert, muss also nicht jedes Mal neu ausgewählt werden.

Scratch 1.4 bietet Unterstützung für verschiedene Hardwarekomponenten am GPIO-Port, die bei jedem Programm über den Menüpunkt **Bearbeiten/Start GPIO server** einmal aktiviert werden muss.



Scratch 1.4 auf Deutsch umschalten.

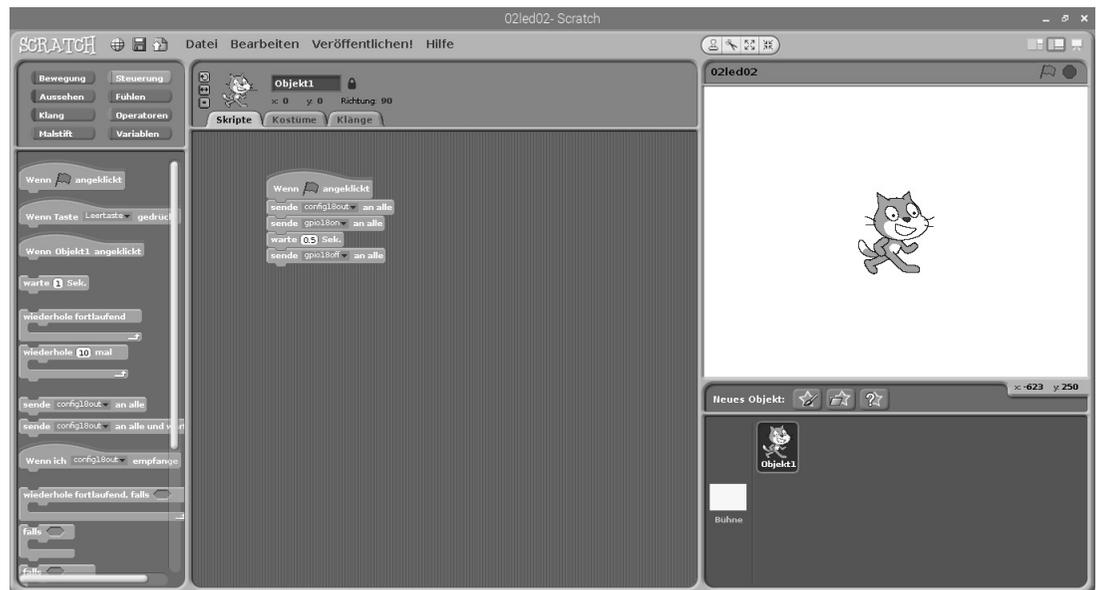


Den Scratch-1.4-GPIO-Server starten.

Dass die GPIO-Unterstützung aktiv ist, erkennen Sie daran, dass sich dieser Menüpunkt in **Stop GPIO server** ändert. Überprüfen Sie das bei jedem neuen Scratch-Programm.

### Das Programm

In Scratch braucht man beim Programmieren keinen Programmcode zu tippen. Die Blöcke werden einfach per Drag-and-drop aneinandergehängt. Die Blockpalette im linken Teil des Scratch-Fensters enthält, nach Themen geordnet, die verfügbaren Blöcke.



Das Scratch-Programm 02Led02 lässt die LED eine halbe Sekunde lang leuchten.

Sie können das Programm auf dem Bildschirm selbst zusammenbauen oder das Programm 02Led02 aus dem Download zum Adventskalender verwenden. Wählen Sie dazu Menü **Datei/Öffnen** und klicken Sie im nächsten Dialogfeld auf die Schaltfläche **pi**, um das persönliche Home-Verzeichnis auszuwählen, in dem die heruntergeladenen Programme liegen.

Klicken Sie in Scratch oben links auf das gelbe Symbol **Steuerung**, damit in der Blockpalette links die Blöcke zur Steuerung angezeigt werden. Für dieses erste Programm brauchen wir nur die gelben Blöcke.

Ziehen Sie die Blöcke, die Sie brauchen, einfach aus der Blockpalette in das Skriptfenster in der Mitte von Scratch.



Der Block **Wenn (grünes Fähnchen) angeklickt** dient dazu, ein Programm zu starten. Die folgenden Skriptelemente werden ausgeführt, wenn man auf das grüne Fähnchen rechts oben in Scratch klickt. Der Block ist oben rund, passt also unter keinen anderen Block. Er muss immer als Erstes gesetzt werden.

Die GPIO-Befehle werden über den Scratch-Block **sende ... an alle** ausgegeben. Im Textfeld werden die jeweilige Pinbezeichnung und entsprechende Schlüsselwörter eingetragen. Klicken Sie dazu in das Textfeld im Block, wählen Sie **Neu/edit...** und geben Sie den Text ein.

Am Anfang wird der GPIO-Pin 18 mit **config18out** als Ausgang definiert. Jeder GPIO-Pin kann entweder Ausgang oder Eingang sein.

Im nächsten Schritt wird über einen weiteren Scratch-Block **sende ... an alle** mit dem Text **gpio18on** die an GPIO-Pin 18 angeschlossene LED eingeschaltet.

Danach wartet das Programm eine halbe Sekunde. Dazu bietet Scratch einen eigenen Block **warte ... Sek.** an. Scratch verwendet wie viele amerikanische Programme den Punkt als Dezimaltrennzeichen, nicht das in Deutschland übliche Komma. Eine halbe Sekunde wird also als 0.5 eingetragen und nicht als 0,5.

Zuletzt wird über einen weiteren Scratch-Block **sende ... an alle** mit dem Text **gpio18off** die an GPIO-Pin 18 angeschlossene LED wieder ausgeschaltet.

Das Programm startet, wenn man oben rechts im Scratch-Fenster auf das grüne Fähnchen klickt.

### 3. Tag

#### Heute im Adventskalender

- 1 LED rot mit Vorwiderstand
- 1 GPIO-Verbindungskabel (kurz)

#### Zwei LEDs blinken abwechselnd

Das Experiment des 3. Tags lässt zwei LEDs abwechselnd rot und gelb leuchten. Gesteuert wird das Ganze über eine Endlosschleife in Scratch.

**Bauteile:** 1 Steckbrett; 1 LED rot mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 3 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett)

Die LEDs stecken bei diesem Experiment direkt auf dem Steckbrett mit den Kathoden in der Masseleiste.

#### Das Programm

Am Anfang werden die beiden GPIO-Pins 18 und 24 mit **config18out** und **config24out** als Ausgänge definiert.



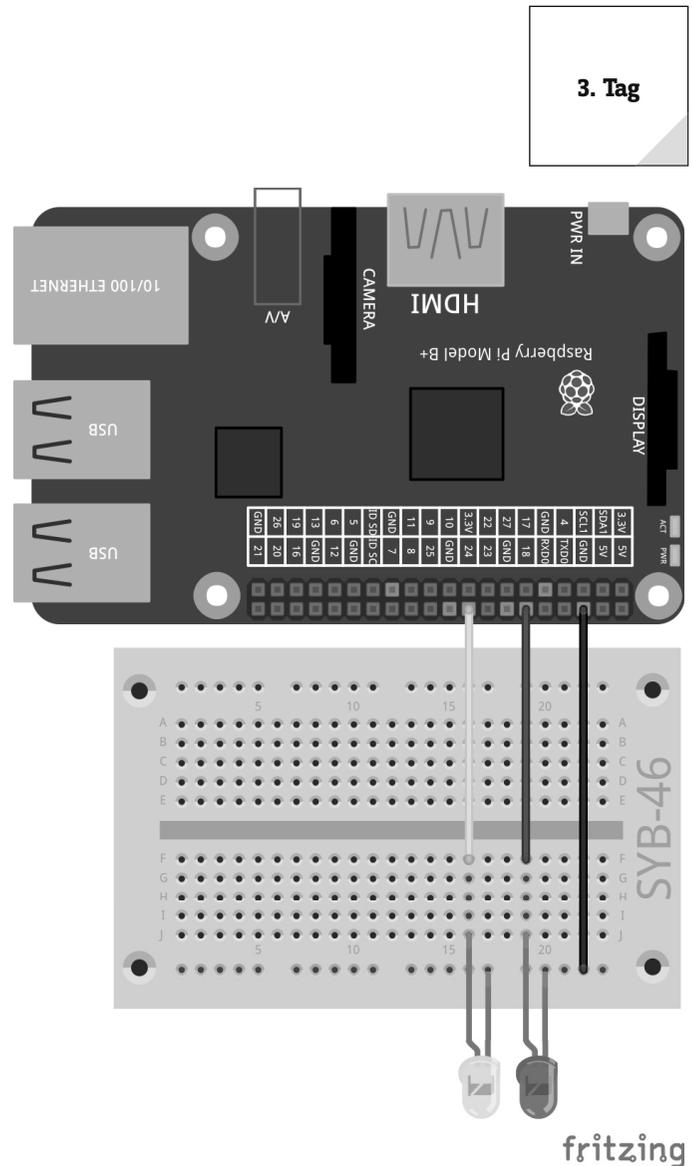
Eine **wiederhole fortlaufend**-Schleife sorgt dafür, dass die beiden LEDs endlos blinken, und zwar so lange, bis der Benutzer auf das rote Stoppsymbol oben rechts in Scratch klickt.



Nachdem die rote LED an Pin 18 eingeschaltet und die gelbe an Pin 24 ausgeschaltet ist, wartet das Programm eine halbe Sekunde. Danach werden auf die gleiche Weise die gelbe LED an Pin 24 eingeschaltet und die rote an Pin 18 ausgeschaltet. Nach einer weiteren halben Sekunde beginnt der Zyklus von vorn.



Das Programm 03led03 steuert die beiden LEDs.



Zwei LEDs blinken am Raspberry Pi.

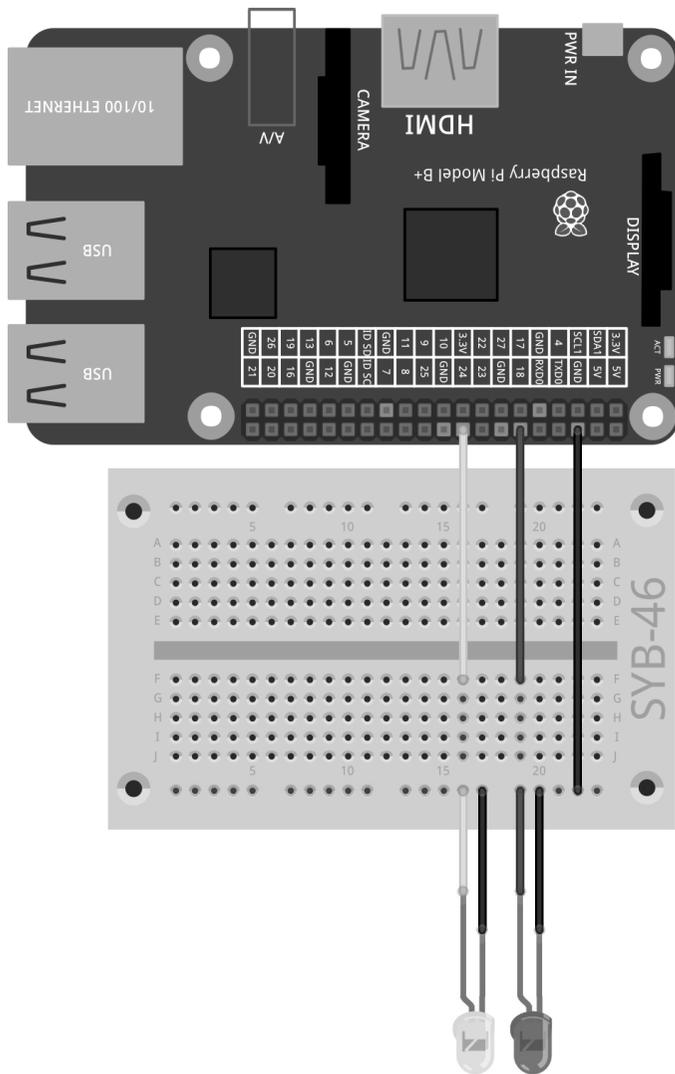


## 4. Tag

### Heute im Adventskalender

- 3 GPIO-Verbindungskabel (lang)

### Ein Stern schaltet zwei LEDs um



Das Experiment des 4. Tags schaltet zwei LEDs ein oder aus. Zur Steuerung wird ein Objekt in Scratch verwendet, das auf dem Bildschirm bewegt werden kann.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED gelb mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 3 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 4 GPIO-Verbindungskabel lang (Steckbrett - LEDs)

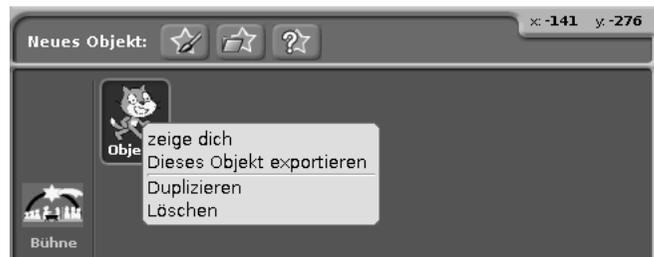
### Das Programm

Das Programm verwendet das Krippenbild von der Rückseite des Adventskalenders als Bildschirmhintergrund.

Klicken Sie im Objektfenster (unten rechts) auf die Bühne und gehen Sie dann im Skriptfenster auf die Registerkarte **Hintergründe**.

Klicken Sie auf **Importieren** und importieren Sie das Bild *Krippe\_blaue.png* aus dem Download. Entfernen Sie anschließend den vorgegebenen Hintergrund **Hintergrund1** mit einem Klick auf das X-Symbol.

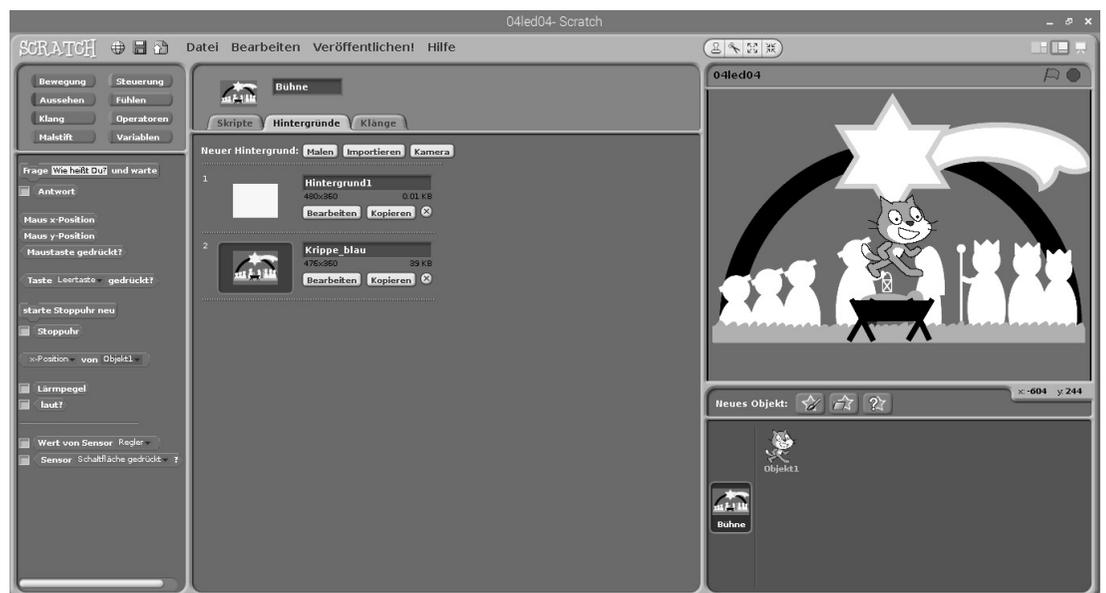
Löschen Sie als Nächstes die Katze, die in diesem Programm nicht gebraucht wird. Klicken Sie dazu mit der rechten Maustaste auf die Katze im Objektfenster und wählen Sie im Kontextmenü **Löschen**.



Scratch-Katze löschen.

Im Programm soll ein **Stern**-Objekt die LEDs umschalten. Scratch enthält ein einfaches Malprogramm, mit dem Sie dieses Objekt direkt in Scratch malen können. Klicken Sie dazu auf das Symbol **Neues Objekt malen** im Objektfenster.

Zwei LEDs werden über ein Scratch-Programm umgeschaltet.

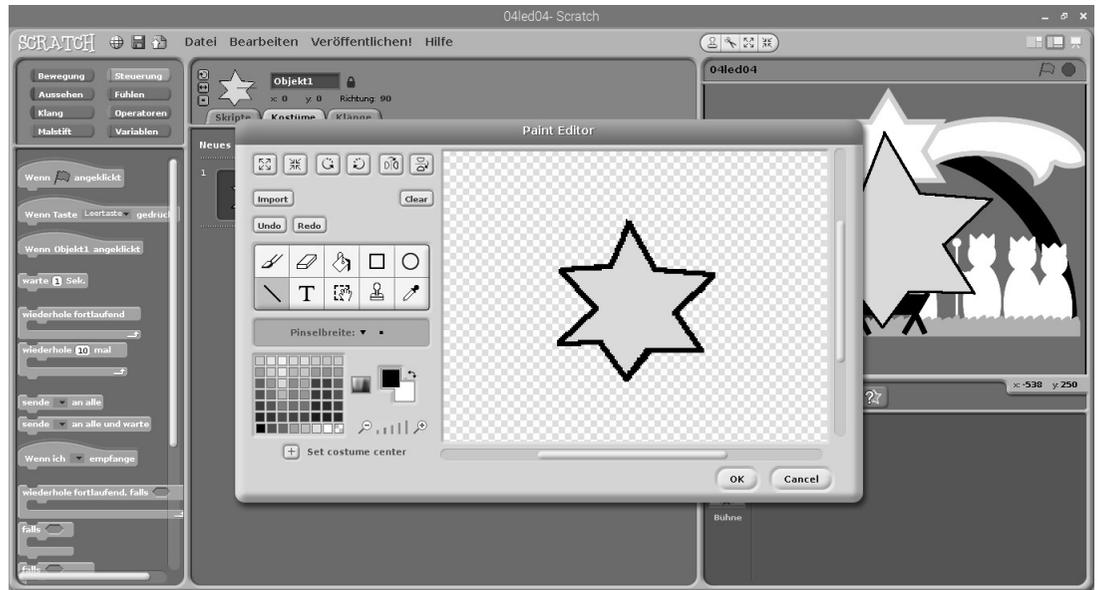


Das neue Hintergrundbild für das Programm.

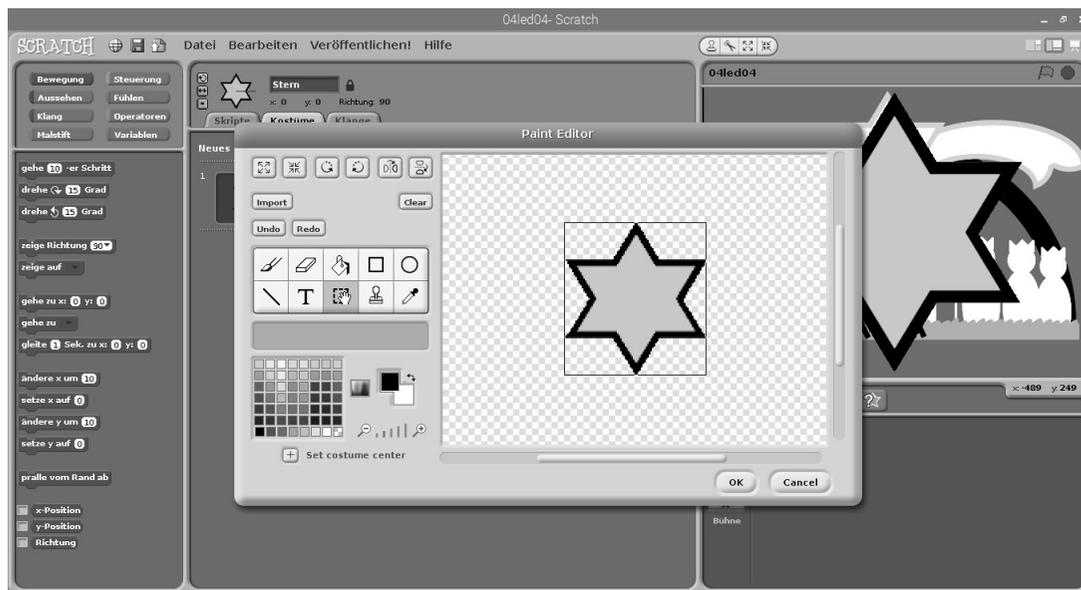
Alternativ können Sie über das Symbol **Neues Objekt aus Datei laden** die Datei stern.png aus dem Download als neues Objekt importieren.

Der Stern ist standardmäßig viel zu groß für die Szene. Schalten Sie im Skriptfenster auf die Registerkarte **Kostüme** um und klicken Sie auf **Bearbeiten**, um das Malprogramm zu öffnen.

Schrumpfen Sie den Stern über das Symbol **Schrumpfen** links oben auf die passende Größe.

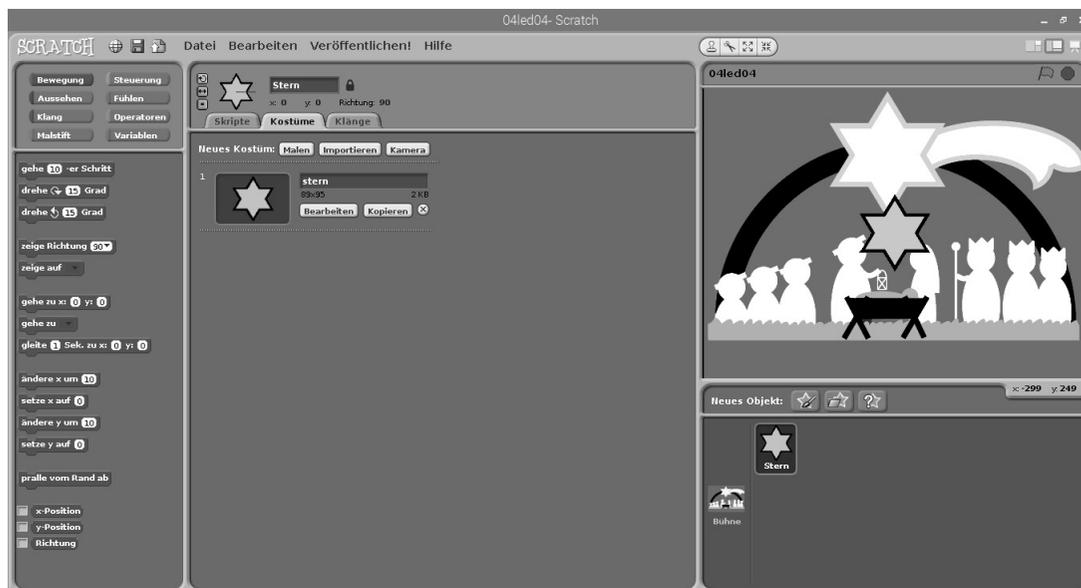


Das Malprogramm in Scratch.



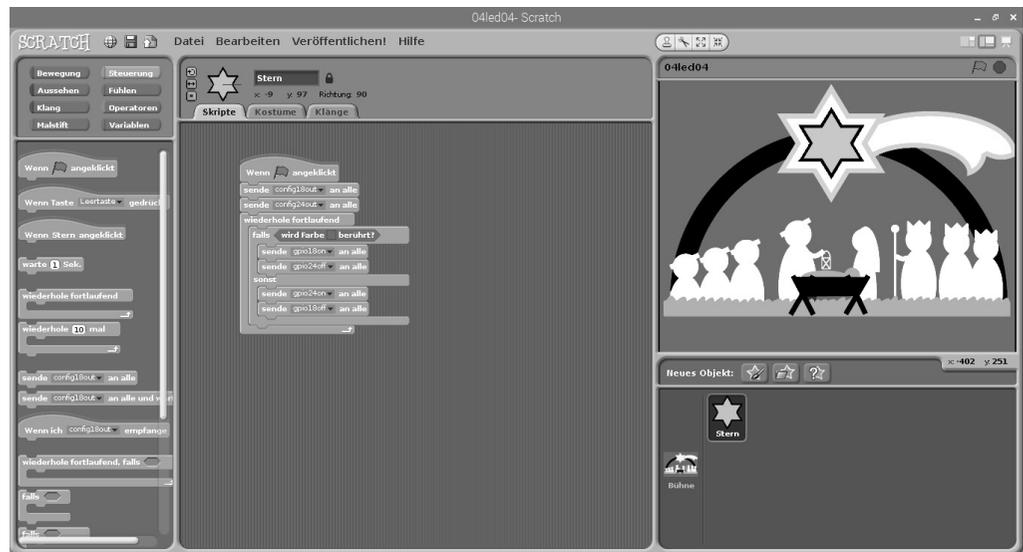
Der importierte Stern im Malprogramm in Scratch.

Benennen Sie das neue Objekt im Namensfeld über dem Skriptfenster in **Stern** um.



Der verkleinerte, umbenannte Stern.

Schieben Sie den Stern anschließend auf den großen Stern des Hintergrundbilds.



Das Programm 04Led04 schaltet zwei LEDs mit einem Stern um.

Das Programm 04Led04 startet, nachdem die GPIO-Pins initialisiert wurden, eine Endlosschleife, die ständig prüft, ob der Stern eine blaue Fläche berührt. Solange das nicht der Fall ist, befindet er sich auf dem Stern im Hintergrundbild - die einzige weiße Fläche, die groß genug ist.

Ein **Falls ... sonst ...**-Block schaltet je nach Position des Sterns eine LED ein und die andere aus.

Ziehen Sie einen Block **wird Farbe ... berührt** von der Blockpalette **Fühlen** in das Abfragefeld des **Falls ... sonst ...**-Blocks.

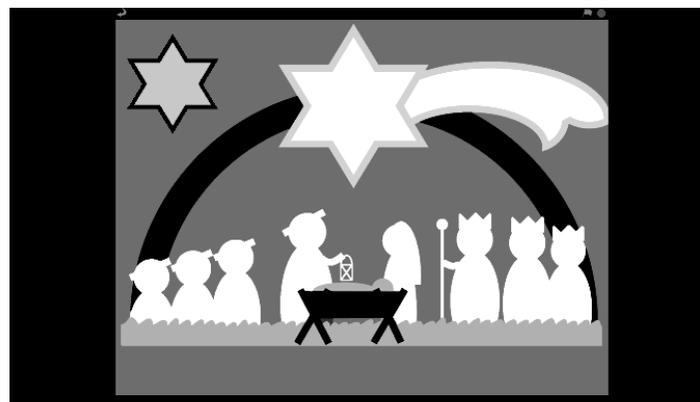
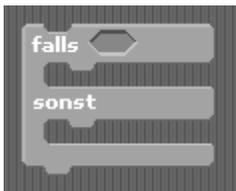
Tippen Sie auf das farbige Quadrat innerhalb dieses Blocks, erscheint eine Pipette. Klicken Sie mit ihr auf eine blaue Fläche im Hintergrundbild. Damit wird diese Farbe in der Abfrage ausgewählt.

