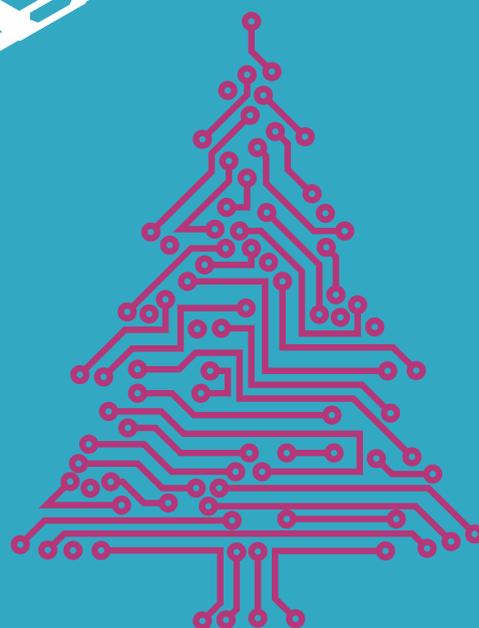
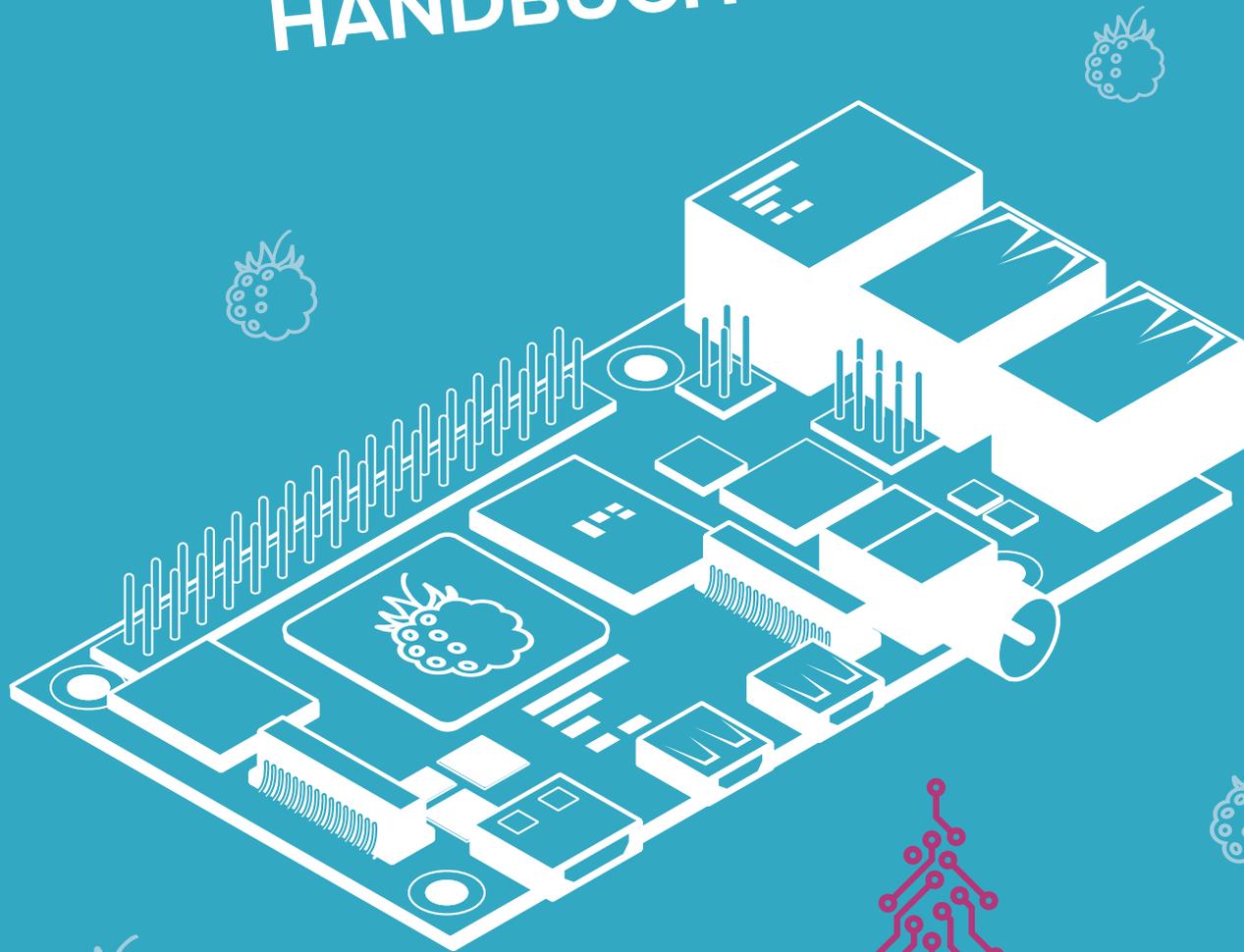