Fügen Sie jetzt noch die **Sende ... an alle**-Blöcke zum Umschalten der LEDs in die beiden Klammern des **Falls ... sonst ...**-Blocks ein, wie in der Abbildung gezeigt.

Starten Sie das Programm mit einem Klick auf das grüne Fähnchen, leuchtet die LED an Pin 24.

Ziehen Sie den Stern in Richtung einer der Bildschirmecken, sodass er den blauen Hintergrund berührt. Das Programm schaltet automatisch auf die andere LED um.

Das Präsentationssymbol oben rechts in der Ecke des Scratch-Fensters schaltet auf den Präsentationsmodus um, in dem die Scratch-Bühne im Vollbildmodus erscheint. Alle Bedienelemente verschwinden. Ein Klick auf den Pfeil links oben schaltet wieder zurück.



Das Programm 04Led04 im Präsentationsmodus.

## 5. Tag

### Heute im Adventskalender

- 1 Taster
- 1 GPIO-Verbindungskabel (kurz)

### LEDs mit Taster umschalten

Das Experiment des 5. Tags schaltet die LEDs nicht automatisch um, sondern erst, wenn der Benutzer einen Taster betätigt.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED gelb mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 Taster; 4 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 1 GPIO-Verbindungskabel lang (bis zu diesem Tag sind noch nicht genügend kurze Kabel verfügbar)

### Das Programm

Das Programm funktioniert ähnlich wie das des 3. Tags. Wieder werden in einer Endlosschleife die beiden LEDs abwechselnd ein- und ausgeschaltet. Im Unterschied zum vorherigen Programm erfolgt die Umschaltung nicht nach einer bestimmten Zeit, sondern erst, wenn der Benutzer den Taster gedrückt hat. Jeder Tastendruck wechselt die LEDs.



Das Programm 05Led05 steuert die beiden LEDs über einen Taster.

Am Anfang wird zusätzlich zu den beiden Ausgängen GPIO-Pin 21 als Eingang definiert. Eingänge am Raspberry Pi verarbeiten digitale Logiksignale. Ist ein Eingang mit +3,3 V verbunden, bekommt er das Signal logisch **High**, was Scratch als **1** auswertet; ist der Eingang mit GND verbunden, bekommt er das Signal logisch **Low**, was Scratch als **0** auswertet.

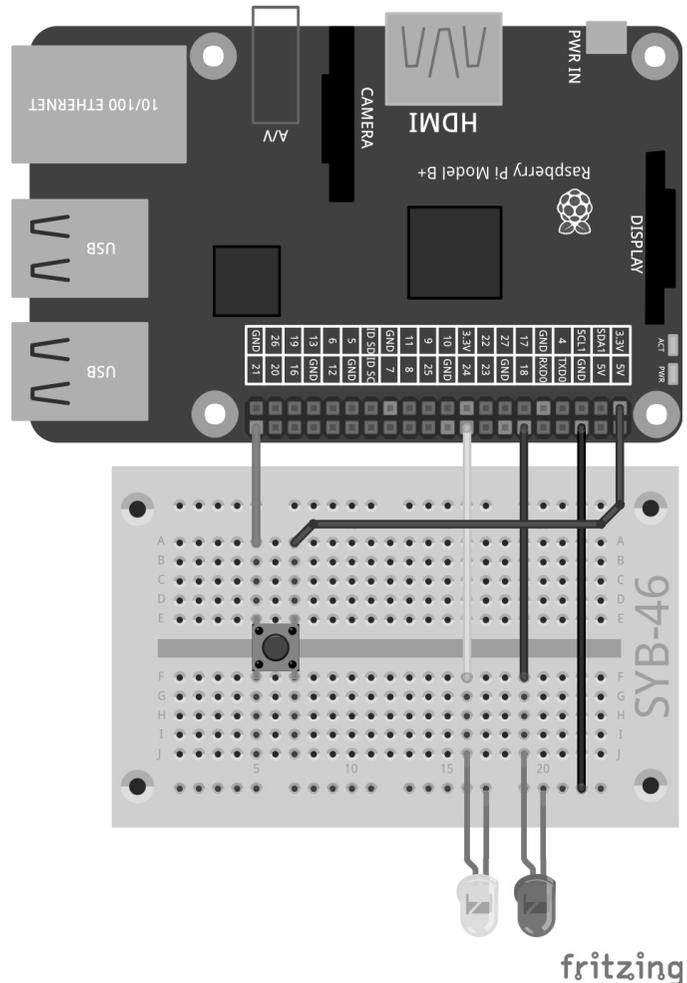
### Achtung

Verwenden Sie nie die +5-V-Pins des Raspberry Pi für Logiksignale in Schaltungen. 5 V würden die GPIO-Eingänge überlasten und den Raspberry Pi beschädigen.

Ein Druck auf den Taster verbindet in unserer Schaltung GPIO-Pin 21 mit +3,3 V. Lässt man den Taster wieder los, bekommt der Eingang einen undefinierten Zustand, was in der Digitalelektronik nicht passieren darf. Für solche Fälle verfügen alle GPIO-Pins über sogenannte Pull-down-Widerstände, die einen Eingang, an dem kein Signal anliegt, automatisch auf **Low** herunterziehen.

Definieren Sie den GPIO-Pin für den Taster mit **config21inputpulldown**, um den eingebauten Pull-down-Widerstand am Eingang zu aktivieren.

Klicken Sie jetzt einmal auf das grüne Fähnchen oben rechts, um das noch unfertige Programm zu starten. Damit werden die GPIO-Pins definiert. Nur so können sie in den Listen ausgewählt werden. Das Programm endet automatisch.



LEDs mit Taster umschalten.



Die GPIO-Pins werden initialisiert.

Danach beginnt eine Endlosschleife, die in jedem Durchlauf als Erstes die LED an GPIO-Pin 18 ein- und die an Pin 24 ausschaltet.



Wenn eine LED leuchtet, wartet das Programm nicht mehr eine bestimmte Zeitlang, sondern mit einem **warte bis ...**-Block so lange, bis ein bestimmtes Ereignis eintritt, in diesem Fall, bis GPIO-Pin 21 den Wert **1** annimmt, der Taster also gedrückt wurde.

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



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

In unserem Fall soll der Wert des GPIO-Pins 21 dem Wert **1** entsprechen. Die Ziffer 1 steht für **High**. Schreiben Sie also eine 1 in das rechte der beiden Textfelder in dem grünen Block.

Zur Abfrage von GPIO-Eingängen wird der Block **Wert von Sensor ...** aus der blauen Blockpalette **Fühlen** verwendet. Wählen Sie im Listenfeld des blauen Blocks den Sensor **gpio21** aus. Neben einigen vordefinierten Sensoren werden alle GPIO-Pins zur Auswahl angeboten, die als Eingang definiert sind. Deshalb musste das Programm vorher einmal kurz gestartet werden.



Ziehen Sie dann den blauen Block **Wert von Sensor ...** in das linke Feld des grünen Blocks innerhalb des **warte bis ...**-Blocks.

Danach wartet das Programm mit einem **warte ... Sek.**-Block 0,2 Sekunden lang. Damit wird verhindert, dass der Taster sofort wieder als gedrückt gilt, wenn das Programm weiterläuft. Der Benutzer hat so lange Zeit, den Taster wieder loszulassen.

Erst anschließend werden die LEDs umgeschaltet, und das Programm wartet erneut darauf, dass der Benutzer den Taster drückt.

**Blöcke duplizieren**

Beim Bau eines Scratch-Programms brauchen Sie ähnliche Blockkombinationen nicht jedes Mal neu anzulegen. Klicken Sie mit der rechten Maustaste auf den ersten Block, der dupliziert werden soll. Wählen Sie dann im Menü **Duplizieren**. Alle darunterhängenden Blöcke werden automatisch mit dupliziert. Die duplizierten Blöcke können an passender Stelle im Programm eingefügt werden.

## 6. Tag

### Heute im Adventskalender

• 4 GPIO-Verbindungskabel (lang)

### LEDs mit Scratch-Steuerpult umschalten

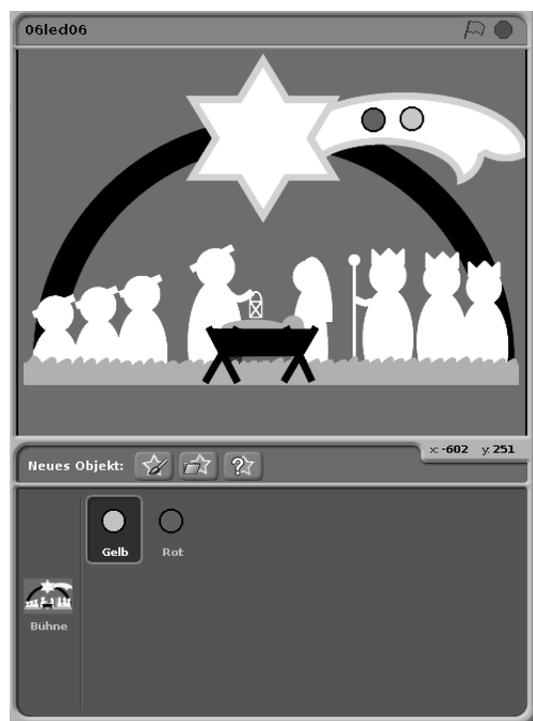
Im Programm des 6. Tags werden durch Klicks in Scratch zwei LEDs ein- und ausgeschaltet. Dabei zeigt das Scratch-Steuerpult den Zustand der LEDs an.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED gelb mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 3 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 4 GPIO-Verbindungskabel lang (Steckbrett - LEDs)

### Das Programm

Das Programm 06Led06 verwendet das gleiche Hintergrundbild wie die Programme der letzten Tage.

Für die beiden LEDs werden auf dem Schweif des Sterns zwei kreisförmige Objekte mit den Namen **Gelb** und **Rot** angelegt. Löschen Sie die Katze und malen Sie eines der Objekte. Sie können auch die Datei `kreis_gelb.png` aus den Downloads importieren. Um das zweite Objekt zu erstellen, duplizieren Sie das erste später mitsamt allen Einstellungen und Skriptblöcken und passen es nur noch geringfügig an.

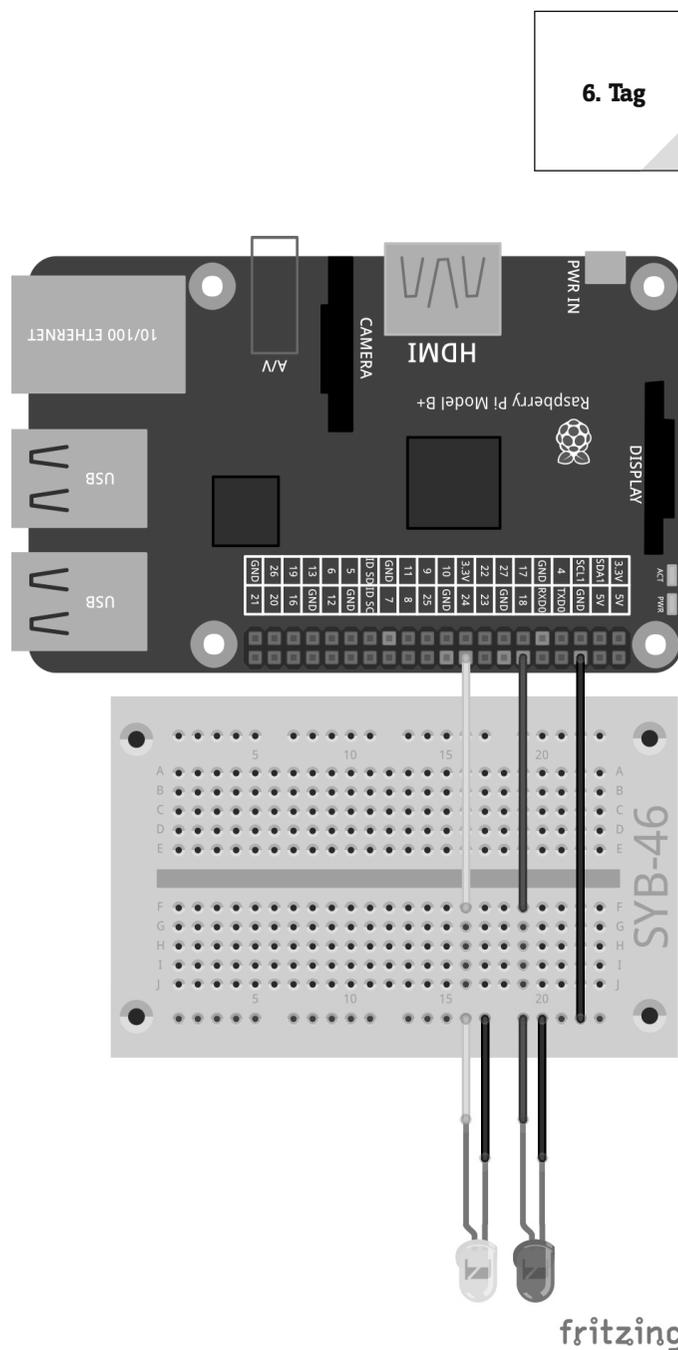


Die beiden Objekte für die LEDs.

Jedes Objekt in Scratch kann über sogenannte Kostüme sein Aussehen verändern. Wählen Sie das LED-Objekt aus und gehen Sie im Skriptfenster auf die Registerkarte **Kostüme**.

Kopieren Sie das vorhandene Kostüm mit einem Klick auf **Kopieren**. Benennen Sie die beiden Kostüme um: **ein** für die eingeschaltete und **aus** für die ausgeschaltete LED.

Bearbeiten Sie das Kostüm **aus** mit dem Malprogramm. Füllen Sie die farbige Fläche mit Grau, um die ausgeschaltete LED darzustellen.

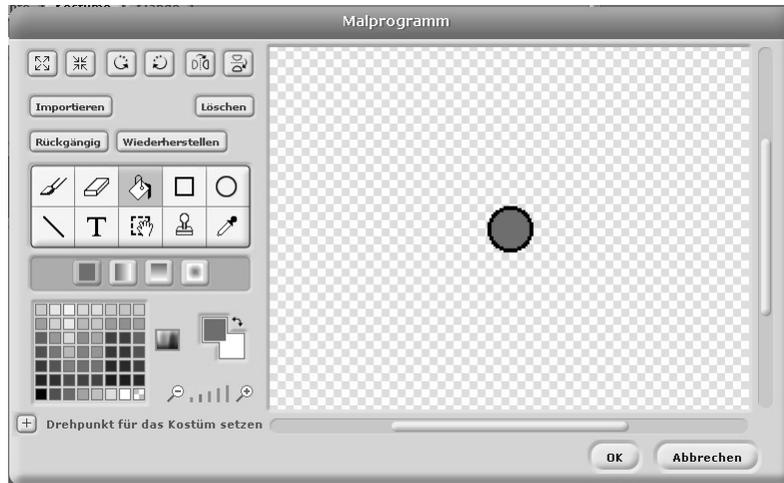


Drei LEDs auf der Weihnachtskrippe.

fritzing



Die beiden Kostüme des Objekts Gelb.



Kostüm für eine ausgeschaltete LED im Malprogramm.



Das Programm besteht aus mehreren Teilen – für die Bühne und die Objekte. Wählen Sie zuerst im Objektfenster die Bühne. Der Skriptblock der Bühne initialisiert beim Klick auf das grüne Fähnchen die GPIO-Pins.

Das Objekt **Gelb** bekommt eigene Skriptblöcke. Wenn das grüne Fähnchen angeklickt wird, soll das Objekt ausgeschaltet, also grau erscheinen. Hängen Sie dazu an den Block **Wenn grünes (grünes Fähnchen) angeklickt** einen Block **Ziehe Kostüm ... an** von der Blockpalette **Aussehen**. Wählen Sie im Auswahlfeld dieses Blocks das Kostüm **aus**.



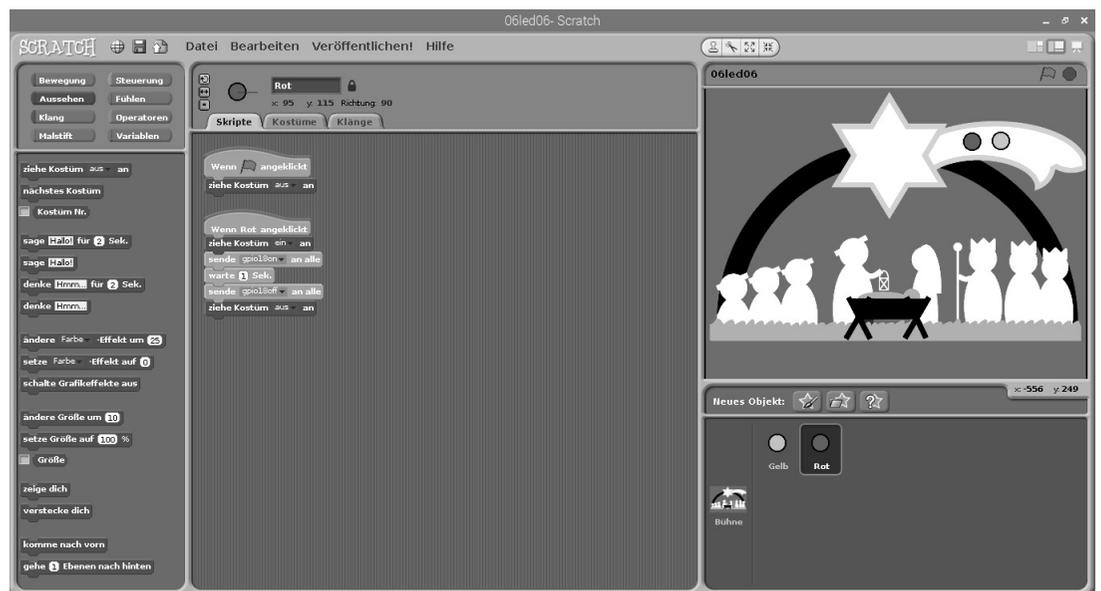
Wird das Objekt selbst angeklickt, soll es auf dem Bildschirm für eine Sekunde farbig aufleuchten. Gleichzeitig soll auch die entsprechende LED leuchten.

Verwenden Sie dazu den Block **Wenn Gelb angeklickt** von der Registerkarte **Steuerung**.

In diesem Fall wird zuerst das Kostüm auf **ein** gewechselt, dann wird die LED an GPIO-Pin 24 eingeschaltet. Nach einer Wartezeit von einer Sekunde wird diese LED wieder ausgeschaltet und das Kostüm zurück auf **aus** gewechselt.



Sind alle Skriptblöcke fertiggestellt, duplizieren Sie mit einem Rechtsklick im Objektfenster das Objekt **Gelb**. Benennen Sie das neue Objekt in **Rot** um.



Das duplizierte Objekt für die rote LED.

Bearbeiten Sie das Kostüm **ein** im Malprogramm. Füllen Sie hier den Kreis mit roter Farbe. Ändern Sie anschließend im Skriptblock noch den verwendeten GPIO-Pin auf 18.

Starten Sie das Programm mit einem Klick auf das grüne Fähnchen. Beide LEDs sind aus, die Objekte erscheinen in Grau. Klicken Sie auf eines der Objekte, erscheint es in Farbe, und die zugehörige LED leuchtet für eine Sekunde.

## 7. Tag

### Heute im Adventskalender

• 1 LED grün mit Vorwiderstand



### Lauflicht mit drei LEDs

Das Programm des 7. Tags lässt drei LEDs abwechselnd leuchten. Der Lauflichteffekt entsteht dadurch, dass eine LED für kurze Zeit leuchtet, und sobald sie ausgeschaltet ist, sofort die LED direkt daneben eingeschaltet wird. Nachdem die letzte LED in der Reihe ausgeschaltet wurde, wird die erste wieder eingeschaltet.

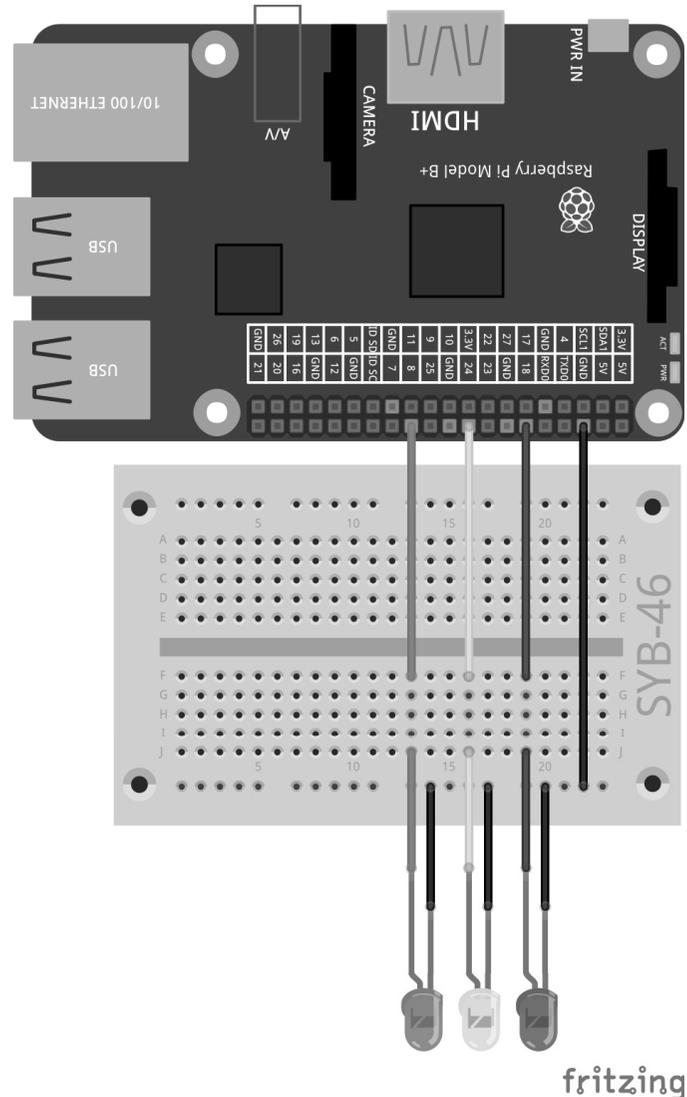
**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 4 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 6 GPIO-Verbindungskabel lang (Steckbrett - LEDs)

### Das Programm

Das Programm 07Led07 lässt drei LEDs als Lauflicht leuchten. Die Geschwindigkeit lässt sich interaktiv einstellen, während das Programm läuft. Dazu verwendet das Programm eine Variable namens **pause**, die die Pause zwischen zwei Schaltvorgängen angibt.



Das Programm 07Led07 lässt drei LEDs als Lauflicht leuchten.



Lauflicht mit drei LEDs.

**Variablen in Scratch**  
 Variablen sind kleine Speicherplätze, in denen sich ein Programm eine Zahl oder irgendetwas anderes merkt. Wenn das Programm beendet wird, werden diese Variablenspeicher automatisch wieder geleert. Variablen müssen in Scratch auf der Blockpalette **Variablen** erst einmal mit einem Klick auf **Neue Variable** angelegt werden, bevor man sie benutzen kann. Anschließend können Sie das Symbol der neu angelegten Variablen aus der Blockpalette in ein dafür vorgesehenes Feld eines Blocks im Programm ziehen. Auf der Blockpalette stehen zusätzlich verschiedene Blöcke zum Auslesen und Verändern der Variablen zur Verfügung.

Das Programm läuft komplett auf der Bühne, es enthält keine Objekte. Nach dem Start werden als Erstes die drei GPIO-Pins 18, 24 und 8 für die LEDs als Ausgänge definiert und alle ausgeschaltet.



Ein Block **setze pause auf 5** von der Blockpalette **Variablen** setzt die Pause auf 5, was im Programm einer halben Sekunde entspricht.

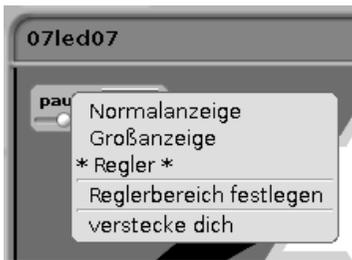
Anschließend startet eine Endlosschleife, die nacheinander die LEDs ein- und wieder ausschaltet. Da in Scratch nur Ganzzahlen als Variablenwerte interaktiv einstellbar sind, dividiert das Programm den Wert der Variablen **pause** durch 10. Verwenden Sie dazu den Block **.../...** von der Blockpalette **Operatoren**. Ziehen Sie das Variablensymbol **pause** von der Blockpalette **Variablen** in das linke Feld des Operators und tragen Sie rechts 10 ein.

Um den Wert der Variablen im Programm interaktiv ändern zu können, setzen Sie das Häkchen links neben dem Variablensymbol auf der Blockpalette **Variablen**. Dann erscheint die Variable auf der Bühne und zeigt dort immer den aktuellen Wert.

Klicken Sie mit der rechten Maustaste auf die Variable auf der Bühne und wählen Sie im Kontextmenü **Regler**. Es erscheint ein Regler, mit dem der Variablenwert eingestellt werden kann.

Klicken Sie dann noch auf **Reglerbereich festlegen** und setzen den Maximalwert auf **10**, was beim späteren Teilen durch 10 einer Sekunde Pause entspricht.

Starten Sie das Programm. Automatisch blinken die LEDs nacheinander auf. Jetzt können Sie mit dem Schieberegler bei der Variablen **pause** auf der Bühne die Geschwindigkeit einstellen. Variablen auf der Bühne werden auch im Präsentationsmodus angezeigt.



## 8. Tag



### Heute im Adventskalender

• Schaltdraht

### Schaltdraht

Heute ist Schaltdraht im Adventskalender enthalten. Damit stellen Sie kurze Verbindungsbrücken her, mit denen Kontaktreihen auf der Steckplatine verbunden werden. Schneiden Sie den Draht mit einem kleinen Seitenschneider je nach Experiment auf die passenden Längen ab. Um die Drähte besser in die Steckplatine stecken zu können, empfiehlt es sich, sie leicht schräg abzuschneiden, sodass eine Art Keil entsteht.

### LEDs dimmen

LEDs können zwei verschiedene Zustände annehmen: ein und aus. Das Gleiche gilt für die als digitale Ausgänge definierten GPIO-Ports. Demnach wäre es theoretisch nicht möglich, eine LED zu dimmen.



Links: Tastverhältnis 50 % - rechts: Tastverhältnis 20 %.

Mit einem Trick erreicht man es dennoch, die Helligkeit einer LED an einem digitalen GPIO-Port zu regeln. Lässt man eine LED schnell genug blinken, nimmt das menschliche Auge das nicht mehr als Blinken wahr. Die als Pulsweitenmodulation (PWM) bezeichnete Technik erzeugt ein pulsierendes Signal, das sich in sehr kurzen Abständen ein- und ausschaltet. Die Spannung des Signals bleibt immer gleich, nur das Verhältnis zwischen Level **False** (0 V) und Level **True** (+3,3 V) wird verändert. Das Tastverhältnis gibt das Verhältnis der Länge des eingeschalteten Zustands zur Gesamtdauer eines Schaltzyklus an. Je kleiner das Tastverhältnis, desto kürzer ist die Leuchtzeit der LED innerhalb eines Schaltzyklus. Dadurch wirkt die LED dunkler als eine permanent eingeschaltete LED.

### Warum 50 Hertz die ideale Frequenz für PWM ist

Das menschliche Auge nimmt Lichtwechsel schneller als 20 Hertz nicht mehr wahr. Da das Wechselstromnetz in Europa eine Frequenz von 50 Hertz nutzt, blinken viele Beleuchtungskörper mit dieser Frequenz, die vom Auge nicht wahrgenommen wird. Würde eine LED mit mehr als 20 Hertz, aber weniger als 50 Hertz blinken, käme es zu Interferenzen mit anderen Lichtquellen, sodass der Dimmeffekt nicht mehr gleichmäßig erschiene.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 2 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 4 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 1 Drahtbrücke

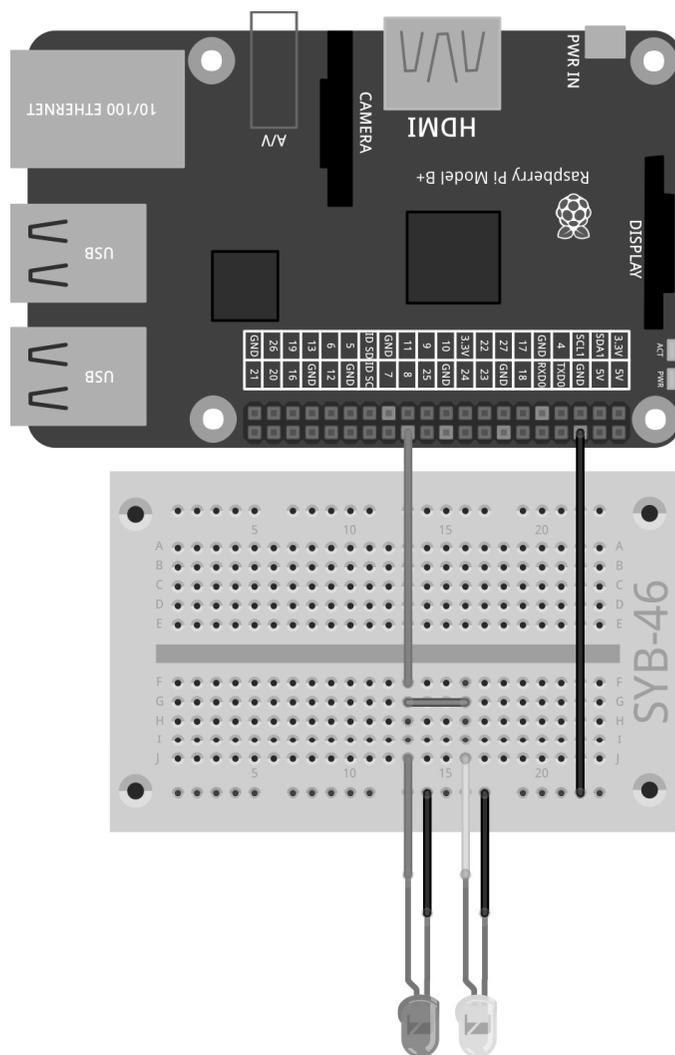
Der PWM-Effekt wirkt bei verschiedenen LED-Farben unterschiedlich stark. In der Schaltung sind die beiden LEDs am gleichen GPIO-Pin angeschlossen, um diesen Effekt deutlich zu zeigen. Dazu werden zwei Reihen des Steckbretts mit einer kurzen Drahtbrücke aus Schaltdraht verbunden.

### Das Programm

Das Programm 08Led08 dimmt die beiden LEDs anhand eines interaktiv einstellbaren Werts.



Das Programm 08Led08 dimmt die beiden LEDs.



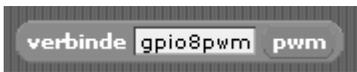
fritzing

Zwei LEDs werden über einen GPIO-Pin gedimmt.

Zunächst definiert das Programm den GPIO-Pin 8 als Ausgang mit PWM-Funktion. Bei einem so definierten Pin bietet Scratch spezielle Funktionen zur Steuerung eines PWM-Signals.

Danach wird die vorab definierte Variable **pwm** auf den Wert **0** gesetzt. In diesem Zustand ist der PWM-Pin komplett ausgeschaltet. Danach fragt eine Endlosschleife die interaktiv einstellbare Variable **pwm** ab und setzt GPIO-Pin 8 anschließend auf den neuen Wert.

Stellen Sie den Reglerbereich der Variablen auf **Min.:0** und **Max:500**. In diesem Bereich erwartet Scratch die Werte für PWM.



Um den eingestellten Wert über **sende ... an alle** an den Pin zu senden, verwenden wir den Block **verbinde ...**, der zwei beliebige Texte zu einem verbindet, indem sie hintereinandergehängt werden. Zahlen werden dabei wie Text behandelt, also nicht addiert, sondern ebenfalls einfach hintereinandergesetzt.

Schreiben Sie in das erste Feld des Blocks die Zeichenfolge `gpio8pwm` und ziehen Sie die Variable **pwm** in das zweite Feld. Auf diese Weise setzen Sie einen PWM-Pin auf einen bestimmten Wert. Ziehen Sie dann den **verbinde ...**-Block in das Textfeld des **sende... an alle**-Blocks.

Die LEDs leuchten mit der eingestellten Helligkeit, danach startet die Endlosschleife neu und fragt den neuen Wert der Variablen **pwm** ab.

## 9. Tag

### Heute im Adventskalender

• 4 GPIO-Verbindungskabel (kurz)



### Lauflicht mit Taster

Das Programm des 9. Tags ist wieder ein Lauflicht, diesmal aber mit besonderen Fähigkeiten. Die LEDs leuchten nur, wenn ein Taster gedrückt ist, und die Helligkeit jeder LED lässt sich über PWM einzeln einstellen.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 Taster; 6 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 6 GPIO-Verbindungskabel lang (Steckbrett - LEDs)

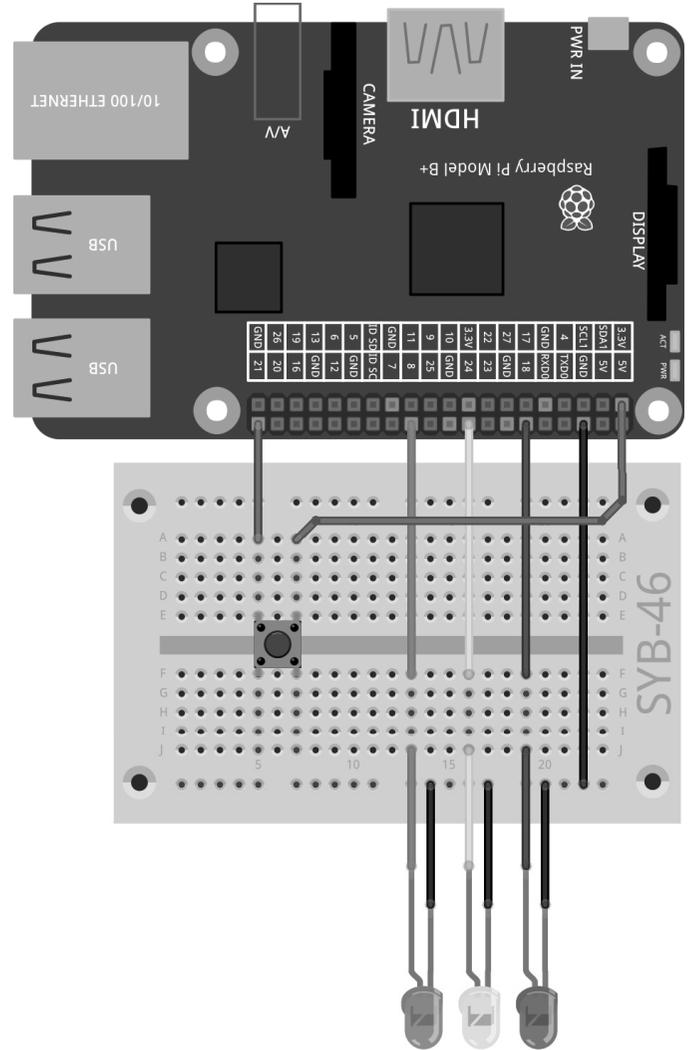
### Das Programm

Das Programm 09Led09 initialisiert am Anfang die GPIO-Pins 8, 24 und 18 als PWM-Ausgänge und den Pin 21 als Eingang mit Pull-down-Widerstand.

Eine Endlosschleife fragt ständig ab, ob an Pin 21 das Signal 1 anliegt, der Taster also gedrückt ist.

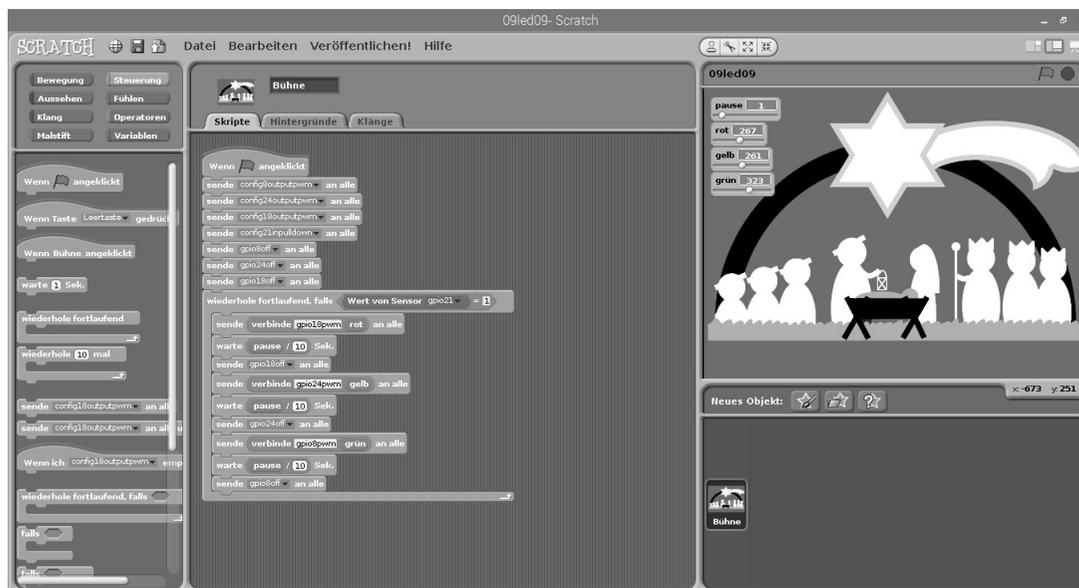
Die PWM-Werte der drei LEDs werden in den Variablen **rot**, **gelb** und **grün** gespeichert, die alle über einen Regler im Bereich **0...500** einstellbar sind. Die Pause ist wieder im Bereich **0...10** einstellbar und wird wie an Tag 7 auf Zehntelsekunden umgerechnet.

Starten Sie das Programm mit einem Klick auf das grüne Fähnchen. Stellen Sie die PWM-Werte ein und halten Sie den Taster gedrückt, um das Lauflicht zu sehen. Die Werte können auch bei nicht gedrücktem Taster geändert werden, da das Programm auch dann weiterläuft. Sobald Sie den Taster wieder drücken, startet das Lauflicht mit den aktuellen PWM-Einstellungen.



fritzing

Drei LEDs und ein Taster.



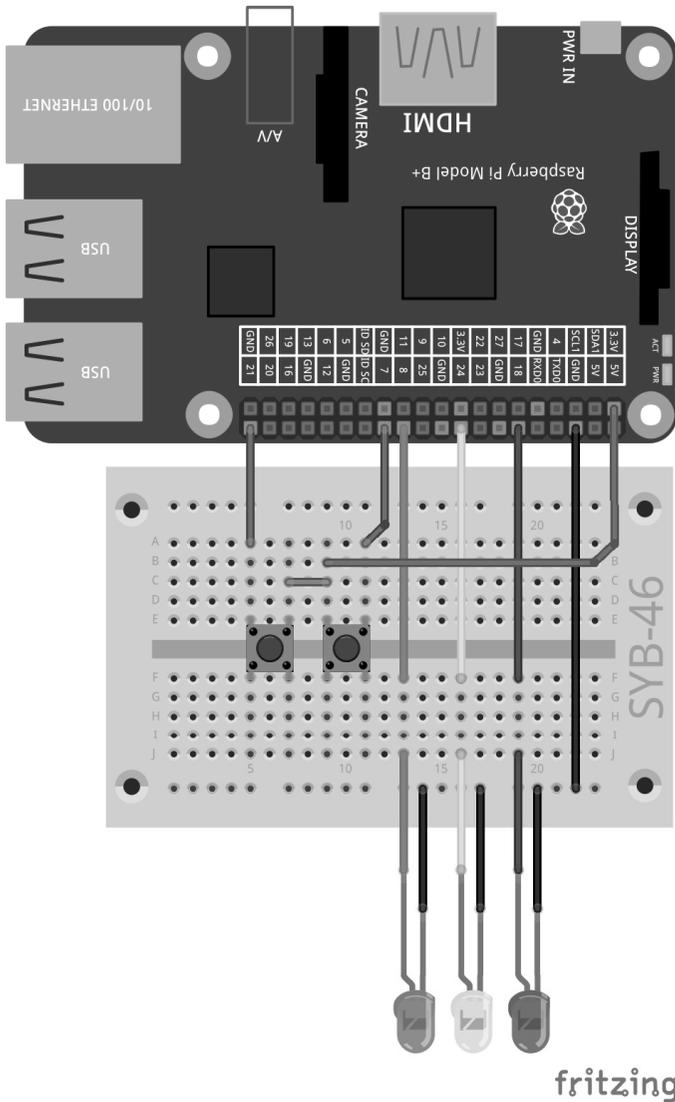
Das Programm 09Led09 steuert ein Lauflicht mit PWM-LEDs.

10. Tag

### 10. Tag

#### Heute im Adventskalender

- 1 Taster
- 1 GPIO-Verbindungskabel (kurz)



Drei LEDs und zwei Taster.

#### LEDs mit zwei Tastern umschalten

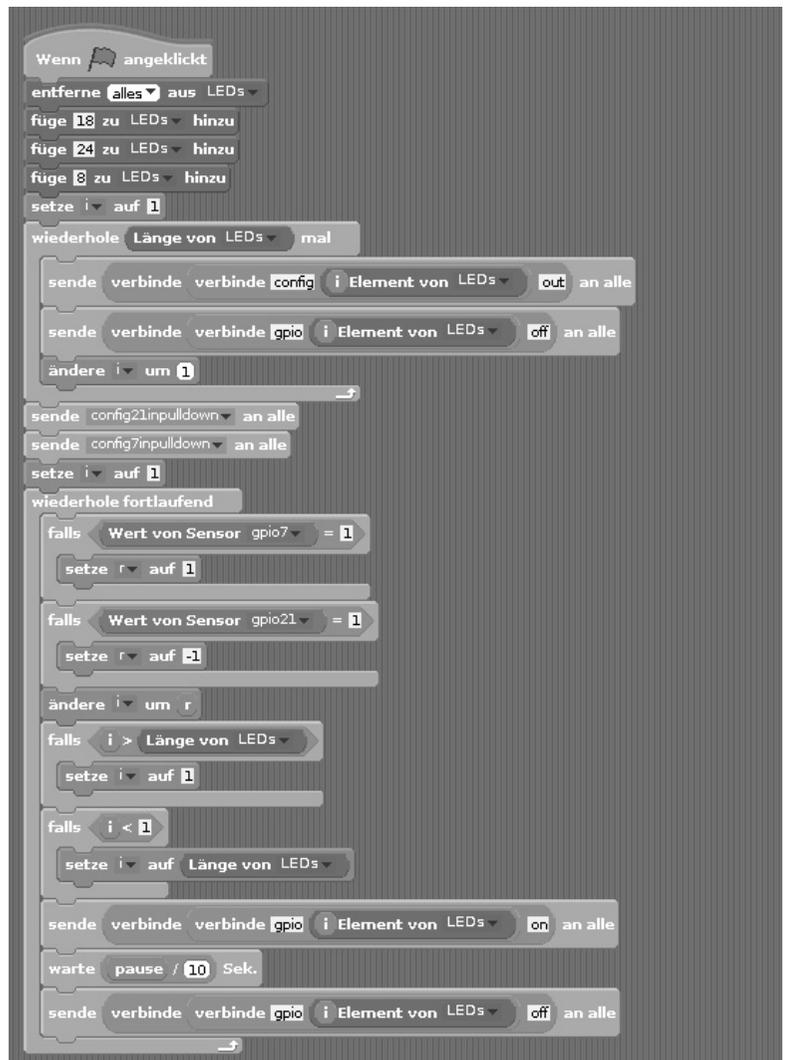
Über zwei Taster kann die Richtung des Lauflichts gewählt werden.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 2 Taster; 7 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 6 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 1 Drahtbrücke

Die Drahtbrücke wird dazu verwendet, zwei Kontaktreihen des Steckbretts zu verbinden, über die die beiden Taster am +3,3V-Pin des Raspberry Pi angeschlossen sind.

#### Das Programm

Das Programm 10led10 lässt drei LEDs als Lauflicht leuchten, wobei die Richtung über zwei Taster umgeschaltet wird. Dazu verwendet das Programm eine Liste anstatt einzelner Variablen für die drei GPIO-Pins.



Das Programm 10led10 lässt drei LEDs als Lauflicht leuchten.

Erstellen Sie für dieses Programm eine Liste mit dem Namen **LEDs**. Klicken Sie dazu auf der Blockpalette **Variablen** auf **Neue Liste**. In solchen Listen können mehrere Werte gespeichert und über die Position in der Liste adressiert werden.

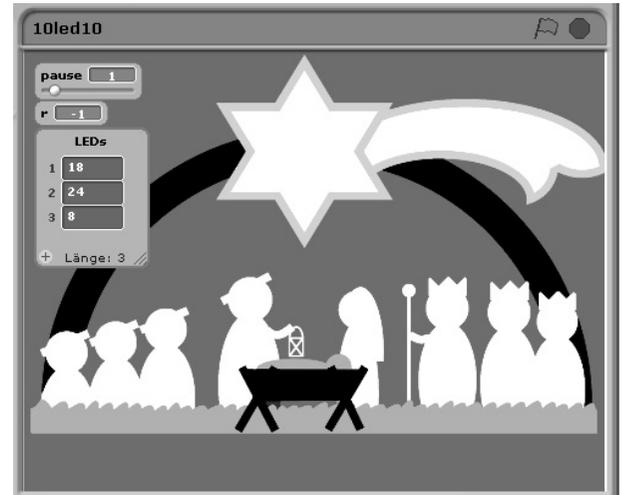
Zusätzlich werden drei einfache Variablen benötigt:

- **i** - ein einfacher Zähler, der an verschiedenen Stellen im Programm verwendet wird.
- **r** - gibt die Richtung des Lauflichts an.
- **pause** - gibt wie in früheren Programmen an, wie lange eine LED leuchtet.

Die Variable **pause** ist auf der Scratch-Bühne interaktiv einstellbar, die Variable **r** wird angezeigt, ist aber nicht einstellbar. Der Zähler **i** ist für den Benutzer nicht interessant und wird nicht angezeigt. Die Liste ist auf der Bühne zu sehen. Da sich die Werte aber nie verändern, können Sie sie auch abschalten.

Im ersten Schritt werden alle Elemente der Liste entfernt, falls aus einem früheren Programmlauf noch Werte dort vorhanden sind. Anschließend werden die Nummern der für die drei LEDs verwendeten GPIO-Pins in den ersten drei Listenpositionen gespeichert.

Eine Schleife läuft dreimal durch, adressiert dabei die drei GPIO-Pins für die LEDs und schaltet sie aus. Durch Verwendung des Blocks **Länge von ...** anstatt der Zahl 3 ist das Programm sehr leicht für längere Listen anpassbar.



Die Variable **i** steht für eine Zahl im Bereich **1...3**. Somit gibt der Block **i Element von LEDs** die Nummer des jeweiligen GPIO-Pins an. Über Kombinationen von **verbinde ...**-Blöcken werden die GPIO-Pins initialisiert, z. B. **config18out** und **gpio18off**.

Nachdem die beiden GPIO-Pins für die Taster als Eingänge initialisiert wurden, beginnt eine Endlosschleife, die als Erstes die beiden Taster abfragt. Je nachdem, welcher Taster gedrückt ist, wird die Richtung des Lauflichts in der Variablen **r** auf **1** oder **-1** gesetzt.

Je nach Richtung wird der Zähler **i**, der die LED bezeichnet, die als Nächstes leuchten soll, erhöht oder verringert.

Dieser Zähler darf nicht über die Grenzwerte 3 bzw. 1 hinauslaufen. Ist die Variable **i** nach dem Erhöhen größer als die Länge der Liste, wird sie auf **1** zurückgesetzt. Nach der letzten LED leuchtet wieder die erste. Nach dem gleichen Schema wird **i** auf die Nummer der letzten LED gesetzt, wenn sie durch Verringerung unter 1 gefallen ist.



Nachdem das Programm festgelegt hat, welche LED leuchten soll, wird diese über eine Kombination aus **Verbinde ...**-Blöcken ein- und nach der festgelegten Pause wieder ausgeschaltet. Danach startet die Endlosschleife von Neuem.





### 11. Tag

**Heute im Adventskalender**  
 • 1 RGB-LED mit Vorwiderstand

#### RGB-LEDs

Eine normale LED leuchtet immer nur in einer Farbe. Die im Adventskalender verwendeten RGB-LEDs können wahlweise in mehreren Farben leuchten. Bei RGB-LEDs



Anschlusspins einer RGB-LED.

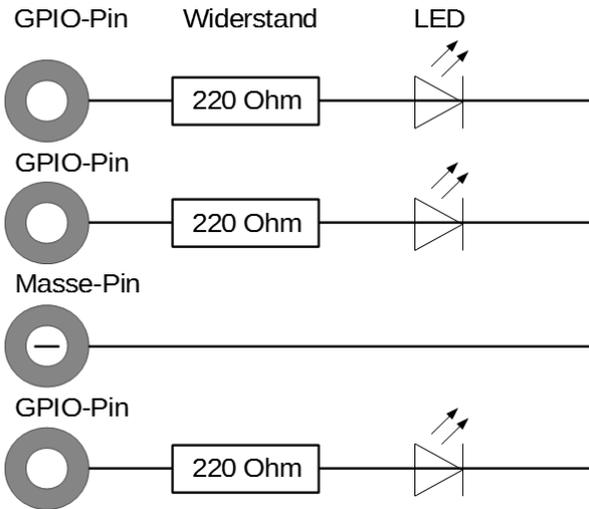
sind im Prinzip drei verschiedenfarbige LEDs in einem transparenten Gehäuse eingebaut. Jede dieser drei LEDs hat eine eigene Anode, über die sie mit einem GPIO-Pin verbunden wird. Die Kathode, die mit der Masseleitung verbunden wird, ist nur einmal vorhanden. Deshalb hat eine RGB-LED vier Anschlussdrähte.

Die Anschlussdrähte der RGB-LEDs sind unterschiedlich lang, um sie eindeutig erkennbar zu machen. Anders als bei der normalen LED ist die Kathode hier der längste Draht.

RGB-LEDs funktionieren wie drei einzelne LEDs und brauchen deshalb auch drei Vorwiderstände. Diese sind in den im Adventskalender enthaltenen RGB-LEDs ebenfalls eingebaut.

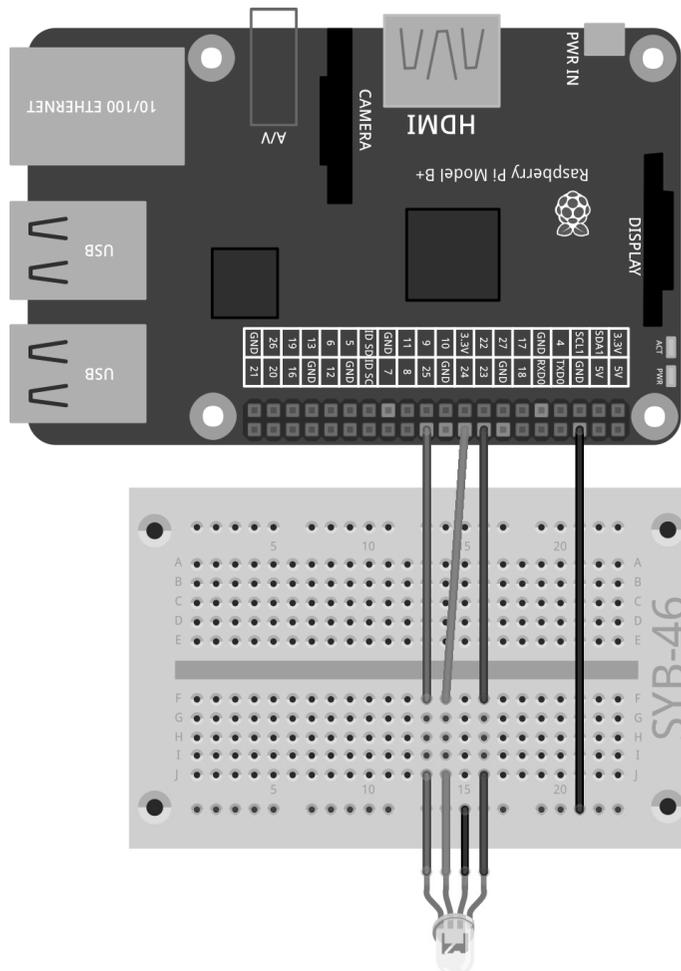
#### RGB-Farbspiele

Das Experiment des 11. Tags lässt eine RGB-LED automatisch nacheinander in verschiedenen Farben aufleuchten.