ADVENTSKALENDER FÜR

# RASPBERRY PI

HANDBUCH



FRANZIS

## IMPRESSUM

© 2021 Franzis Verlag GmbH, Richard-Reitzner-Allee 2, D-85540 Haar, Germany  
Änderungen, Innovationen und Druckfehler vorbehalten

www.franzis.de

2021/1

Autor: Christian Immler  
Produktmanagement: Tobias Schärfl  
Satz: PC-DTP-Satz und Informations GmbH  
Copy Editor: Claudia Fliedner  
Coverdesign: Thomas Preischl  
GTIN: 4019631551030

Nicht für Kinder unter 14 Jahren geeignet.

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.

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.

## 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.

## KURZSCHLÜSSE VERMEIDEN

Eine direkte Verbindung zwischen Minus- und Pluspol muss unbedingt vermieden werden, weil Drähte und Batterien heiß werden können und weil die Batterien sich dann schnell verbrauchen. Im Extremfall können Drähte glühend heiß werden, und die Batterie kann explodieren. Es besteht Brand- und Verletzungsgefahr. Verwenden Sie nach Möglichkeit nur normale Zink-Kohle-Batterien (6F20), die einen geringeren Kurzschlussstrom liefern und deshalb weniger gefährlich sind als Alkali-batterien (6RL61). Verwenden Sie keinesfalls Akkus.

**BENÖTIGEN SIE HILFE ODER HABEN SIE FRAGEN ZU DIESEM ADVENTSKALENDER?**

Unter **<https://www.franzis.de/faq-zu-den-adventskalendern>** finden Sie Antworten auf die häufigsten Fragen und Kontaktmöglichkeiten zu unserem Support-Team.

# Raspberry-Pi-Adventskalender 2021

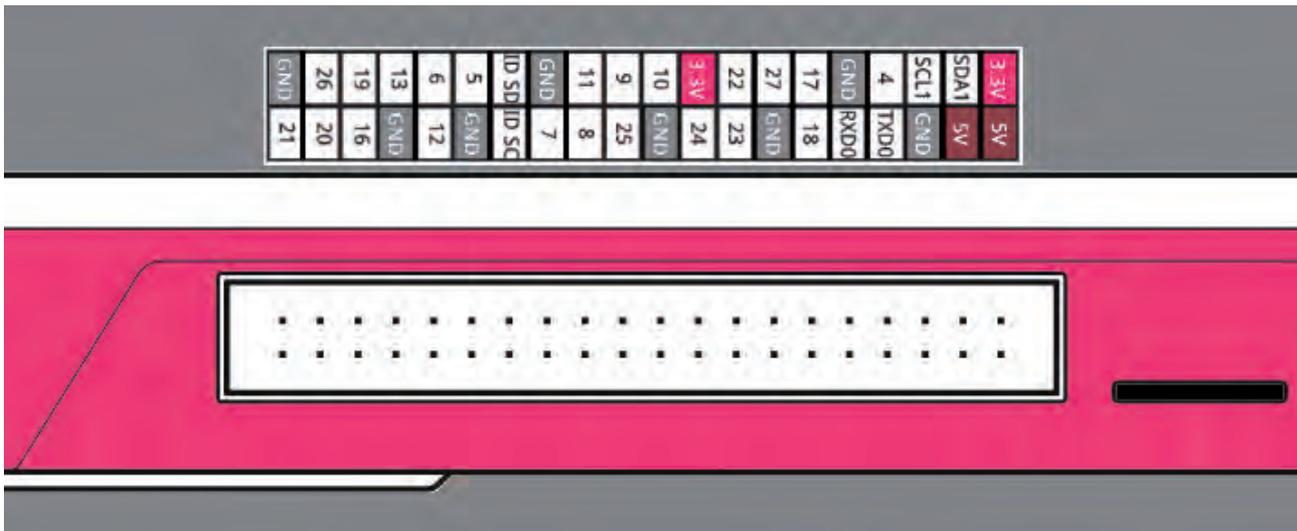
Dieser Adventskalender enthält für jeden Tag ein Hardwareexperiment mit dem Raspberry Pi. Nach und nach wird eine Weihnachtskrippe aufgebaut, die am 24. Dezember in voller Pracht erstrahlt. Die Experimente werden mit Scratch 3 programmiert. Diese Programmiersprache ist auf dem Raspberry Pi vorinstalliert. Alle Experimente funktionieren mit dem Raspberry Pi 4 und dem Raspberry Pi 400.

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 erforderlichen Schnittstellen. Außerdem ist das Windows-Betriebssystem denkbar ungeeignet dafür, mit Elektronik zu kommunizieren.

Der Raspberry Pi ist – obwohl es auf den ersten Blick gar nicht so aussieht – 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.







Belegung der GPIO-Schnittstelle am Raspberry Pi 400.

## 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. Dort 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.

## Raspberry Pi vorbereiten

Der Raspberry Pi 4 ist nur eine einfache Platine in der Größe einer Kreditkarte, der Raspberry Pi 400 dagegen ist elektronisch fast gleich, aber in eine Tastatur eingebaut und somit ein eigenständiger Computer für den Schreibtisch. Um den Raspberry Pi in Betrieb zu nehmen, braucht man Folgendes:

- USB-Tastatur (nicht beim Raspberry Pi 400) und Maus
- HDMI-Kabel für Monitor mit Micro-HDMI-Stecker oder -Adapter
- Netzkabel oder WLAN
- MicroSD-Karte mit Betriebssystem Raspberry Pi OS
- USB-Typ-C-Netzteil

Das Netzteil muss als Letztes angeschlossen werden. Damit schaltet sich der Raspberry Pi automatisch ein. Am Raspberry Pi 4 gibt es keinen eigenen Ein-/Ausmacher. Der Raspberry Pi 400 kann über die Tastenkombination **[Fn] + [F10]** ein- und ausgeschaltet werden.

## Betriebssysteminstallation in Kürze

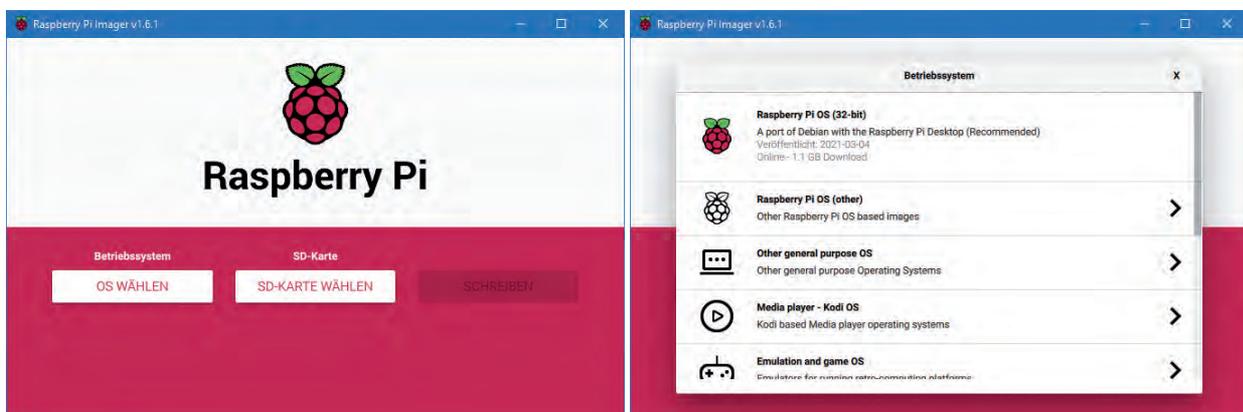
Für alle, die ihren Raspberry Pi noch nicht mit der aktuellen Version des Betriebssystems Raspberry Pi OS ausgerüstet haben, folgt hier die Systeminstallation in wenigen Schritten:

# Raspberry-Pi-Adventskalender

## Betriebssystem installieren

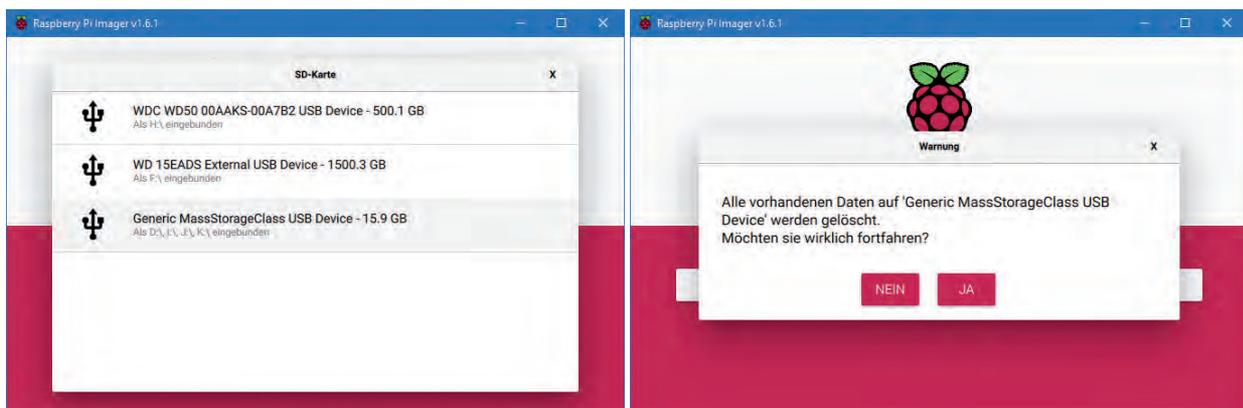
Der neue **Raspberry Pi Imager** erledigt das Formatieren, das Herunterladen des Betriebssystems und die Installation mit wenigen Klicks.

- 1 Laden Sie den **Raspberry Pi Imager** bei [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads) herunter und installieren Sie ihn auf dem Windows-PC.
- 2 Stecken Sie die Speicherkarte in den Kartenleser und starten Sie den **Raspberry Pi Imager**. Dazu sind Administratorrechte auf dem PC nötig. Bestätigen Sie die Anfrage der Benutzerkontensteuerung.
- 3 Klicken Sie auf *OS wählen*. Es erscheint eine Liste mit verschiedenen Betriebssystemen zur Auswahl. Wählen Sie unter *Raspberry Pi OS (other)* die Version *Raspberry Pi OS Full (32-bit)*, in der wesentlich mehr Programme standardmäßig vorinstalliert sind, unter anderem auch die Programmiersprache Scratch 3, die wir in diesem Adventskalender verwenden.



Raspberry Pi Imager starten und Betriebssystem auswählen.

- 4 Wählen Sie im nächsten Schritt die Speicherkarte aus, auf der das Raspberry Pi OS installiert werden soll. Achten Sie darauf, wirklich die Speicherkarte auszuwählen und nicht versehentlich eine externe USB-Festplatte. Auch diese werden angezeigt. Auf dem ausgewählten Laufwerk vorhandene Daten werden überschrieben. In den meisten Fällen erkennt man die Speicherkarte schon an der Größe. Festplatten sind wesentlich größer.



Speicherkarte im Raspberry Pi Imager auswählen und *Schreiben* auswählen.

- 5 Klicken Sie auf *Schreiben*, um das Betriebssystem herunterzuladen, die Speicherkarte zu formatieren und das Raspberry Pi OS auf der Speicherkarte zu installieren.
- 6 Wenn die Meldung *Sie können die SD-Karte nun aus dem Lesegerät entfernen* erscheint, nehmen Sie die Speicherkarte aus dem Kartenleser am PC heraus und klicken auf *Weiter*.

## LED leuchtet

Für das erste Experiment wird kein Programm benötigt. Der Raspberry Pi dient dabei nur als Stromversorger für die LED. Das Experiment zeigt, wie LEDs angeschlossen werden. Achten Sie darauf, dass die 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. Dort hinein werden die Kathoden aller LEDs gesteckt 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

### Programme zum Download

Die im Adventskalender verwendeten Programme gibt es zum Download unter [www.franzis.de/adventskalender/adventskalender-fuer-raspberry-pi](http://www.franzis.de/adventskalender/adventskalender-fuer-raspberry-pi).