Interner Schaltplan für eine RGB-LED mit drei Vorwiderständen.

**Bauteile:** 1 Steckbrett SYB-46; 1 RGB-LED mit Vorwiderständen; 4 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 4 GPIO-Verbindungskabel lang (Steckbrett - LEDs)



fritzing

RGB-LED an den GPIO-Pins 23, 24, 25.

### Das Programm

Das Programm 11rgbLed01 schaltet nacheinander verschiedene der drei Farbkomponenten der RGB-LED ein und andere wieder aus. Da zeitweise zwei Farben eingeschaltet sind, leuchtet die RGB-LED abwechselnd in drei Grundfarben und in drei unterschiedlichen Mischfarben.

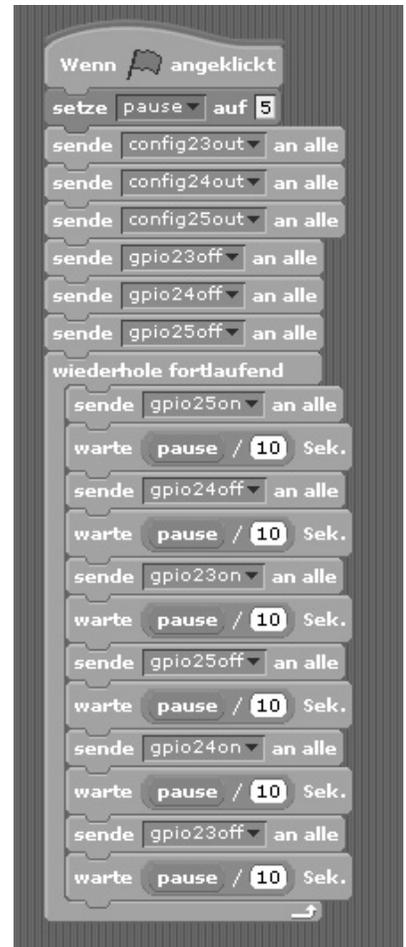
Auch hier wird die Variable **pause**, die die Zeit zwischen dem Aufleuchten der einzelnen Farben und damit die Blinkgeschwindigkeit der RGB-LED steuert, auf der Scratch-Bühne mit einem Regler angezeigt.

Nachdem die drei verwendeten GPIO-Ausgänge initialisiert wurden, werden der Reihe nach verschiedene Pins ein- und ausgeschaltet. Die Wartezeit zwischen den Schaltvorgängen kann vom Benutzer über den Regler auf der Scratch-Bühne interaktiv verändert werden.

Die Verwendung von GPIO-Nummern im Programm ist nicht besonders übersichtlich und zudem fehleranfällig, wenn man das Programm umbaut. Eine zweite Programmversion 11rgbLed02 funktioniert exakt gleich, verwendet aber Variablenamen für die GPIO-Pins. Das Programm wird dadurch zwar länger, aber übersichtlicher.



Das Programm 11rgbLed02 verwendet Variablen für die GPIO-Pinnummern.

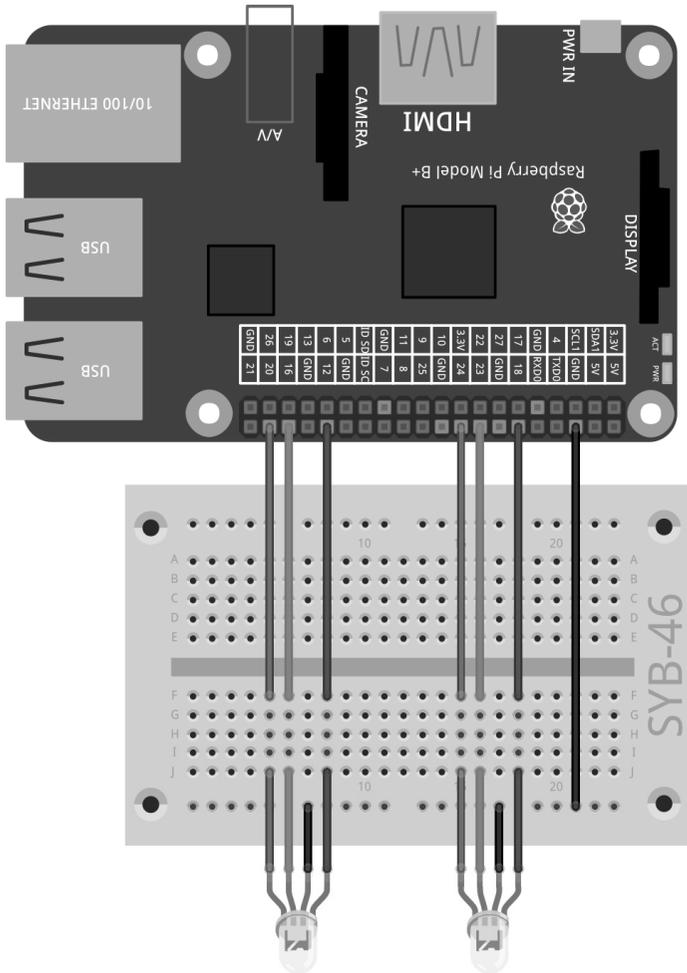


Das Programm 11rgbLed01 steuert eine RGB-LED.

12. Tag

### 12. Tag

Heute im Adventskalender  
• 1 RGB-LED mit Vorwiderstand



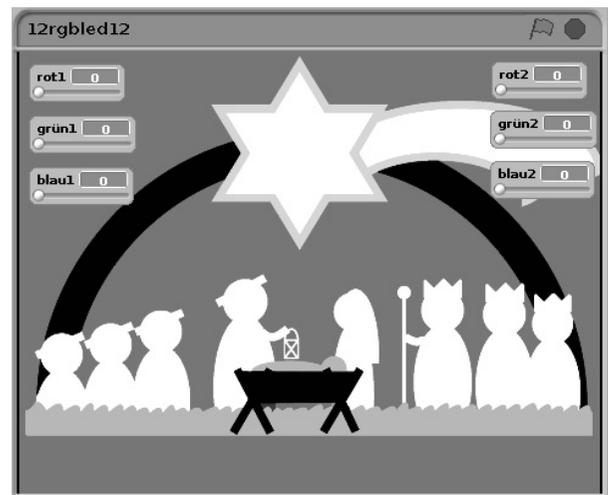
#### Farbmischung mit RGB-LEDs

Das Experiment des 12. Tags mischt auf zwei RGB-LEDs über PWM verschiedene Farben.

**Bauteile:** 1 Steckbrett SYB-46; 2 RGB-LEDs mit Vorwiderständen; 7 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 8 GPIO-Verbindungskabel lang (Steckbrett - LEDs)

#### Das Programm

Jede Farbkomponente einer RGB-LED lässt sich über PWM dimmen. Dadurch sind beliebige Mischfarben möglich. Das Programm 12rgbLed12 enthält sechs Variablen, die über Regler eingestellt werden können und die drei Farbkomponenten der beiden RGB-LEDs unabhängig steuern.



Sechs einstellbare Variablen mischen Farben auf den RGB-LEDs.

Beim Klick auf das grüne Fähnchen werden die Pinnummern für die beiden RGB-LEDs in den Variablen **r1**, **g1**, **b1** und **r2**, **g2**, **b2** gespeichert, und anschließend werden diese Pins als PWM-Ausgänge eingerichtet.

Zwei RGB-LEDs auf der Weihnachtskrippe.

fritzing

Nach der Initialisierung fragt eine Endlosschleife ständig die eingestellten Werte der Variablen **rot1**, **grün1**, **blau1** und **rot2**, **grün2**, **blau2** ab und setzt die PWM-Signale für die RGB-LEDs entsprechend.



Das Programm 12rgbLed12 mischt über PWM bunte Farben auf zwei RGB-LEDs.

### 13. Tag



#### Heute im Adventskalender

• 1 x Knete

• 1 Widerstand 20 MOhm (rot-schwarz-blau)

#### Widerstand

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 auch das Omega-Zeichen  $\Omega$  verwendet.

Die farbigen Ringe auf den Widerständen geben den Widerstandswert an. Mit etwas Übung sind Ringe deutlich leichter zu erkennen als winzig kleine 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. Dieser Toleranzring ist meistens gold- oder silberfarben - 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.

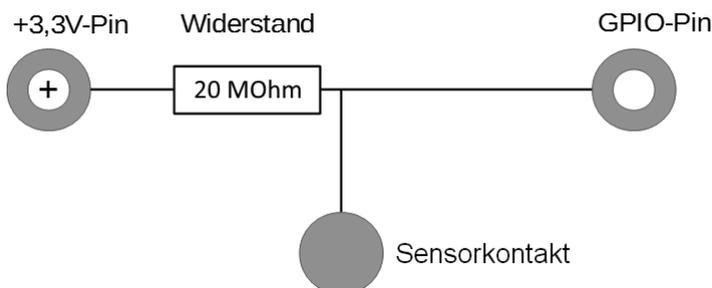
Farbe	Widerstandswert in Ohm			
	1. Ring (Zehner)	2. Ring (Einer)	3. Ring (Multiplikator)	4. Ring (Toleranz)
Silber			$10^{-2} = 0,01$	$\pm 10 \%$
Gold			$10^{-1} = 0,1$	$\pm 5 \%$
Schwarz		0	$10^0 = 1$	
Braun	1	1	$10^1 = 10$	$\pm 1 \%$
Rot	2	2	$10^2 = 100$	$\pm 2 \%$
Orange	3	3	$10^3 = 1.000$	
Gelb	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	$10^9 = 1.000.000.000$	

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

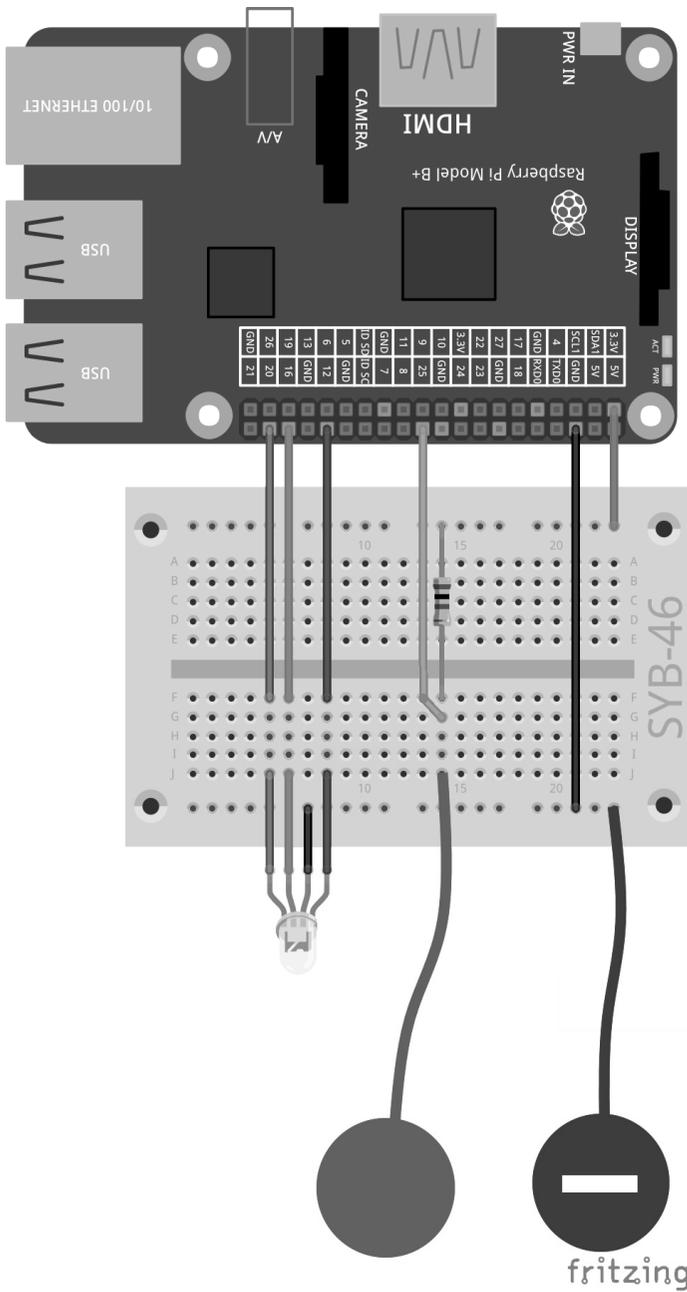
#### Sensorkontakt aus Knete

Ampeln, Türöffner, Lichtschalter und Automaten werden heute oft mit Sensorkontakten gesteuert, die man nur zu berühren braucht. Taster, die wirklich gedrückt werden müssen, werden immer seltener. Das Experiment des 13. Tags steuert eine RGB-LED über einen einfachen Sensorkontakt.

Der als Eingang geschaltete GPIO-Pin ist über einen extrem hochohmigen Widerstand (20 MOhm) mit +3,3 V verbunden, sodass ein schwaches, aber eindeutig als **High** definiertes Signal anliegt. Ein Mensch, der nicht gerade frei in der Luft schwebt, ist immer geerdet und liefert über die elektrisch leitfähige Haut einen **Low**-Pegel. Berührt dieser Mensch einen Sensorkontakt, wird das schwache **High**-Signal von dem deutlich stärkeren **Low**-Pegel der Hand überlagert und zieht den GPIO-Pin auf **Low**-Pegel. Der Pull-down-Widerstand am GPIO-Eingang muss dazu ausgeschaltet sein.



Wie hoch allerdings der Widerstand zwischen Hand und Masse wirklich ist, hängt von vielen Dingen ab, unter anderem von Schuhen und Fußböden. Barfuß im nassen Gras ist die Verbindung zur Masse der Erde am besten, aber auch auf Steinfußböden funktioniert es meistens gut. Holzfußböden isolieren stärker, Kunststoffbodenbeläge sind oft sogar positiv aufgeladen. Für den Fall, dass der Sensorkontakt nicht funktioniert, enthält die Schaltung einen zweiten Knetekontakt, der mit der Masseschiene des Steckbretts verbunden ist. Berühren Sie diesen und den eigentlichen Sensor gleichzeitig. Dann ist die Masseverbindung auf jeden Fall hergestellt.



Knetkontakt steuert RGB-LED.

Knete leitet den Strom etwa so gut wie menschliche 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 die Hand den Kontakt berührt, ist deutlich größer. So kommt es nicht so leicht zu einem „Wackelkontakt“. Stecken Sie ein Stück abisolierten Scheldraht in ein Stück Knete. Das andere Drahtende stecken Sie in das Steckbrett.

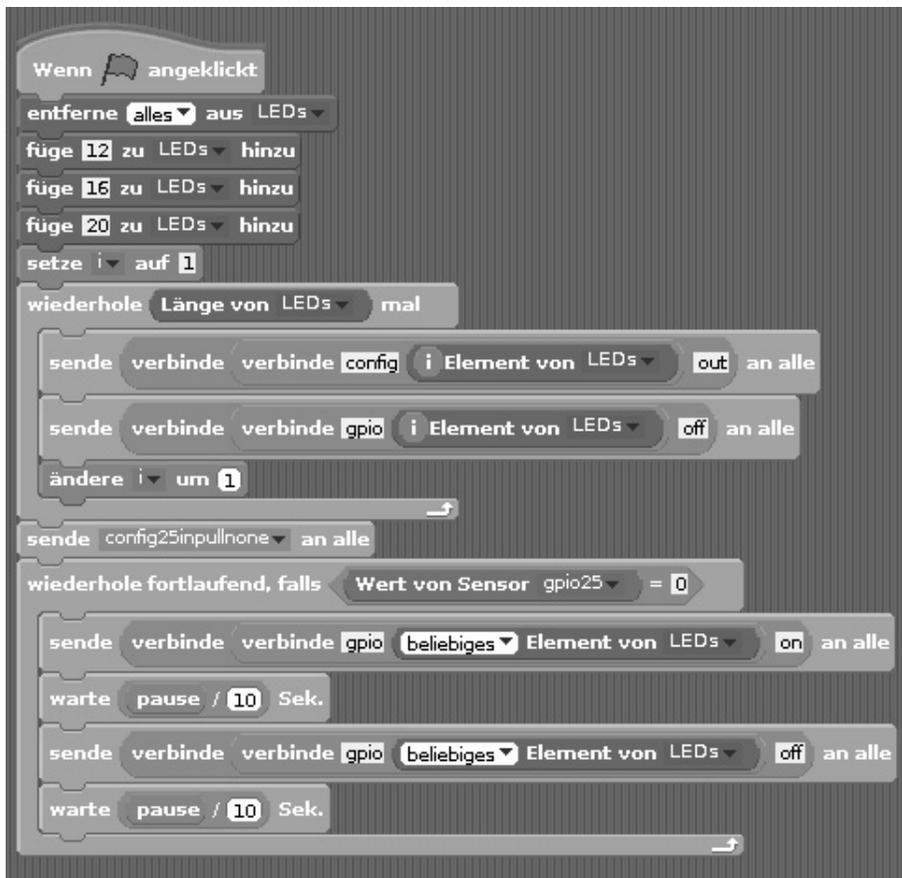
**RGB-LED blinkt zufällig**

Im Experiment des 13. Tags blinkt eine RGB-LED, solange der Sensorkontakt berührt wird, in zufälligen bunten Farben.

**Bauteile:** 1 Steckbrett SYB-46; 1 RGB-LED mit Vorwiderständen; 1 Widerstand 20 MOhm; 6 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 4 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 2 Knetkontakte

**Das Programm**

Das Programm `13rgbled13` verwendet eine Liste für die drei GPIO-Pins der RGB-LED. Eine Schleife initialisiert die Pins als Eingänge und schaltet sie alle aus. Der Block `sende config25inpullnone` initialisiert GPIO-Pin 25 für den Sensorkontakt als Eingang ohne Pull-down-Widerstand.



Das Programm 13rgbLed13 steuert eine RGB-LED über einen Sensorkontakt.

Die RGB-LED soll nur blinken, wenn der Sensorkontakt berührt wird. Das erledigt eine Schleife vom Typ **wiederhole fortlaufend falls ...** Wird der Sensorkontakt berührt, wird er mit Masse verbunden, und das Signal am GPIO-Eingang geht auf 0. Die Bedingung in der Schleife verwendet wieder den Block **Wert von Sensor ...** Starten Sie das noch unfertige Programm einmal mit einem Klick auf das grüne Fähnchen, um den GPIO-Eingang zu initialisieren, damit dieser in der Auswahlliste der Sensoren erscheint.

In jedem Schleifendurchlauf wird eine zufällig gewählte Farbkomponente der RGB-LED eingeschaltet. Dazu wird ein Block **beliebiges Element von LEDs** verwendet.

### Wie entstehen Zufallszahlen?

Gemeinhin denkt man, in einem Programm könne nichts zufällig geschehen – wie also 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 ohne jede Regelmäßigkeit, wenn man den Divisor regelmäßig erhöht. Dieses Ergebnis ist zwar scheinbar zufällig, lässt sich aber durch ein identisches Programm oder den mehrfachen Aufruf des gleichen Programms jederzeit reproduzieren. Nimmt man aber eine aus einigen dieser Ziffern zusammengebaute Zahl und teilt sie wiederum durch eine Zahl, die sich aus der aktuellen Uhrzeitsekunde oder dem Inhalt einer beliebigen Speicherstelle des Computers ergibt, kommt ein Ergebnis heraus, das sich nicht reproduzieren lässt und daher als Zufallszahl bezeichnet wird.

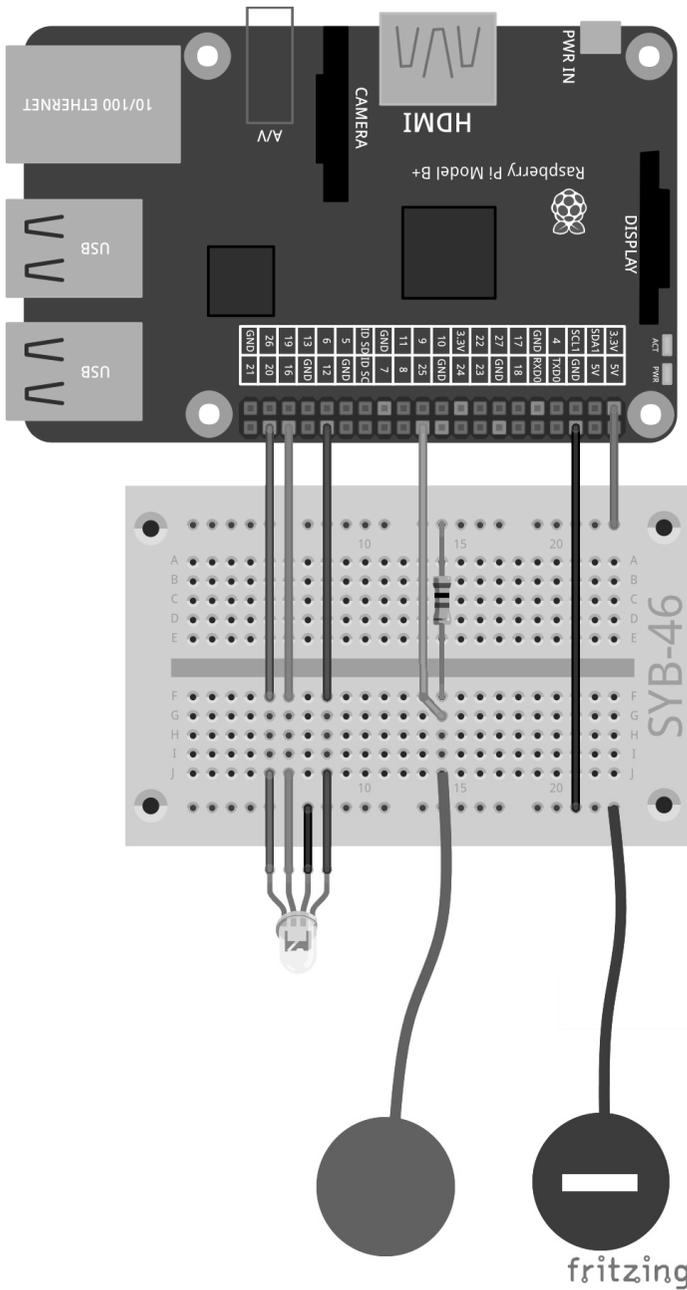
Nach einer wieder über einen Regler einstellbaren Pause wird eine zufällig gewählte Farbkomponente der RGB-LED ausgeschaltet. Durch den verwendeten Algorithmus kann es passieren, dass die RGB-LED kurzfristig ganz ausgeschaltet wird oder zweimal hintereinander die gleiche Farbe zeigt.

14. Tag

### 14. Tag

#### Heute im Adventskalender

• 1 x Knete



Knetekontakt steuert RGB-LED.

Mit der zweiten Knete können Sie die beiden Sensorkontakte am GPIO-Pin und am Massepin in unterschiedlichen Farben bauen.

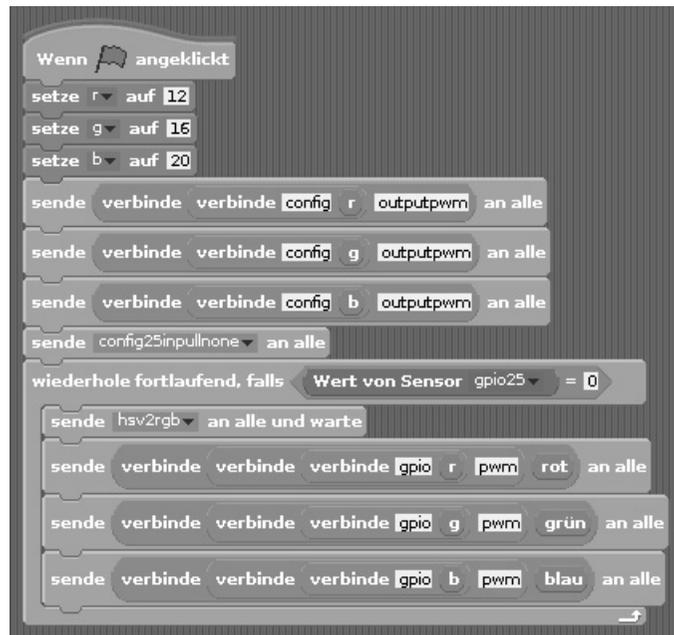
#### RGB-LED-Lichteffekte

Das Programm des 14. Tags lässt die RGB-LED in einer beliebigen Farbe des Regenbogens leuchten, wenn der Sensorkontakt berührt wird. Die Farbe wird über einen einzelnen Regler ausgewählt anstatt über drei RGB-Werte. Lässt man den Sensor los, bleibt die zuletzt angezeigte Farbe stehen.

**Bauteile:** 1 Steckbrett SYB-46; 1 RGB-LED mit Vorwiderständen; 1 Widerstand 20 MOhm; 6 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 4 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 2 Knetkontakte

#### Das Programm

Das Programm 14rgbled14 besteht aus zwei unabhängigen Programmblöcken. Der Block, der beim Klick auf das grüne Fähnchen gestartet wird, entspricht weitgehend den bereits bekannten Programmen. Die drei Pins der RGB-LED werden als PWM-Ausgänge initialisiert, Pin 25 für den Sensorkontakt als Eingang ohne Pull-down-Widerstand.



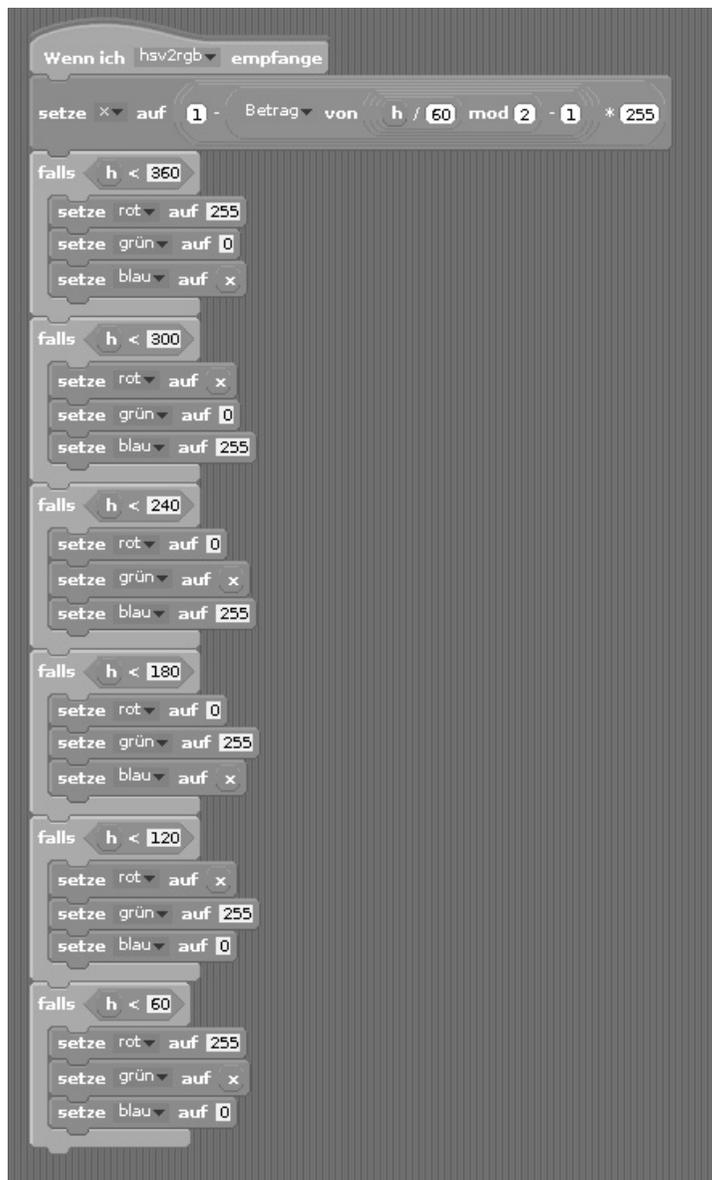
Das Programm 14rgbled14 lässt die RGB-LED in bunten Regenbogenfarben leuchten.

Eine Endlosschleife prüft, ob der Sensorkontakt berührt wird. Ist das der Fall, sendet ein Block **sende ... an alle und warte** die Nachricht **hsv2rgb**. Auf diese Weise lassen sich in Scratch Unterprogramme bauen, die aus dem Hauptprogramm heraus aufgerufen werden. Das Hauptprogramm wartet, bis das Unterprogramm abgearbeitet ist, und setzt anschließend die GPIO-Pins der RGB-LED auf die errechneten PWM-Werte.

### HSV- und RGB-Farbsystem

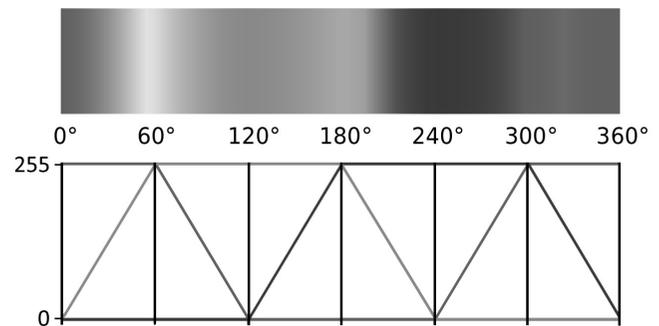
Das RGB-Farbsystem, das bisher in allen Programmen verwendet wurde, beschreibt Farben als die drei Komponenten Rot, Grün und Blau, die miteinander gemischt werden. Für Menschen ist es relativ schwierig, sich eine Mischfarbe vorzustellen. Im Gegensatz dazu beschreibt das HSV-Farbsystem die Farben über die Werte H = Hue (Farbwert), S = Saturation (Sättigung) und V = Value (Helligkeitswert). Durch eine einfache Veränderung des H-Werts können alle Farben des Farbspektrums in voller Intensität beschrieben werden, wenn man die beiden anderen Werte auf Maximum einstellt.

Die Nachricht **hsv2rgb** startet einen Programmblock, der aus dem eingestellten H-Wert die drei Farbkomponenten R, G und B errechnet. Der H-Wert kann entsprechend den Gradzahlen auf einem Farbkreis Werte zwischen 0 und 360 annehmen. Die Werte S = Saturation (Sättigung) und V = Value (Helligkeitswert) werden automatisch auf den Maximalwert gesetzt.



Unterprogramm zur Umrechnung von HSV- in RGB-Werte.

Der Block **wenn ich ... empfangen** startet ein Unterprogramm, wenn eine bestimmte Nachricht empfangen wurde. Diese Nachricht kann von einem Block **sende ... an alle und** oder **sende ... an alle und warte** gesendet worden sein.



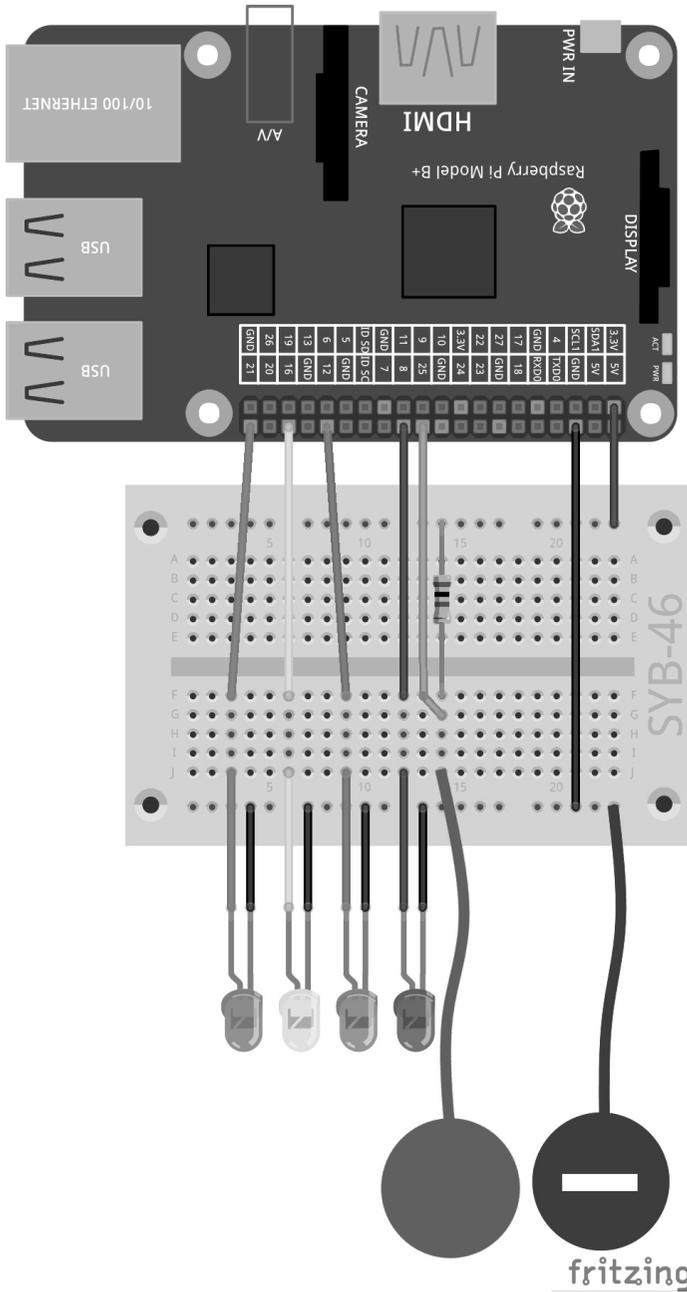
Die Grafik zeigt, wie der H-Wert einer HSV-Farbe in RGB-Werte umgerechnet wird.

15. Tag

### 15. Tag

#### Heute im Adventskalender

• 1 LED orange mit Vorwiderstand



Knetekontakt steuert Lauflicht.

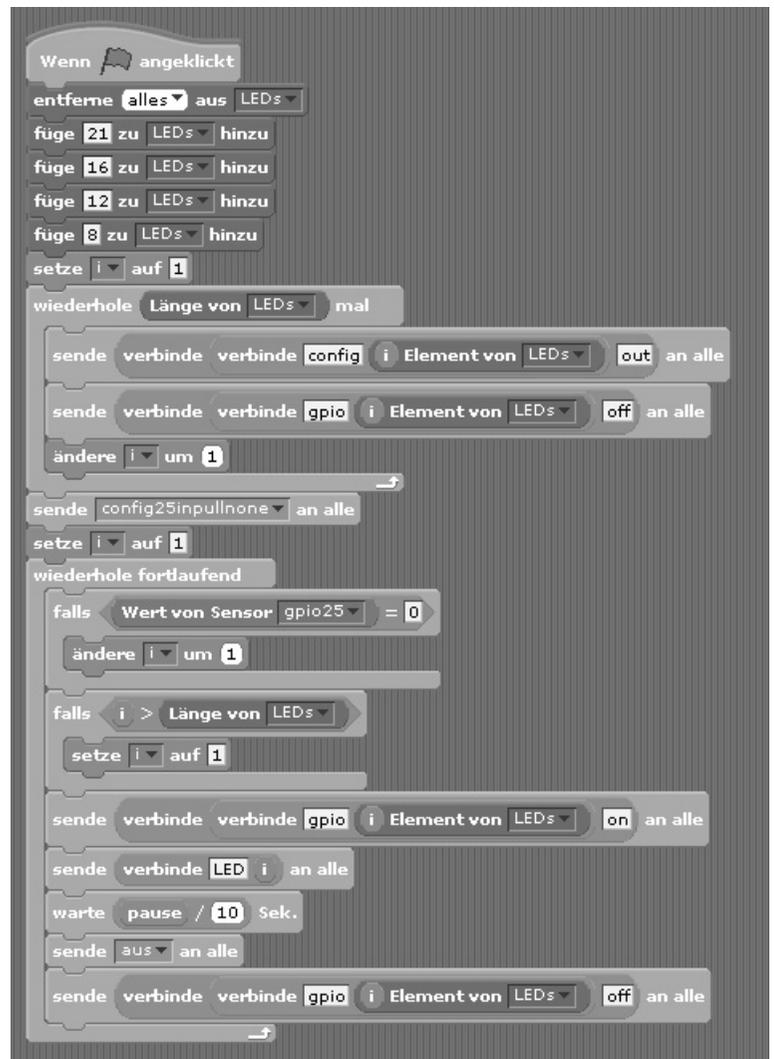
#### Lauflicht mit Knetekontakt steuern

Das Experiment des 15. Tags ist wieder ein Lauflicht, das diesmal über einen Sensorkontakt aus Knete gesteuert wird. Zusätzlich zu den echten LEDs auf der Weihnachtskrippe leuchten farbige Symbole auf der Scratch-Bühne. Solange der Sensorkontakt berührt wird, blinken die LEDs abwechselnd als Lauflicht. Wird der Sensorkontakt losgelassen, leuchtet die zuletzt eingeschaltete LED weiter.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED orange mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 Widerstand 20 MOhm; 7 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 8 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 2 Knetekontakte

#### Das Programm

Das Programm 15led15 besteht aus mehreren unabhängigen Blöcken. Der Programmblock der Bühne wird beim Klick auf das grüne Fähnchen gestartet, legt eine Liste mit den GPIO-Pins der vier LEDs an, initialisiert diese als Ausgänge und Pin 25 als Eingang ohne Pull-down-Widerstand.



Der Programmblock der Bühne enthält die Hauptschleife des Programms.

Die Endlosschleife fragt als Erstes ab, ob der Sensorkontakt berührt wird. Ist das der Fall, wird der Zähler *i* um 1 erhöht, um die nächste LED aus der Liste leuchten zu lassen.

Wird **i** größer als die Länge der Liste, wird **i** wieder auf **1** gesetzt, und das Lauflicht beginnt erneut bei der ersten LED.

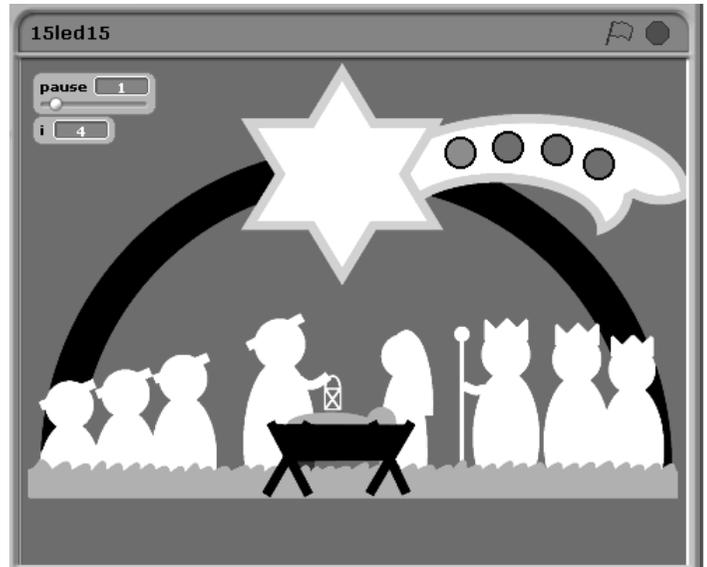
Jetzt wird der der Variablen **i** in der Liste entsprechende GPIO-Pin eingeschaltet. Anschließend baut ein **verbinde... ..**-Block eine Nachricht aus dem Wort **LED** und der aktuellen Nummer aus der Variablen **i** zusammen: **LED1**, **LED2**, **LED3** oder **LED4**. Diese Nachricht wird gesendet, um das Symbol der LED auf der Scratch-Bühne einzuschalten.

Nach einer einstellbaren Pause wird die Nachricht **aus** gesendet, die alle LED-Symbole auf der Scratch-Bühne ausschaltet. Im Gegensatz zum Einschalten braucht hier keine LED-Nummer angegeben zu werden. Das Programm schaltet einfach alle LED-Symbole aus, unabhängig davon, ob sie vorher eingeschaltet waren oder nicht.

Anschließend wird die gerade leuchtende LED über ihren GPIO-Pin ausgeschaltet, bevor der nächste Durchlauf der Schleife beginnt.

Die vier LED-Symbole auf dem Schweif des Sterns sind vier einzelne Scratch-Objekte. Laden Sie das Bild `kreis_gruen.png` als neues Objekt in die Szene. Sobald das Objekt fertig gebaut ist, werden daraus die anderen Objekte dupliziert.

Markieren Sie das neue Objekt in der Objektliste und kopieren Sie auf der Registerkarte **Kostüme** ein zweites Kostüm. Füllen Sie hier den Kreis im Malprogramm mit grauer Farbe.



Die vier LEDs auf der Scratch-Bühne.



Die Kostüme des grünen LED-Symbols.

Benennen Sie das Objekt in **grün** um und die beiden Kostüme in **an** und **aus**. Legen Sie danach auf der Registerkarte **Skripte** zwei Programmblöcke für dieses Objekt an.



Die Programmblöcke des grünen LED-Symbols.

Empfängt das Objekt die Nachricht **LED1**, wird auf das Kostüm **an** gewechselt. Das Symbol leuchtet in Grün.

Empfängt das Objekt die Nachricht **aus**, wird auf das Kostüm **aus** gewechselt. Das Symbol erscheint in Grau.

Duplizieren Sie das Objekt mit einem Rechtsklick auf der Objektpalette dreimal. Benennen Sie die neuen Objekte in **gelb**, **orange** und **rot** um und schieben Sie sie an die entsprechenden Positionen auf dem Schweif des Sterns.

Klicken Sie bei jedem Objekt im Kostüm **an** auf den Button **Bearbeiten** und füllen Sie den Kreis mit Gelb, Orange bzw. Rot. Ändern Sie dann noch auf der Registerkarte **Skripte** die Nachricht im ersten Block auf **LED2**, **LED3** bzw. **LED4**.

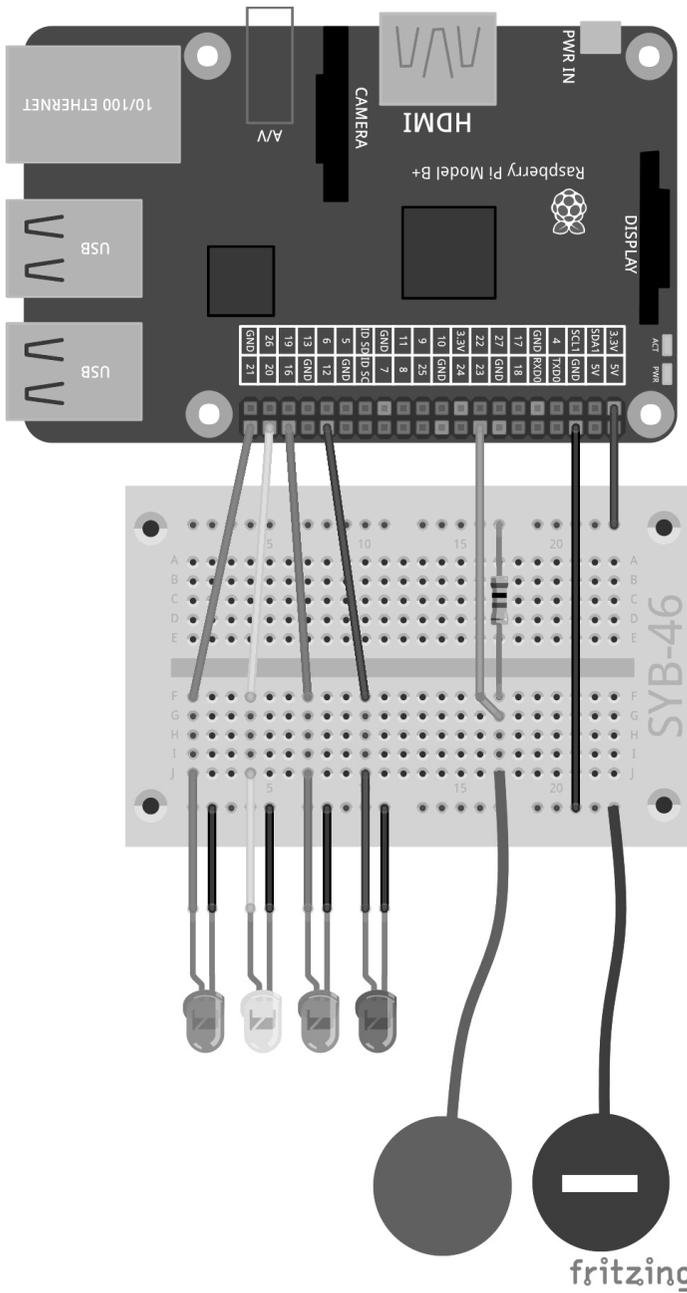
Starten Sie das Programm mit einem Klick auf das grüne Fähnchen und berühren Sie den Sensorkontakt. Das Lauflicht läuft los.

16. Tag

## 16. Tag

### Heute im Adventskalender

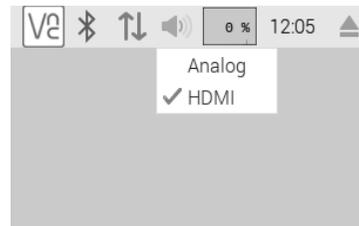
- 4 GPIO-Verbindungskabel (lang)



Knetkontakt steuert vier LEDs.

### Der Raspberry Pi erzeugt Töne

Der Raspberry Pi kann über einen HDMI-Monitor, einen externen Lautsprecher oder Kopfhörer an der analogen 3,5-mm-Klinkenbuchse Musik abspielen. Bei Computermonitoren mit DVI-Anschluss, die am HDMI-Ausgang angeschlossen sind, muss am analogen Ausgang ein Lautsprecher angeschlossen werden, da das Audiosignal nicht über das DVI-Kabel übertragen wird. Klicken Sie mit der rechten Maustaste oben rechts auf das Lautsprechersymbol, um den Audioausgang auszuwählen.



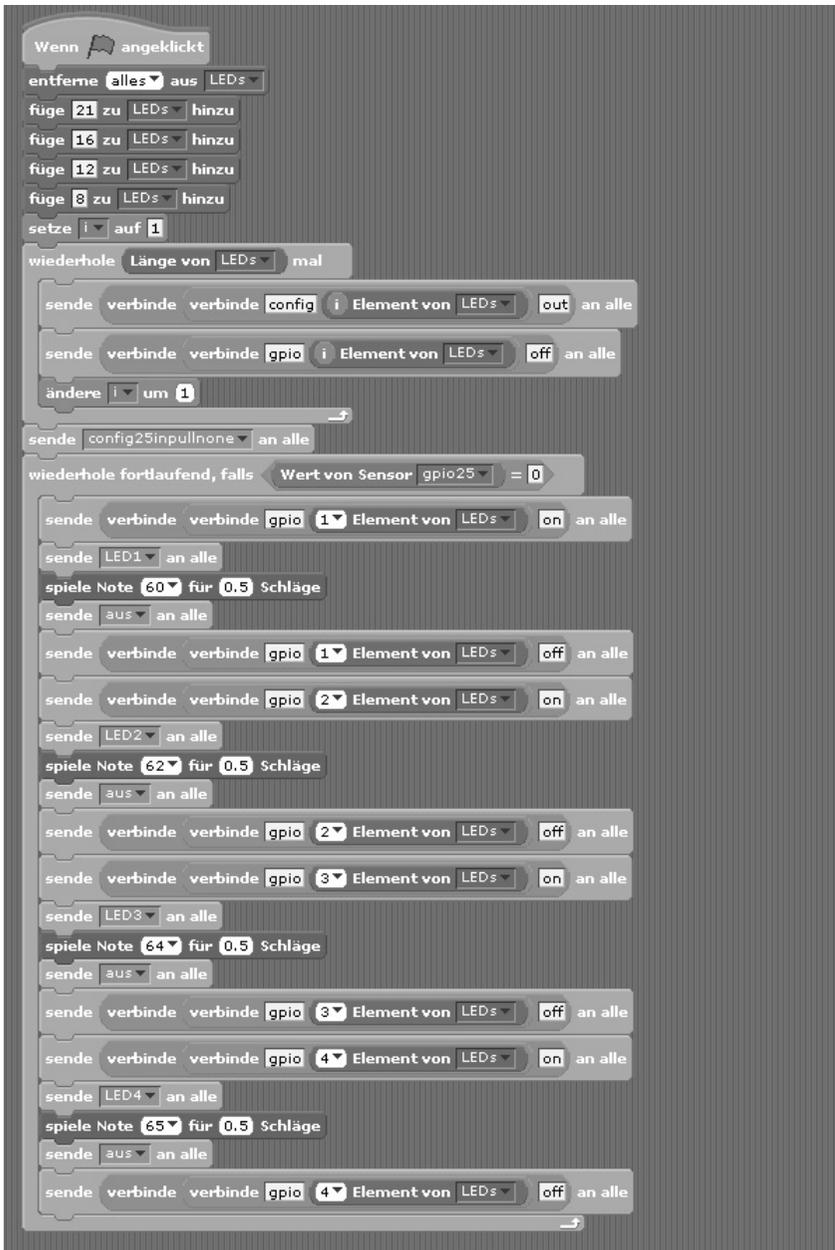
Audioausgang auswählen.

Jedes Mal, wenn man den Knetkontakt berührt, spielt der Raspberry Pi eine Tonfolge ab und lässt die LEDs dazu nacheinander leuchten.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED orange mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 Widerstand 20 MOhm; 7 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 8 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 2 Knetkontakte

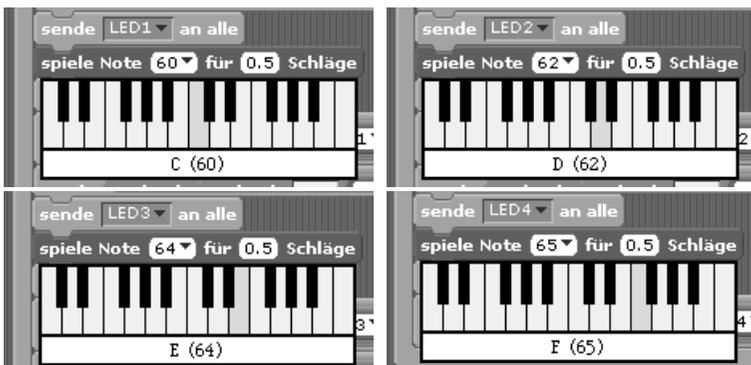
### Das Programm

Das Programm 16led16 funktioniert ähnlich wie das Programm des vorherigen Tags und lässt vier LEDs in kurzen Abständen nacheinander aufblincken, wenn der Knetkontakt berührt wird. Zu jeder LED ertönt ein Ton.



Das Programm 16Led16 lässt vier LEDs in kurzen Abständen aufblinken und spielt dazu Töne ab.

Die Nummern der LEDs sind wieder in einer Liste gespeichert. In der Hauptschleife des Programms befinden sich aber trotzdem für jede LED eigene Programmblöcke, damit Sie im Block **spiele Note ... für ... Schläge** den Ton jeder einzelnen LED über eine Klaviatur interaktiv festlegen können. Das Programm verwendet die Töne C, D, E und F.



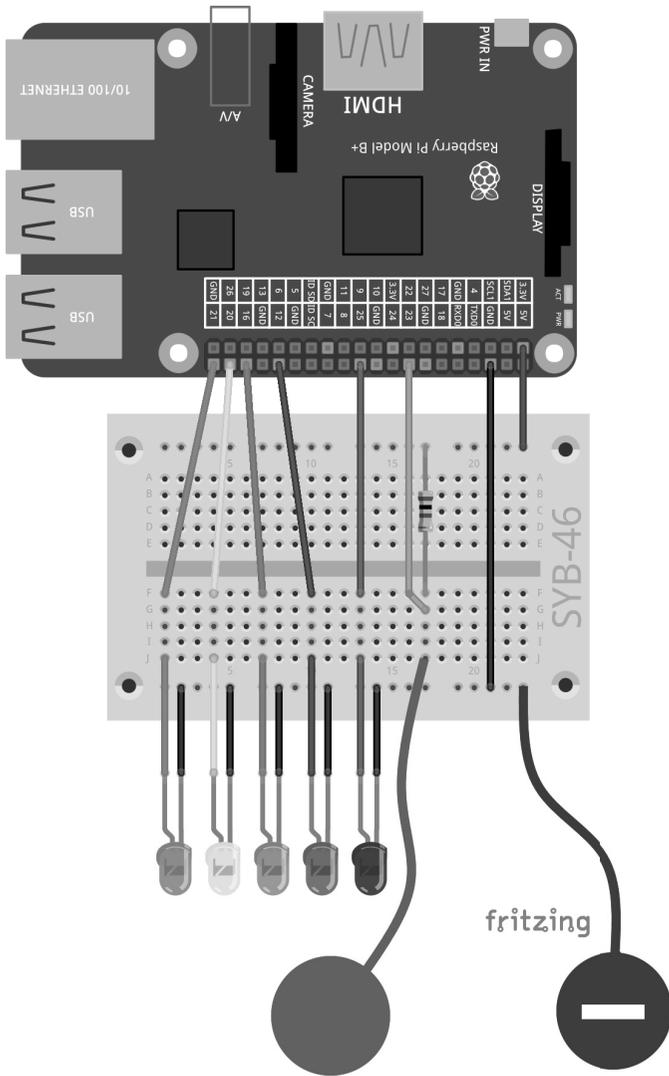
Die vier verwendeten Töne.

17. Tag

### 17. Tag

#### Heute im Adventskalender

• 1 LED lila mit Vorwiderstand



Laufflicht mit fünf LEDs.

Die Initialisierung der GPIO-Pins läuft wie im vorhergehenden Programm. Die Hauptschleife des Programms setzt in jedem Durchlauf, wenn der Sensorkontakt berührt wird, den Zähler **i** auf 1. Eine innere Schleife läuft fünfmal und schaltet in jedem Durchlauf eine der LEDs und das entsprechende LED-Symbol auf der Bühne an. Danach ertönt der in der Liste **Ton** unter der passenden Nummer gespeicherte Ton. Dieser Ton definiert gleichzeitig, wie lange die LED leuchtet. Danach werden LED-Symbol und LED ausgeschaltet, der Schleifenzähler wird um 1 erhöht, und der nächste Durchlauf wird mit der nächsten LED gestartet.

#### Erweitertes Laufflicht

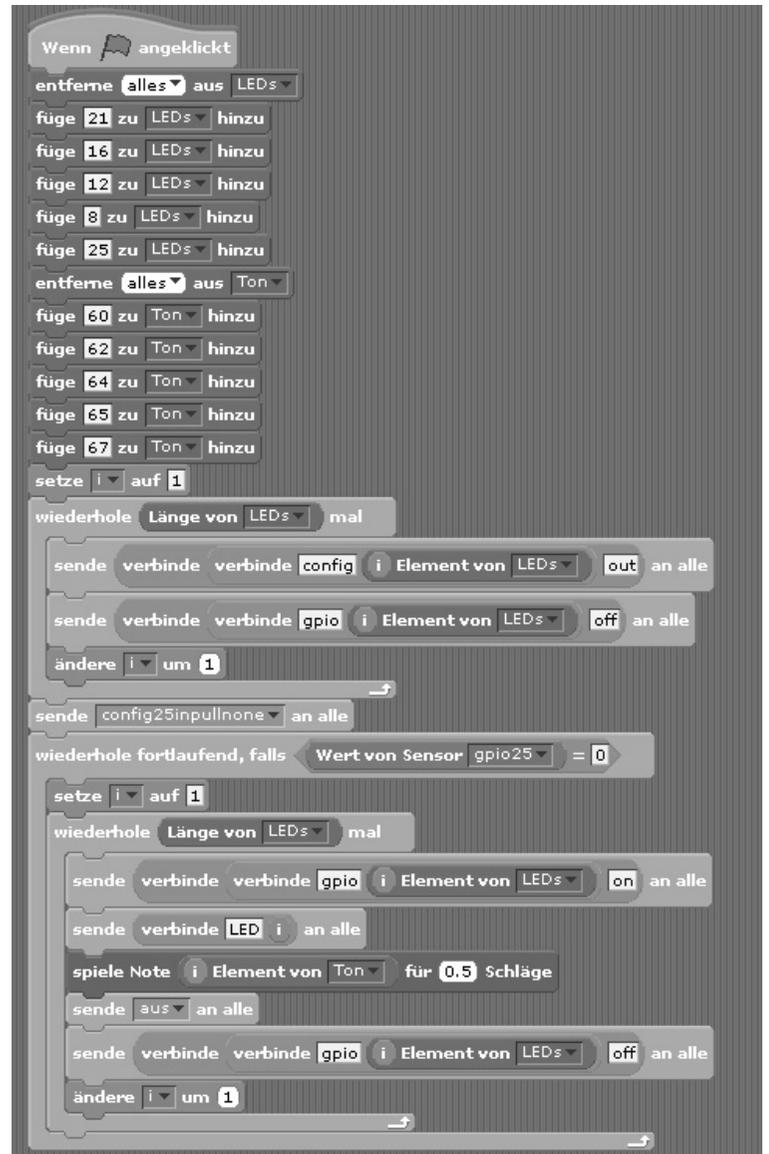
Das Experiment des 17. Tags erweitert das Laufflicht um eine fünfte LED. Auch hier wird eine Tonfolge abgespielt.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED orange mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 LED lila mit Vorwiderstand; 1 Widerstand 20 MOhm; 8 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 10 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 2 Knetkontakte

#### Das Programm

Im Programm 17Led17 werden die Töne auch über eine Liste abgespielt, was die Hauptschleife des Programms deutlich verkürzt.

Am Anfang wird zusätzlich zur Liste **LEDs** eine Liste **Ton** mit den Tönen angelegt, die zu den einzelnen LEDs abgespielt werden. Bei dieser Methode haben Sie keine Klaviatur zur Verfügung. Probieren Sie die Töne am besten mit einem **Spieler Note ... für ... Schläge** aus. Das Programm verwendet die Töne **C(60), D(62), E(64), F(65), G(67)**.



Laufflicht aus fünf LEDs mit einer Tonfolge.

# 18. Tag

## Heute im Adventskalender

• 3 GPIO-Verbindungskabel (kurz)



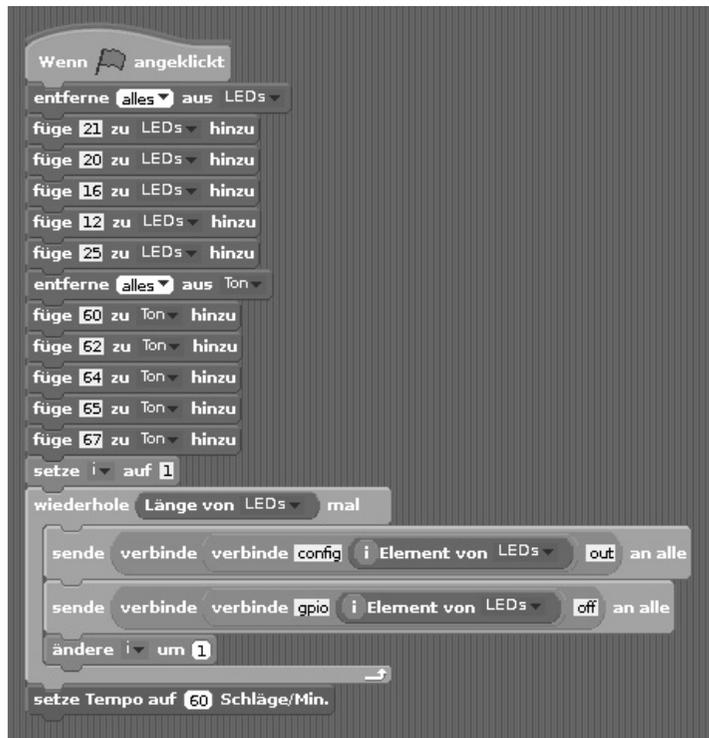
### Drei verschiedene Lauflichtmuster

Das Programm des 18. Tags zeigt auf den fünf LEDs neben dem einfachen Lauflicht zwei weitere Leuchtmuster an. Das Programm 18led18 enthält dazu drei Buttons auf der Scratch-Bühne, mit denen man diese Muster einschalten kann.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED orange mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 LED lila mit Vorwiderstand; 6 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 10 GPIO-Verbindungskabel lang (Steckbrett - LEDs)

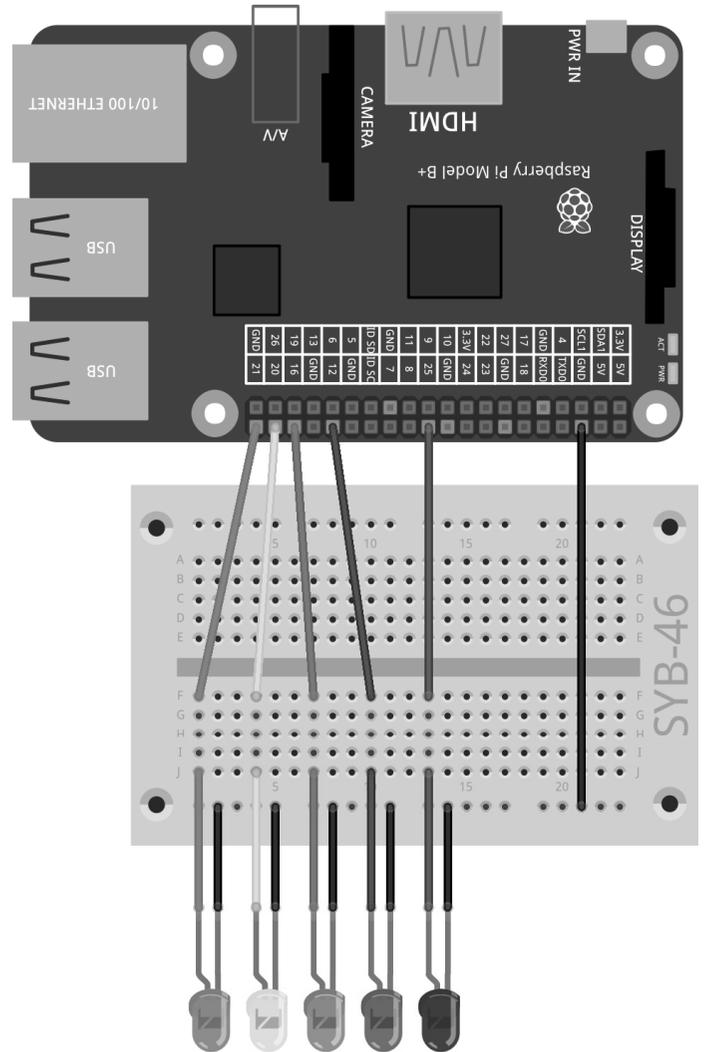
### Das Programm

Das Programm 18led18 initialisiert auf der Bühne nur die beiden Listen für LEDs und Töne sowie die GPIO-Pins. Am Ende wird eine Geschwindigkeit für die Musik festgelegt. Musik wird in einem Tempo abgespielt, das in sogenannten Schlägen angegeben wird, wohingegen alle anderen Zeitangaben in Scratch in Sekunden gemessen werden. Der Block **setze Tempo auf ... Schläge/Min.** legt die Geschwindigkeit für das Abspielen von Musik fest.



Die Programmblöcke der Bühne.

Das Programm enthält, wie frühere Programme, fünf Objekte für die fünf LEDs. Damit diese in verschiedenen Mustern gesteuert werden können, wobei nicht immer alle auf einmal ausgeschaltet werden, reagiert jedes Objekt auf ein eigenes Ausschaltsignal, in der Abbildung z. B. aus5.



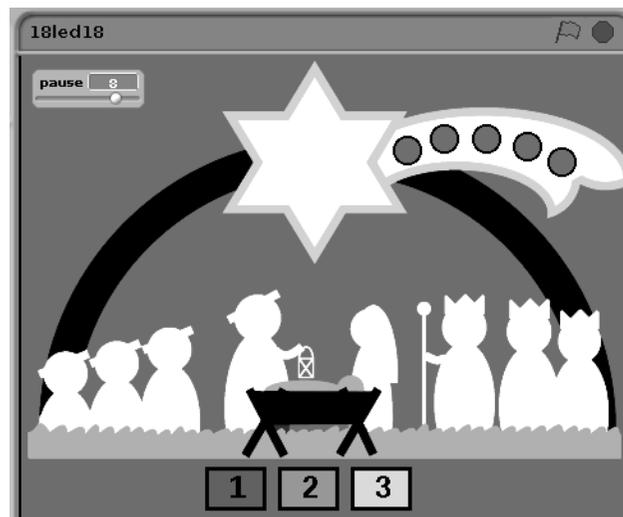
fritzing

Das Lauflicht leuchtet in verschiedenen Mustern.



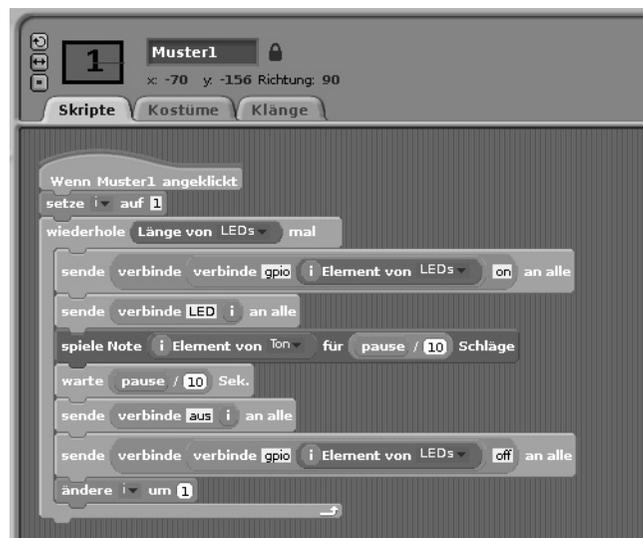
Programmblöcke des Objekts lila.

Zusätzlich zu den Objekten für die LEDs sind noch drei Buttons im Programm. Diese sind eigene Objekte, die beim Anklicken eines der drei vordefinierten LED-Muster abspielen. Die Zeit zwischen den einzelnen Blinkvorgängen wird wieder über die interaktiv einstellbare Variable **pause** geregelt.



Die drei Buttons schalten die Blinkmuster der LEDs.

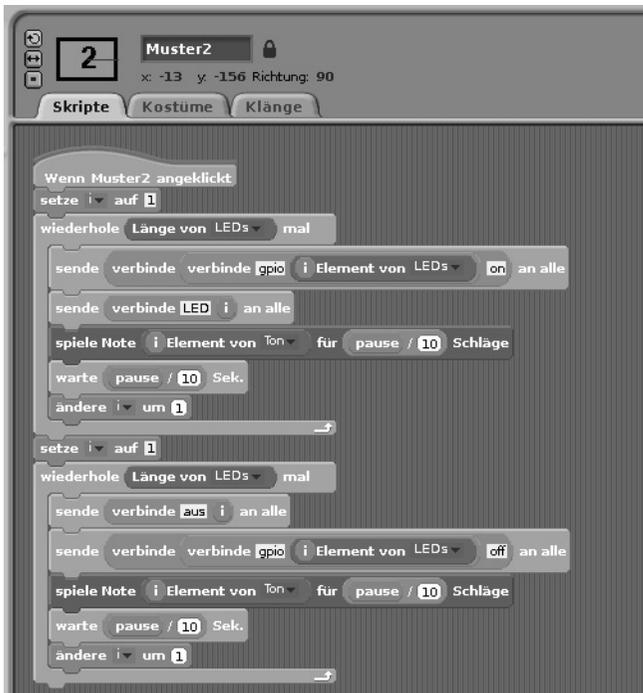
Der Button mit der Nummer 1 startet beim Anklicken die bekannte Lauflichtsequenz. Eine LED nach der anderen wird ein- und nach der eingestellten Pause wieder ausgeschaltet.



Die Programmblöcke für Button 1.

Nach dem Anklicken des Buttons läuft eine Schleife über alle Elemente der Liste **LEDs**. Als Erstes wird eine LED über ihren GPIO-Pin eingeschaltet. Ein **sende ... an alle**-Block sendet eine Nachricht mit der LED-Nummer, um das entsprechende LED-Objekt auf der Bühne einzuschalten.

Anschließend wird der der LED entsprechende Ton aus der Liste **Ton** abgespielt. Das Programm wartet so lange, bis die Töne und das Blinken synchron laufen. Danach sendet ein **sende ... an alle**-Block eine Nachricht mit der LED-Nummer, um das LED-Objekt wieder auszuschalten. Bei diesem Muster könnten auch einfach alle LED-Objekte ausgeschaltet werden. Da das aber bei den anderen Mustern nicht funktioniert, enthält die Nachricht zum Ausschalten diesmal auch die Nummer der LED. Zum Schluss wird die „echte“ LED über ihren GPIO-Pin ausgeschaltet und der Schleifenzähler um 1 erhöht.

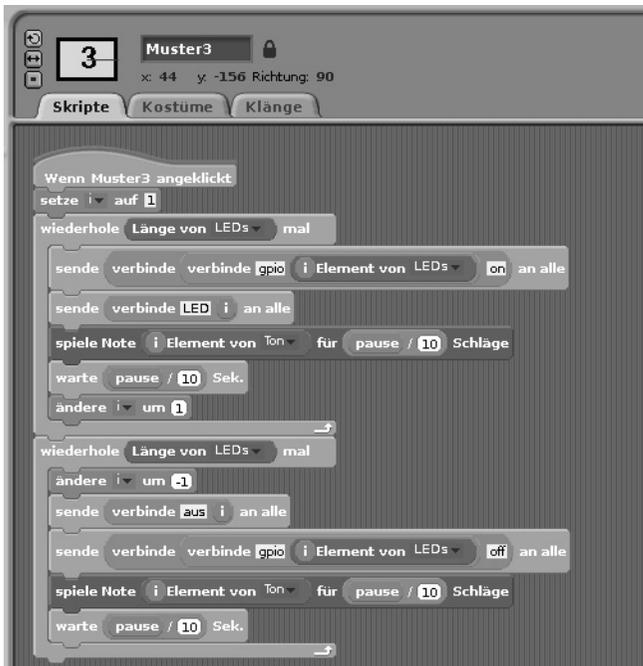


Die Programmblöcke für Button 2.

Der Button mit der Nummer 2 startet beim Anklicken eine LED nach der anderen und lässt sie eingeschaltet. Wenn alle fünf LEDs leuchten, werden sie in der gleichen Reihenfolge wieder ausgeschaltet.

Beim Anklicken des Buttons mit der Nummer 2 werden zwei Schleifen nacheinander abgearbeitet. Die erste Schleife schaltet eine LED über ihren GPIO-Pin ein, sendet die Nachricht zum Einschalten des LED-Objekts auf der Bühne, spielt den Ton ab und wartet.

Nachdem diese Schleife fünfmal durchgelaufen ist, wird der Schleifenzähler zurück auf 1 gesetzt. Danach werden die LED-Objekte und LEDs nacheinander wieder ausgeschaltet, und dazu werden noch einmal die Töne abgespielt.



Die Programmblöcke für Button 3.

Der Button mit der Nummer 3 startet beim Anklicken eine LED nach der anderen und lässt sie eingeschaltet. Wenn alle fünf LEDs leuchten, werden sie in umgekehrter Reihenfolge wieder ausgeschaltet.

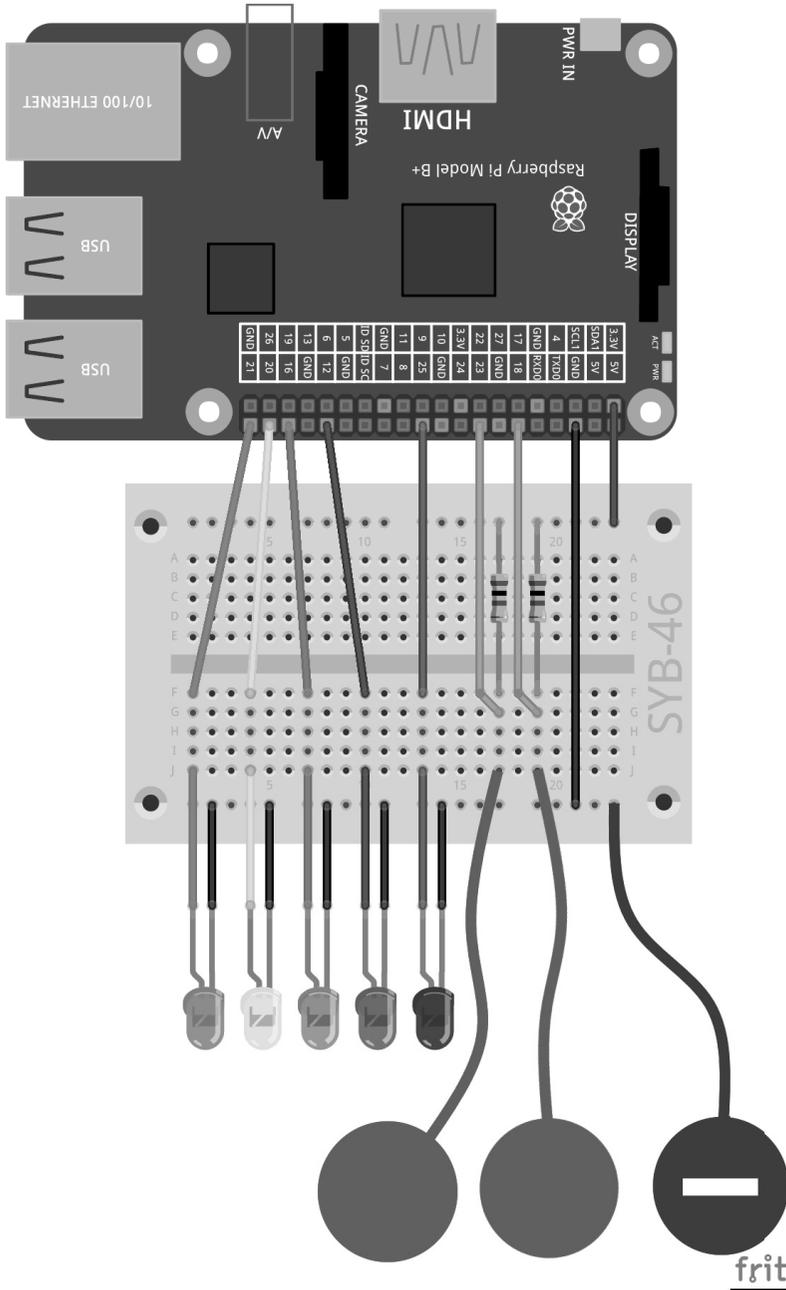
Die erste Schleife ist die gleiche wie die beim Button mit der Nummer 2. Nach den fünf Durchläufen bleibt der Schleifenzähler auf 5 stehen und wird dann bei jedem Durchlauf der zweiten Schleife wieder um 1 verringert. Dadurch werden die LEDs in umgekehrter Reihenfolge ausgeschaltet.

19. Tag

### 19. Tag

#### Heute im Adventskalender

• 1 Widerstand 20 MOhm (rot-schwarz-blau)



Zwei Knetkontakte steuern die LEDs.

#### Lauflicht oder Blinken

Das Experiment des 19. Tags enthält zwei Knetkontakte und einen Massekontakt. Berührt man den Knetkontakt an GPIO-Pin 23, läuft ein Lauflicht nach dem ausgewählten Muster. Berührt man den Knetkontakt an GPIO-Pin 18, leuchten alle LEDs einmal kurz auf.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED orange mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 LED lila mit Vorwiderstand; 2 Widerstände 20 MOhm; 9 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 10 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 3 Knetkontakte

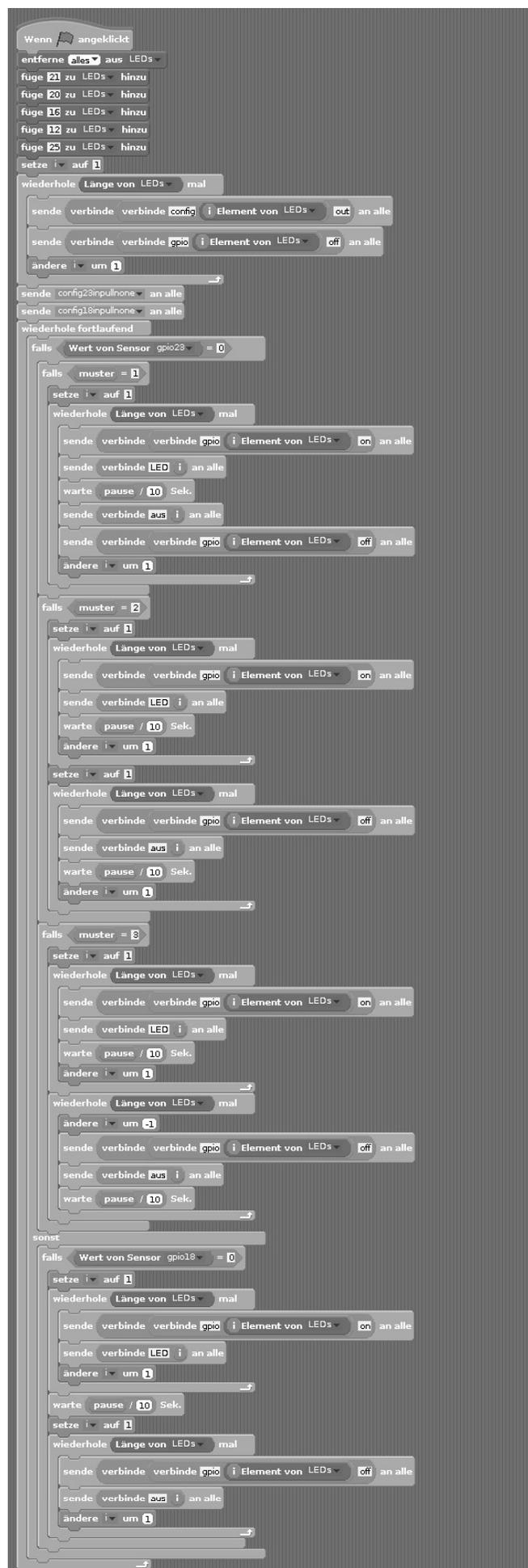
#### Das Programm

Da das Lauflicht im Programm `led19` anders als im vorherigen Programm nicht direkt durch Anklicken eines Buttons ausgelöst wird, sondern wartet, bis der Benutzer einen Knetkontakt berührt, folgt es einer anderen Programmlogik. Beim Klick auf einen Button sendet dieser eine Nachricht mit dem Namen des Musters.



Programmblock eines der drei Buttons.

Das Hauptprogramm auf der Bühne wartet auf die Berührung eines Knetkontakts und wertet dann das ausgewählte Muster aus. Für den zweiten Knetkontakt wird ein weiterer GPIO-Pin als Eingang ohne Pull-down-Widerstand initialisiert.



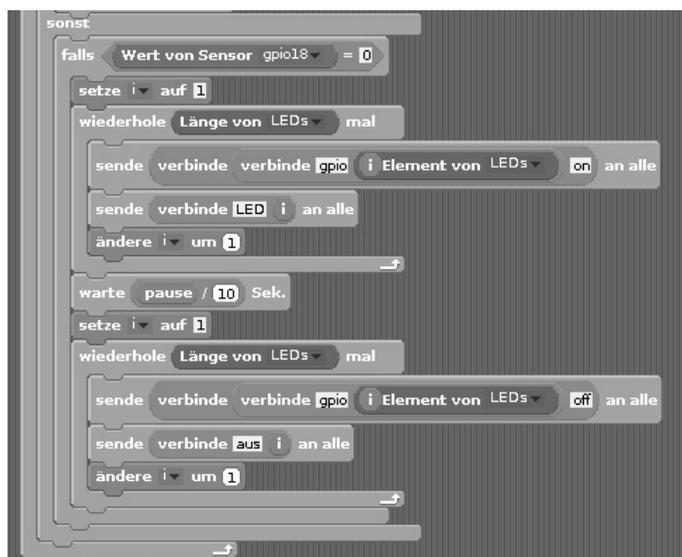
Das Hauptprogramm auf der Bühne.



Der Aufbau der Hauptschleife zur Übersicht.

Wird der Knetkontakt an GPIO-Pin 23 berührt, wird abhängig vom zuletzt angeklickten Button ein Lauffichtmuster abgespielt. Wie in den vorherigen Programmen leuchten auch die LED-Objekte auf der Bühne auf.

Im Fall **sonst** - wenn dieser Knetkontakt nicht berührt wird - prüft eine weitere Abfrage, ob der Knetkontakt an GPIO-Pin 18 berührt wird. In diesem Fall werden alle LEDs ein- und nach kurzer Pause wieder ausgeschaltet.



Wird der Knetkontakt an GPIO-Pin 18 berührt, werden alle LEDs ein- und nach kurzer Pause wieder ausgeschaltet.

20. Tag

## 20. Tag

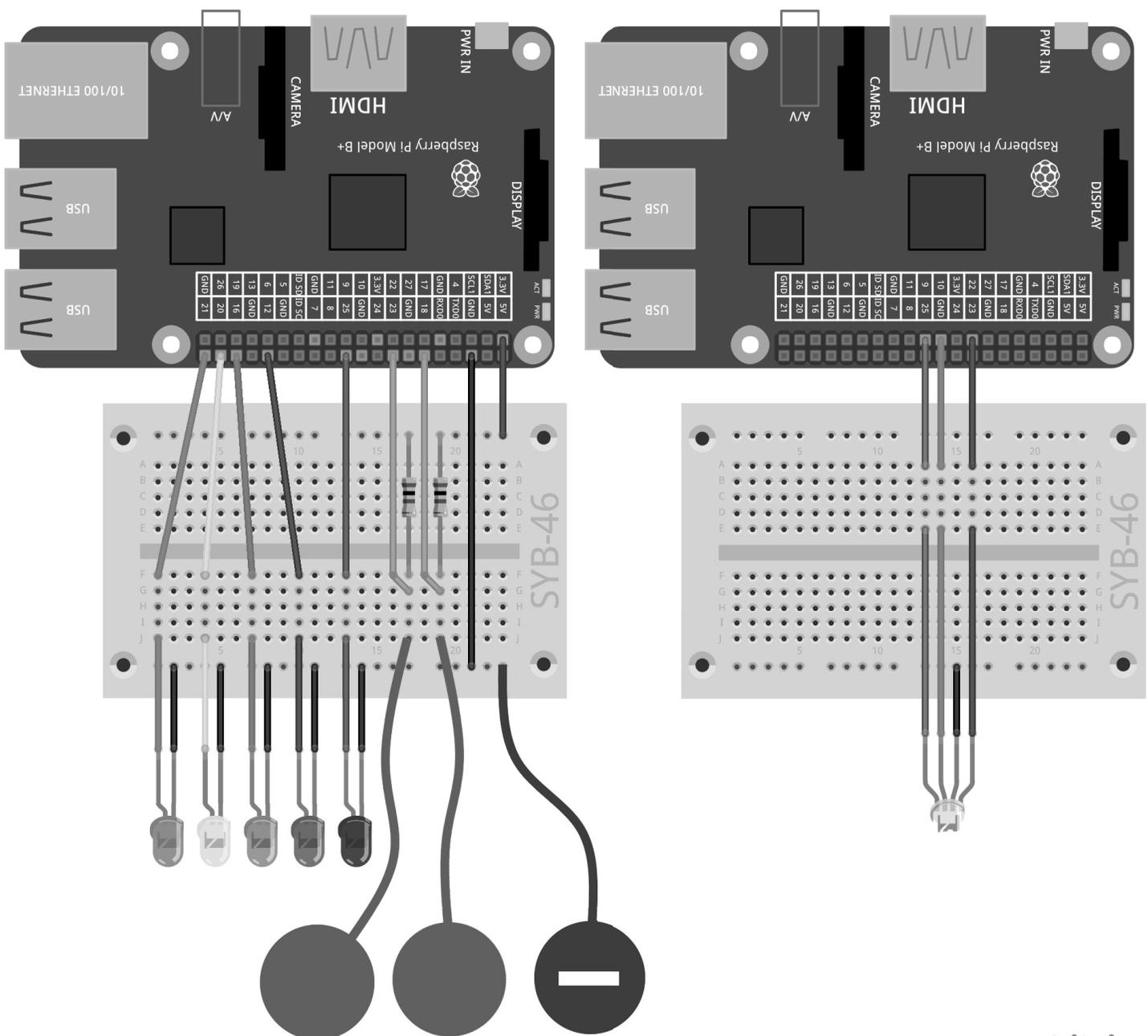
### Heute im Adventskalender

- 4 GPIO-Verbindungskabel (lang)

### RGB-Farbspiele mit Knetesensor steuern

Ähnlich wie das vorherige Programm zeigt das Programm des 20. Tags verschiedene Lauflichtmuster, wenn der Knetekontakt an GPIO-Pin 23 berührt wird. Wird der Knetekontakt an GPIO-Pin 18 berührt, leuchtet das LED-Objekt auf dem Stern der Scratch-Bühne auf. Außerdem leuchtet die RGB-LED in wechselnden Farben.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED orange mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 LED lila mit Vorwiderstand; 1 RGB-LED mit Vorwiderstand; 2 Widerstände 20 MOhm; 12 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 14 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 3 Knetekontakte



fritzing

Zwei Knetekontakte steuern das Lauflicht und eine RGB-LED.

Da es auf dem Steckbrett langsam eng wird, zeigen diese und die Abbildungen der folgenden Tage den Schaltungsaufbau in zwei Teilen. Die Kabel der RGB-LED, die in das Loch auf dem Stern über der Krippe gesteckt wird, würden in der Grafik andere Anschlusskabel verdecken.

## Das Programm

Das Programm `20rgbLed20` initialisiert am Anfang zusätzlich drei GPIO-Pins als PWM-Ausgänge für die RGB-LED.

Außerdem liegt auf der Bühne das bereits bekannte Unterprogramm zum Umrechnen von H-Werten in RGB-Farben. Wenn Sie das Scratch-Programm selbst zusammenbauen, können Sie diesen Programmblock aus einem früheren Programm importieren.

Die Programmblöcke für die RGB-LED sind dem Objekt **Stern** zugeordnet, da dieses sich bei Berührung des Knetekontakts an GPIO-Pin 18 ebenfalls verändern soll.

Eine Schleife erhöht den H-Wert schrittweise von 0 bis 360. Am Ende der Schleife beginnt die Endlosschleife von Neuem und startet die innere Schleife wieder bei 0, sodass sich ein kontinuierlicher Farbverlauf ergibt.

Erst wenn der Knetekontakt an GPIO-Pin 18 berührt wird, wird der zu diesem Zeitpunkt aktuelle H-Wert in RGB umgerechnet. Der Stern erhält das helle Kostüm, das eine eingeschaltete LED darstellt, und der errechnete RGB-Wert wird über PWM-Signale auf der RGB-LED dargestellt.

Solange der Knetekontakt nicht berührt wird, verwendet das Objekt **Stern** das graue Kostüm, und die RGB-LED bleibt ausgeschaltet.

Die Programmblöcke auf der Bühne.

21. Tag

## 21. Tag

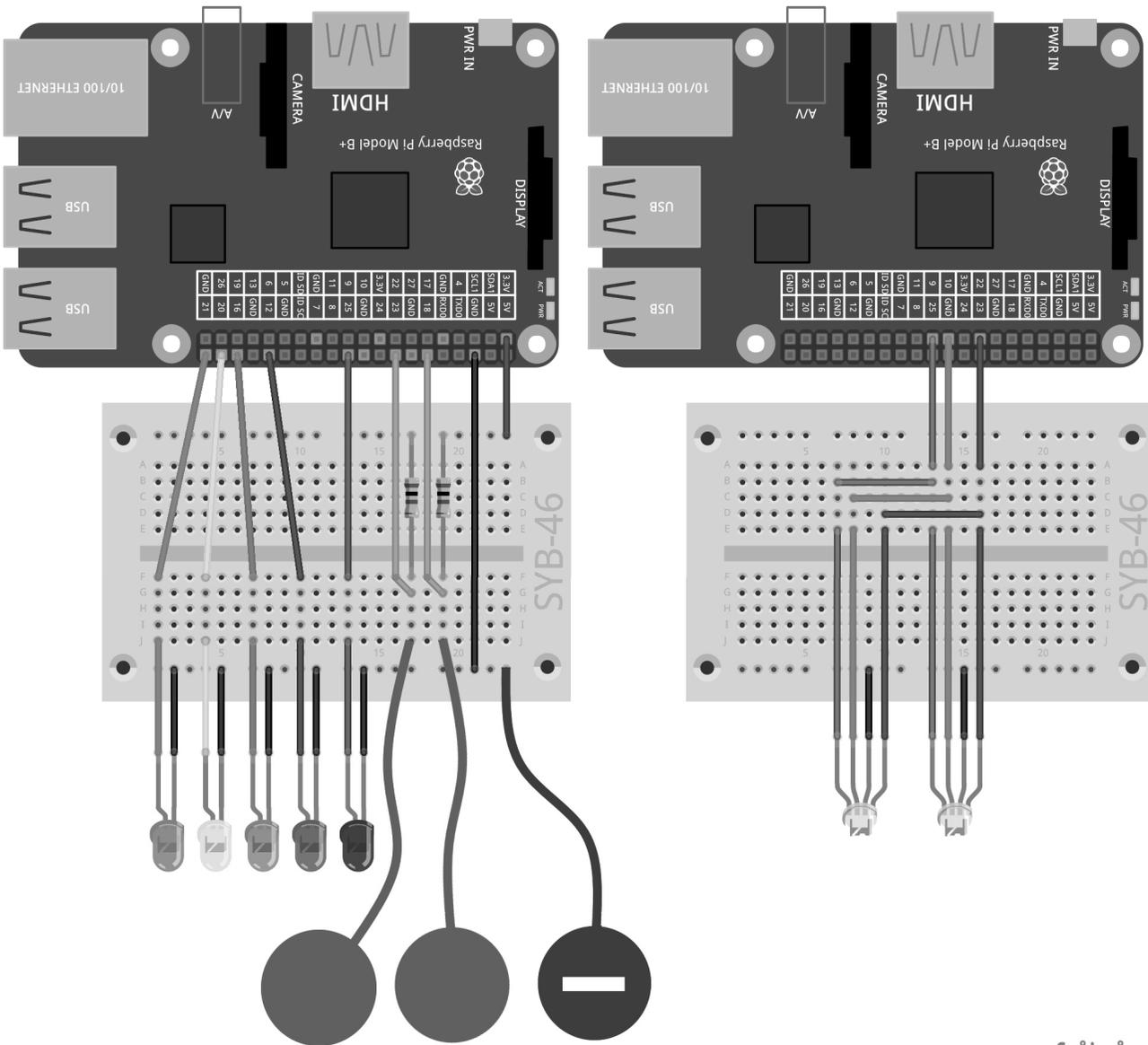
### Heute im Adventskalender

- 4 GPIO-Verbindungskabel (lang)

### Farbspiele auf dem Bildschirm

Das Experiment des 21. Tags ähnelt dem des vorherigen Tags. Allerdings zeigen zwei RGB-LEDs den gleichen Farbverlauf, und ein farbiger Punkt auf dem Stern zeigt die gleichen Farben wie die RGB-LEDs. Die Steuerung der Lauflichtmuster funktioniert wie in den vorherigen Programmen.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED orange mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 LED lila mit Vorwiderstand; 2 RGB-LEDs mit Vorwiderstand; 2 Widerstände 20 MOhm; 12 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 18 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 3 Drahtbrücken; 3 Knetkontakte



fritzing

Zwei Knetkontakte steuern das Lauflicht und zwei parallel geschaltete RGB-LEDs.

Die beiden RGB-LEDs sind über Drahtbrücken an den gleichen GPIO-Pins angeschlossen und zeigen deshalb immer die gleiche Farbe.

### Das Programm

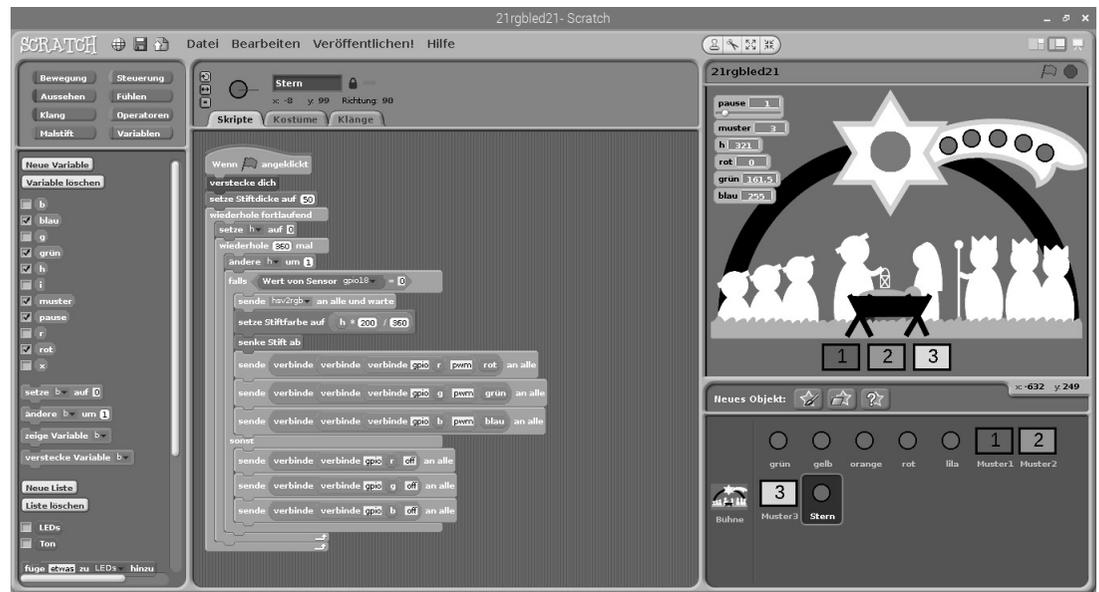
Der Farbwechsel auf dem Stern wird wegen der großen Zahl verschiedener Farben nicht über Kostüme, sondern über den sogenannten Malstift in Scratch dargestellt, mit dem Objekte an ihrer aktuellen Posi-

tion Spuren auf der Bühne hinterlassen können.

Die Programmblöcke des Objekts **Stern** verändern wieder zyklisch die Farbe und fragen den Knetekontakt an GPIO-Pin 18 ab.

Als Erstes wird das Objekt **Stern** versteckt, damit es seine eigenen Malspuren nicht verdeckt. Ein Objekt, das mit dem Block **verstecke dich** von der Blockpalette **Aussehen** versteckt wird, ist auf der Bühne nicht mehr zu sehen, kann sich aber weiter bewegen, Programmblöcke ausführen und auch Malspuren hinterlassen.

Ein Block **setze Stiftdicke auf 50** von der Blockpalette **Malstift** setzt die Dicke des Malstifts so, dass er einen kräftigen Punkt hinterlässt.



Die Programmblöcke für den Stern.

Immer wenn der Knetekontakt an GPIO-Pin 18 berührt wird, wird zusätzlich zum Leuchten der RGB-LED ein farbiger Punkt gemalt. Dieser befindet sich genau an der Position des versteckten Objekts **Stern**.

Dazu wird zuerst die neue Stifffarbe errechnet. Scratch verwendet 200 verschiedene Stifffarben entlang des Farbspektrums, ähnlich den H-Werten im HSV-System, aber nicht im Bereich **0...360**, sondern nur im Bereich **0...200**. Zur Umrechnung wird der aktuelle H-Wert mit 200 multipliziert und durch 360 geteilt.

Anschließend wird der Malstift gesenkt. Ähnlich wie bei einem Stiftplotter hinterlässt das Objekt bei jeder Bewegung mit gesenktem Malstift Spuren, bei angehobenem Malstift nicht. Auch ohne Bewegung setzt das Senken des Malstifts jedes Mal einen Punkt in der aktuellen Stifffarbe.

22. Tag

## 22. Tag

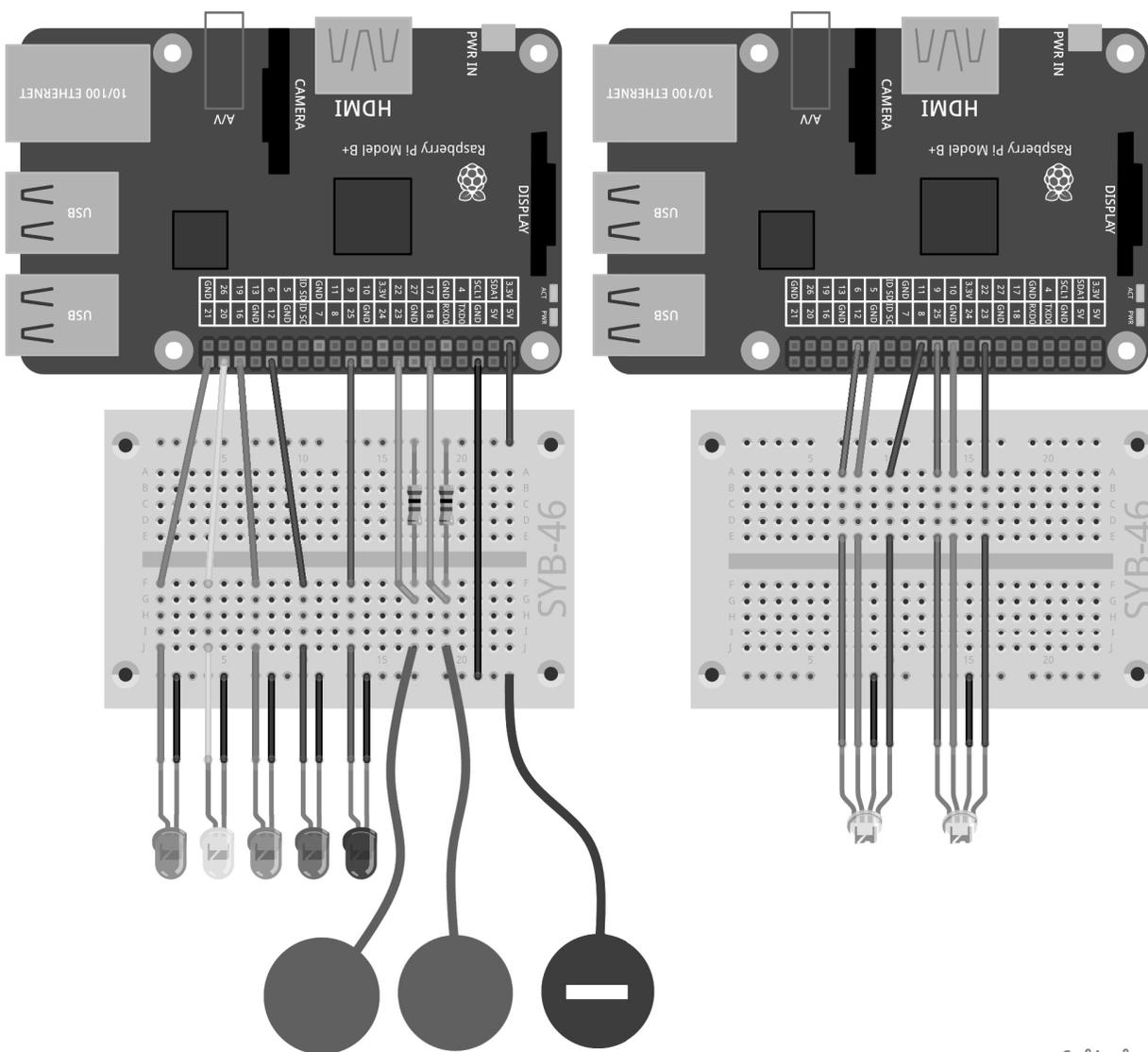
### Heute im Adventskalender

- 6 GPIO-Verbindungskabel (kurz)

### Zwei RGB-LEDs unabhängig steuern

Das Experiment des 22. Tags steuert die beiden RGB-LEDs unabhängig voneinander mit den beiden Knetkontakten. Dazu werden drei zusätzliche GPIO-Pins benötigt. Die Lauflichtmuster werden ausschließlich über die Buttons gesteuert. In diesem Programm sind zwei weitere Buttons hinzugekommen. Das Muster 4 lässt alle LEDs kurz aufblinken, der Button **aus** schaltet alle LEDs aus.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED orange mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 LED lila mit Vorwiderstand; 2 RGB-LEDs mit Vorwiderstand; 2 Widerstände 20 MOhm; 15 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 18 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 3 Knetkontakte



Zwei Knetkontakte steuern das Lauflicht und zwei unabhängig voneinander schaltbare RGB-LEDs.



### Das Programm

Das Hauptprogramm auf der Bühne initialisiert, wie bereits bekannt, eine Liste der fünf LEDs für das Lauflicht sowie die drei PWM-Ausgänge **r**, **g** und **b** für die erste RGB-LED. Neu hinzu kommen die drei weiteren PWM-Ausgänge **r2**, **g2** und **b2** für die zweite RGB-LED.

Eine Endlosschleife stellt anhand der Variablen **muster**, die durch die Buttons gesetzt wird, eines von fünf Blinkmustern auf den fünf LEDs dar. Die Muster 0 und 4 sind neu:

Das Muster 0 schaltet ohne zusätzliche Wartezeiten alle fünf LEDs aus.

Das Muster 4 lässt alle LEDs nahezu gleichzeitig blinken. Da die GPIO-Pins nacheinander eingeschaltet und gleichzeitig noch andere Scratch-Blöcke abgearbeitet werden, sind kurze Verzögerungen zwischen dem Aufblinken der einzelnen LEDs zu sehen.

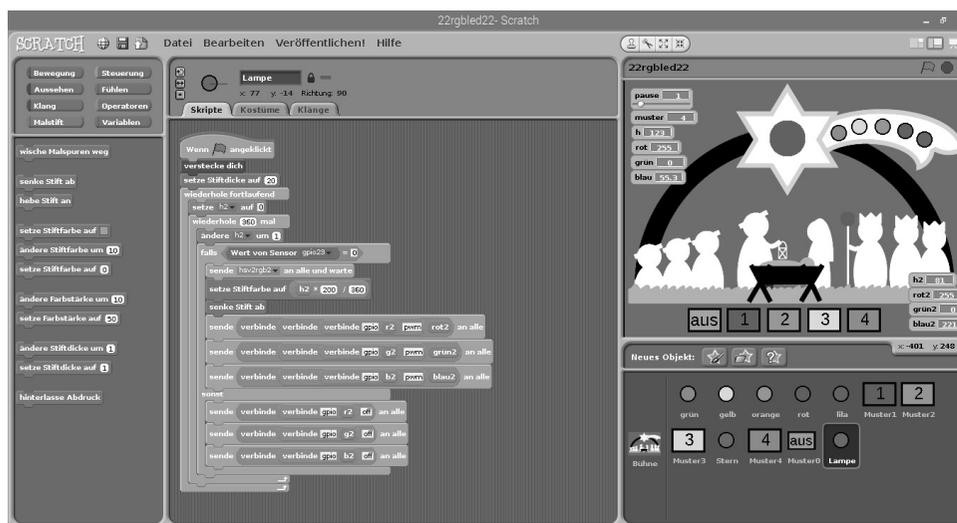
Außerdem befinden sich auf der Bühne zwei Programmblöcke, die abgearbeitet werden, wenn das Signal **hsv2rgb** oder **hsv2rgb2** empfangen wird. Die Arbeitsweise ist bereits bekannt. Der zweite Programmblock könnte auch direkt dem entsprechenden LED-Objekt zugeordnet werden, lässt sich aber durch Duplizieren und Austauschen der Variablen leicht erstellen und liegt deshalb auch auf der Bühne.



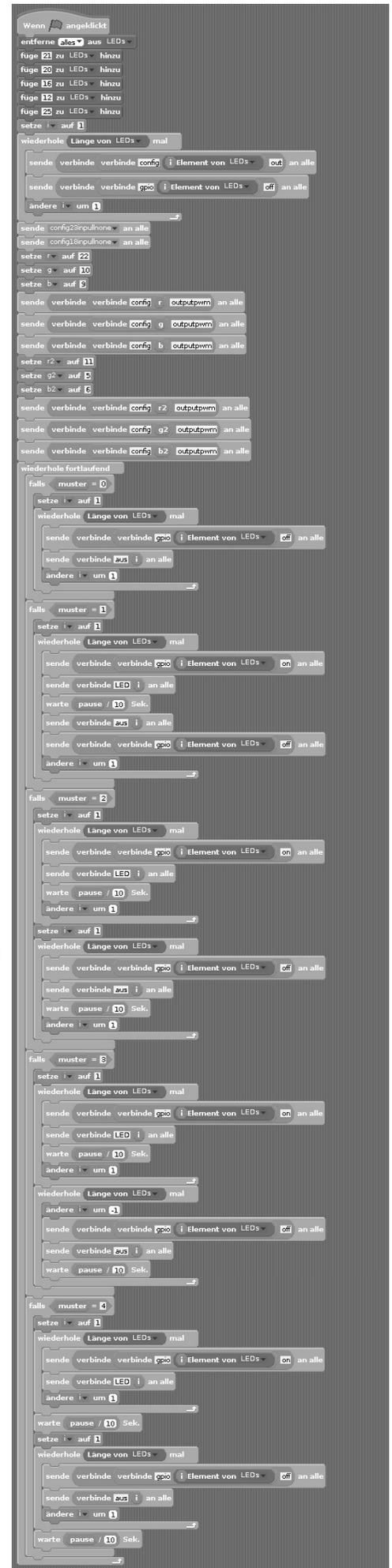
Diese Blöcke ermitteln die RGB-Farben für die beiden RGB-LEDs aus H-Werten.

Das neue Objekt **Lampe** am Stab des ersten Königs zeigt die Farbe der zweiten RGB-LED. Nach dem bereits bekannten Prinzip wird der Farbwert **h2** schrittweise von 0 an in 360 Schritten erhöht. Wird der Knetkontakt an GPIO-Pin 23 berührt, wird die Nachricht **hsv2rgb2** gesendet, und damit werden die RGB-Farbwerte **rot2**, **grün2** und **blau2** errechnet. Anschließend werden eine Stifffarbe ermittelt, ein farbiger Punkt gezeichnet und die GPIO-Pins **r2**, **g2** und **b2** auf die errechneten PWM-Werte gesetzt.

Solange der Knetkontakt nicht berührt wird, bleibt die RGB-LED aus. Das Objekt **Lampe** zeigt die zuletzt eingestellte Farbe.



Die Programmblöcke für das neue Objekt **Lampe**.



Das Hauptprogramm auf der Bühne.

23. Tag

### 23. Tag

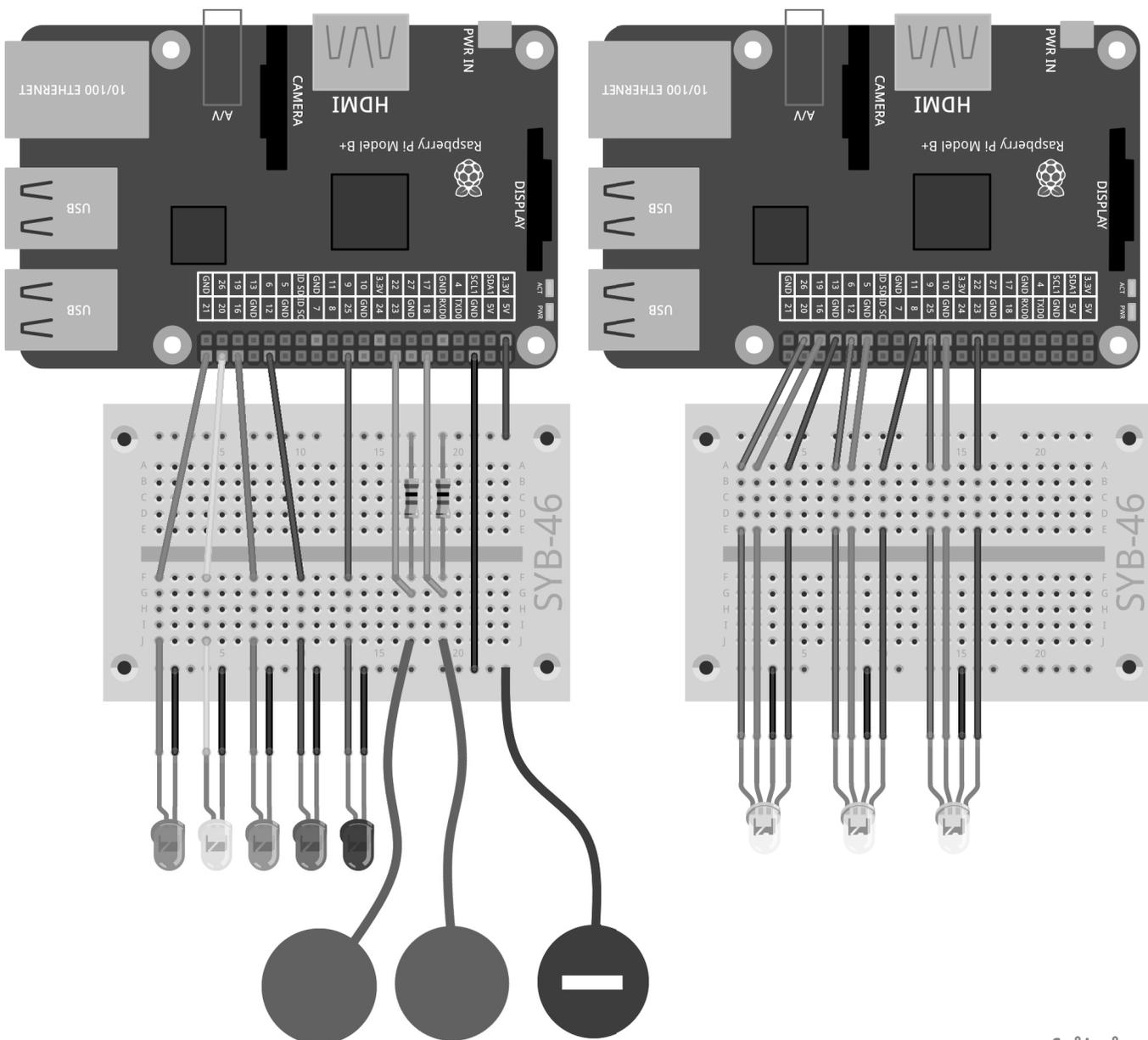
#### Heute im Adventskalender

• 1 RGB-LED mit Vorwiderstand

#### Bunt beleuchtete Weihnachtskrippe

Eine dritte RGB-LED in der Laterne, die Josef auf der Weihnachtskrippe hält, leuchtet in verschiedenen Grundfarben, je nachdem, welche der beiden Knetkontakte berührt werden.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED orange mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 LED lila mit Vorwiderstand; 3 RGB-LEDs mit Vorwiderstand; 2 Widerstände 20 MOhm; 18 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 22 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 3 Knetkontakte

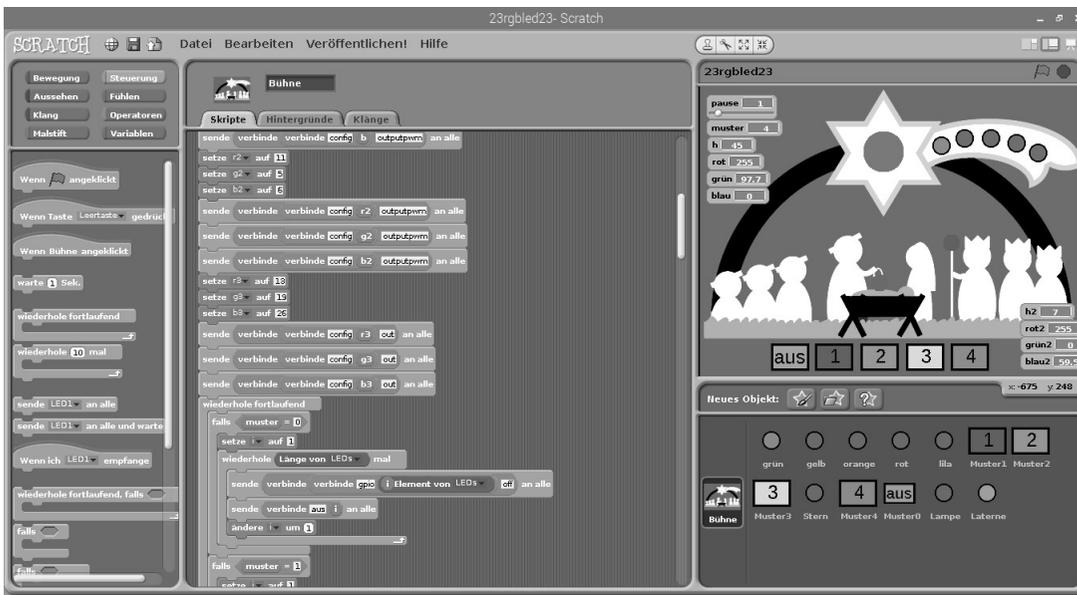


fritzing

Zwei Knetkontakte steuern das Lauflicht und drei unabhängig voneinander schaltbare RGB-LEDs.

#### Das Programm

Das Hauptprogramm auf der Bühne initialisiert wieder die GPIO-Pins für die LEDs und RGB-LEDs. Neu sind hier nur die als **r3**, **g3** und **b3** bezeichneten GPIO-Pins. Diese werden als einfache Ausgänge, nicht als PWM-Ausgänge initialisiert.



Zusätzliche GPIO-Ausgänge und ein neues Objekt auf der Bühne.

Das neue Objekt **Laterne** wird von Anfang an versteckt und hinterlässt nur beim Farbwechsel einen Stiftabdruck. Hier werden beide Knetkontakte abgefragt. Wird der Sensor an Pin 23 berührt, wird die Stiftfarbe auf Rot gesetzt. Dazu wird der Block **setze Stiftfarbe auf ...** mit dem Farbfeld verwendet. Tippen Sie auf dieses Farbfeld, erscheint ein Pipettensymbol, mit dem Sie eine beliebige Farbe vom Bildschirm aufnehmen können, die der Block später als Stiftfarbe setzen soll. Außerdem wird beim Berühren dieses Knetkontakts der GPIO-Pin **r3** für die rote Farbe der dritten RGB-LED eingeschaltet. Beim Loslassen wird dieser Pin wieder ausgeschaltet.

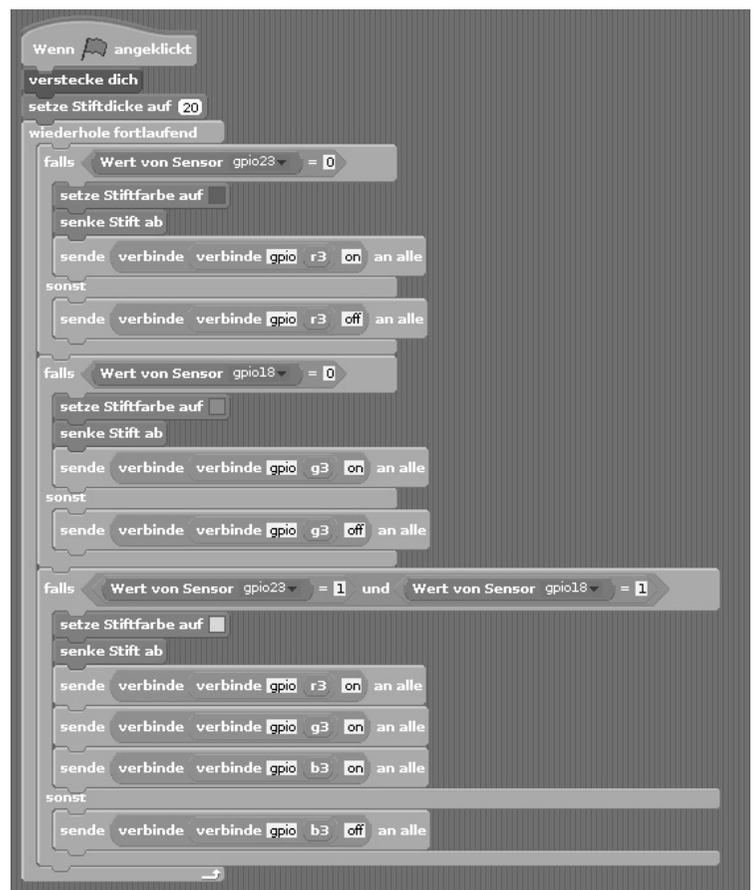
Auf die gleiche Weise wird beim Berühren des Knetkontakts an GPIO-Pin 18 die Stiftfarbe auf Grün gesetzt und die grüne Farbkomponente der RGB-LED eingeschaltet. Werden beide Knetkontakte gleichzeitig berührt, sind die Pins **r3** und **g3** eingeschaltet, die RGB-LED leuchtet in einer gelben Mischfarbe. Das Objekt **Laterne** zeigt die zuletzt eingeschaltete Farbe, da hier keine Farben gemischt werden.

Wird keiner der beiden Knetkontakte berührt - stehen also beide Sensorwerte auf 1 -, leuchtet die RGB-LED in hellem Weiß. Dazu werden die drei GPIO-Pins **r3**, **g3** und **b3** eingeschaltet. Das Objekt **Laterne** zeigt Gelb, da Weiß auf der Hintergrundgrafik schwer zu erkennen ist. Trifft die Bedingung nicht mehr zu, weil mindestens einer der Knetkontakte berührt wird, wird der GPIO-Pin **b3** ausgeschaltet, damit beim Berühren eines oder beider Knetkontakte wieder die Grundfarben Rot, Grün oder ein gemischtes Gelb angezeigt werden - ohne zugemischtes Blau.

Bis jetzt enthält das Programm noch keine Möglichkeit, die RGB-LED ganz auszuschalten. Ein zusätzlicher Programmblock erledigt das beim Drücken der Leertaste.

Der Block **Wenn Taste ... gedrückt** reagiert auf einen Tastendruck auf der Tastatur. Über das Listenfeld wählen Sie die gewünschte Taste aus.

In diesem Fall wird ein grauer Punkt am Objekt **Laterne** gemalt, und alle drei Farbkomponenten der RGB-LED werden ausgeschaltet.



Die Programmblöcke für das Objekt **Laterne**.



Beim Drücken der Leertaste wird die RGB-LED abgeschaltet.

24. Tag

## 24. Tag

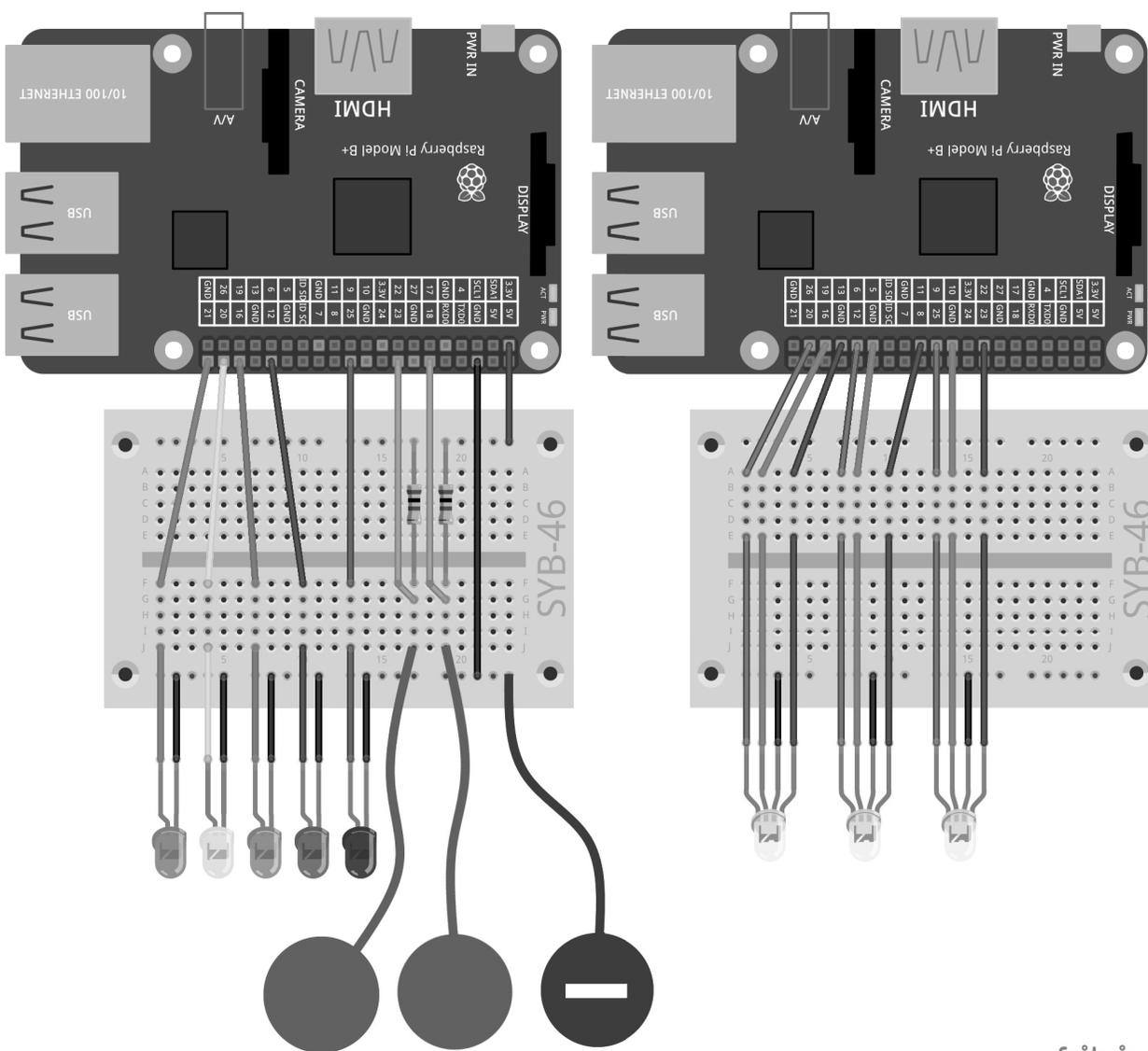
### Heute im Adventskalender

• 1 Downloadcode

### Weihnachtslieder auf dem Raspberry Pi

Am letzten Tag wird das Programm nochmals um zwei interessante Funktionen erweitert. Beim Anklicken eines der fünf LED-Objekte auf der Bühne werden fünf verschiedene Weihnachtslieder abgespielt. Die LED in der Laterne lässt sich über einen regenbogenfarbenen Balken auf der Bühne interaktiv einstellen.

**Bauteile:** 1 Steckbrett SYB-46; 1 LED grün mit Vorwiderstand; 1 LED gelb mit Vorwiderstand; 1 LED orange mit Vorwiderstand; 1 LED rot mit Vorwiderstand; 1 LED lila mit Vorwiderstand; 3 RGB-LEDs mit Vorwiderstand; 2 Widerstände 20 MOhm; 18 GPIO-Verbindungskabel kurz (Raspberry Pi - Steckbrett); 22 GPIO-Verbindungskabel lang (Steckbrett - LEDs); 3 Knetkontakte



Zwei Knetkontakte steuern das Lauflicht und drei unabhängig voneinander schaltbare RGB-LEDs.

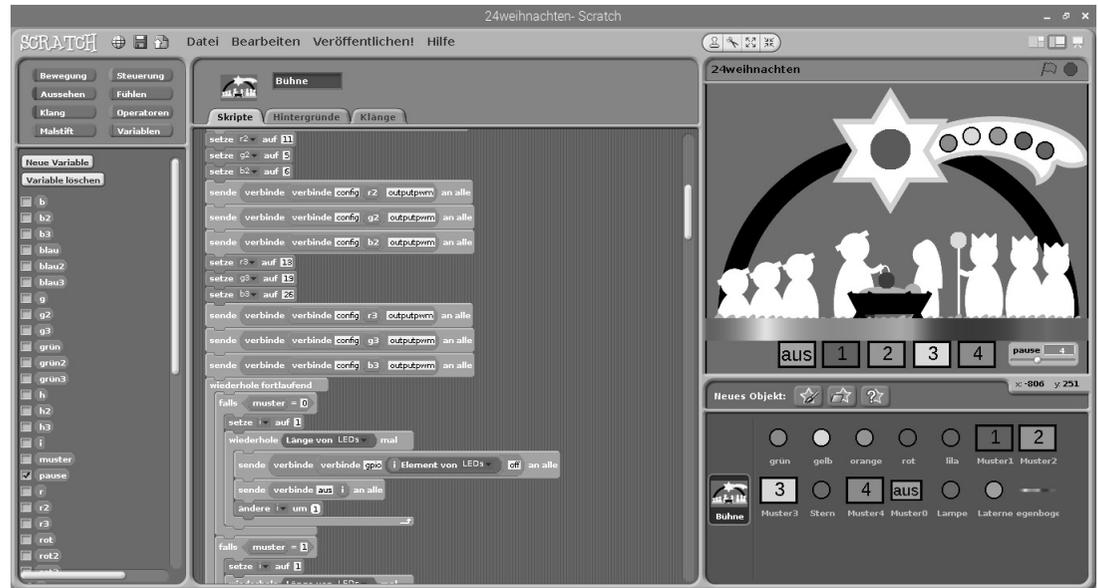
fritzing

### Das Programm

Die Programmblöcke auf der Bühne sind im Wesentlichen bekannt. Die dritte RGB-LED an den GPIO-Pins **r3**, **g3** und **b3** werden diesmal wie die anderen als PWM-Ausgänge eingerichtet. Ein neuer Programmblock auf der Bühne reagiert auf die Nachricht **hsv2rgb3** und errechnet die RGB-Werte für die dritte RGB-LED. Diesen Block können Sie wieder einfach duplizieren und anschließend die Variablen austauschen.

Für das Objekt **Regenbogen** importieren Sie die Grafik `regenbogen.png` aus den Downloads. Beim Klick mit der Maus auf dieses Objekt soll die RGB-LED in der entsprechenden Farbe leuchten.

Da man das Objekt beim Anklicken leicht versehentlich verschiebt, wird es sofort nach dem Anklicken wieder an seine Grundposition verschoben. Anschließend wird der H-Wert der gewünschten Farbe anhand der x-Position der Maus ermittelt. Die Scratch-Bühne ist 480 Einheiten breit und hat x-Koordinaten von **-240...240**. Der Block **Maus x-Position** liest die aktuelle x-Koordinate der Maus aus. Aus diesem Wert, der zwischen -240 und +240 liegen kann, wird ein Wert **h3** zwischen 0 und 360 errechnet. **sende hsv2rgb3 an alle und warte** startet die Berechnung der passenden RGB-Farbwerte. Diese werden anschließend als PWM-Signale auf der RGB-LED dargestellt.



Neue Programmblöcke auf der Bühne.

Das Objekt **Laterne** reagiert ebenfalls auf die Nachricht **hsv2rgb3** und malt einen farbigen Punkt in der entsprechenden Farbe. Der Wert **h3** im Bereich zwischen 0 und 360 wird dazu in einen Farbwert im Bereich zwischen 0 und 200 umgerechnet. Beim Drücken der Leertaste wird die RGB-LED ausgeschaltet.

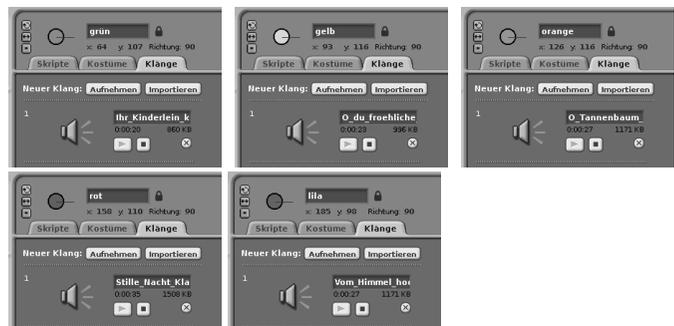
Scratch kann auf dem Raspberry Pi MP3-Dateien abspielen, die in das Programm importiert wurden. Importieren Sie dazu bei den Objekten **grün, gelb, orange, rot** und **lila** auf der Registerkarte **Klänge** je eins der Weihnachtslieder, die Sie über den Downloadcode im heutigen Türchen des Adventskalenders erhalten.



Die Blöcke für das Objekt **Regenbogen**.

Beim Klick auf ein solches Objekt soll das zugehörige Weihnachtslied abgespielt werden.

Bauen Sie dazu in jedes Objekt zusätzlich zu den Blöcken für den Kostümwechsel noch einen Block **Wenn ... angeklickt** ein. Zuerst sollen eventuell noch laufende Lieder gestoppt werden, damit nicht mehrere Lieder gleichzeitig erklingen. Dazu gibt es auf der Blockpalette **Klang** einen Block **stoppe alle Klänge**. Anschließend soll ein Block **spiele Klang ... ganz** das zu diesem Objekt importierte Lied abspielen.



Weihnachtslieder für die LED-Objekte importieren.

### Übersicht

Zur Übersicht noch mal alle Funktionen des Programms:

- Die vier Buttons mit den Ziffern lassen die LEDs sowie die LED-Objekte auf dem Schweif des Sterns in verschiedenen Mustern blinken.
- Der Regler **pause** regelt die Blinkgeschwindigkeit.
- Der Button **aus** schaltet alle LEDs und LED-Objekte aus.
- Der erste Knetkontakt steuert die Farbe der RGB-LED und des farbigen Punkts auf dem Stern.
- Der zweite Knetkontakt steuert die Farbe der RGB-LED und des farbigen Punkts auf der Lampe des Königs.
- Ein Klick auf den regenbogenfarbenen Balken steuert die Farbe der RGB-LED und des farbigen Punkts auf der Laterne von Josef an der Krippe.
- Ein Drücken der Leertaste schaltet diese RGB-LED aus.
- Ein Klick auf eins der fünf LED-Objekte spielt ein Weihnachtslied ab.

Frohe Weihnachten!



Beim Anklicken der Objekte erklingen Weihnachtslieder.

### Vorsichtsmaßnahmen

Auf keinen Fall irgendwelche GPIO-Pins miteinander verbinden und abwarten, was passiert. Nicht alle GPIO-Pins lassen sich frei programmieren. Einige sind für die Stromversorgung und andere Zwecke fest eingerichtet. Einige GPIO-Pins sind direkt mit Anschlüssen des Prozessors verbunden, ein Kurzschluss kann den Raspberry Pi komplett zerstören. Verbindet man über einen Schalter oder eine LED zwei Pins miteinander, muss immer ein Schutzwiderstand dazwischengeschaltet werden. Für Logiksignale immer Pin 1 verwenden, der +3,3 V liefert und bis 50 mA belastet werden kann. Pin 6 ist die Masseleitung für Logiksignale. Pin 2 liefert +5 V zur Stromversorgung externer Hardware. Hier kann so viel Strom entnommen werden, wie das USB-Netzteil des Raspberry Pi liefert. Dieser Pin darf aber nicht mit einem GPIO-Eingang verbunden werden.

Liebe Kunden!



Dieses Produkt wurde in Übereinstimmung mit den geltenden europäischen Richtlinien hergestellt und trägt daher 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.

### Warnung! 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.

© 2018 Franzis Verlag GmbH, Richard-Reitzner-Allee 2, 85540 Haar

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.

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

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.