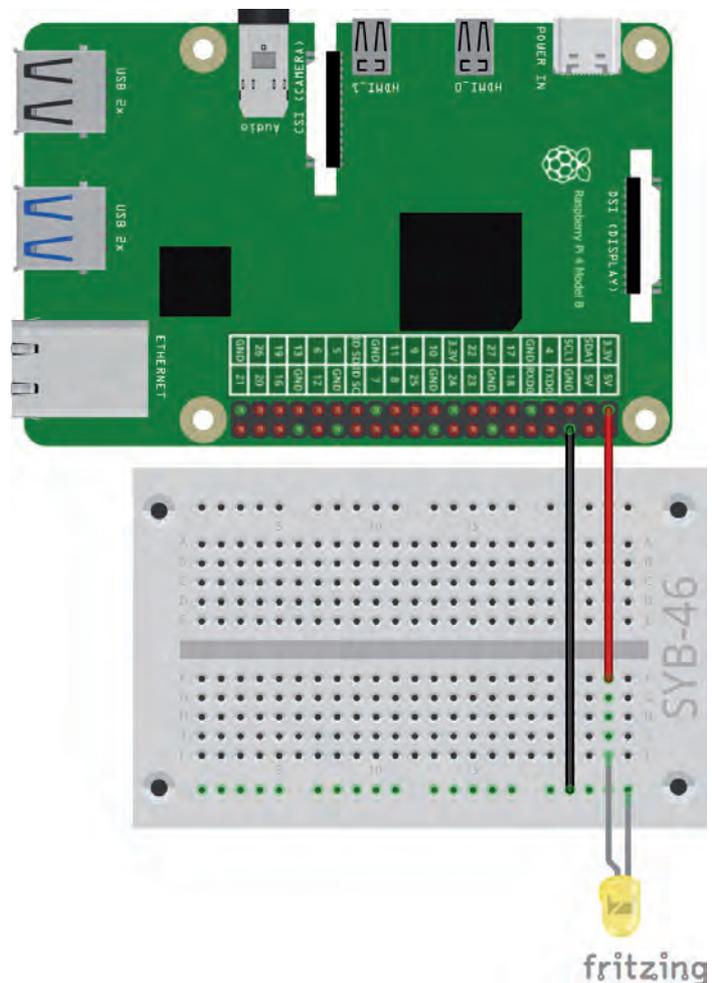
Öffnen Sie die Webseite direkt mit dem vorinstallierten Browser auf dem Raspberry Pi und laden Sie die Zip-Datei in das Home-Verzeichnis `/home/pi` herunter.



Starten Sie den Dateimanager auf dem Raspberry Pi. Er 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 die einzelnen Leitungen auf den Schaltplänen besser erkennen können.



Die erste LED leuchtet am Raspberry Pi.

## Heute im Adventskalender

- 2 GPIO-Verbindungskabel

## LED leuchtet auf der Weihnachtskrippe

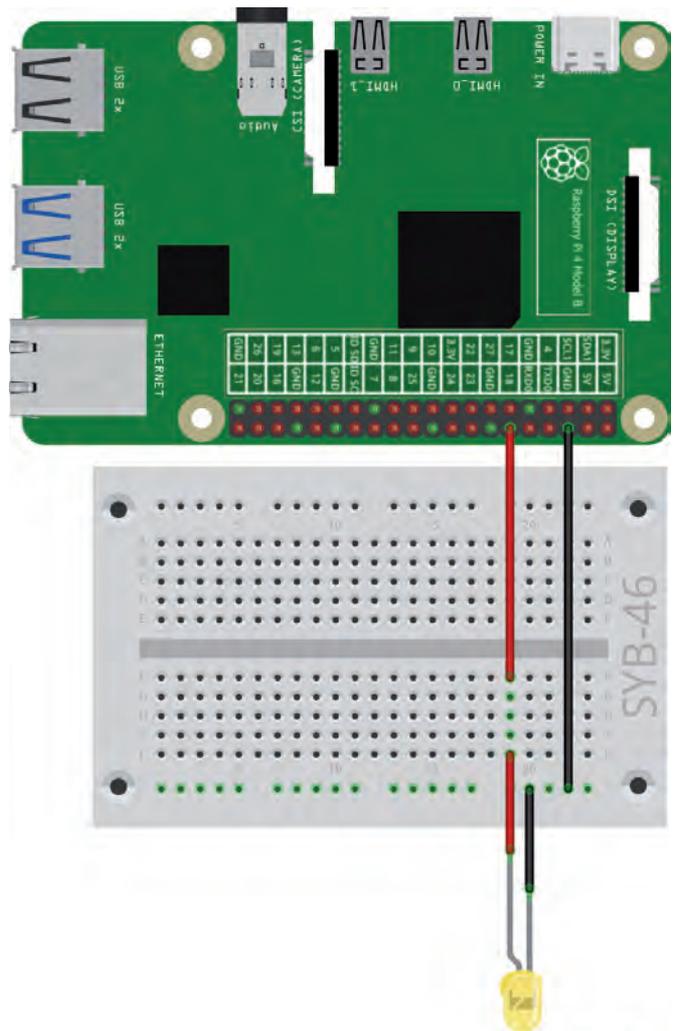
Die LEDs werden in die auf der Weihnachtskrippe gekennzeichneten Löcher gesteckt. Schneiden Sie die Krippe dazu entlang der auf der Rückseitenpappe des Adventskalenders vorgezeichneten Linien aus.

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. Sollten 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, 4 GPIO-Verbindungskabel

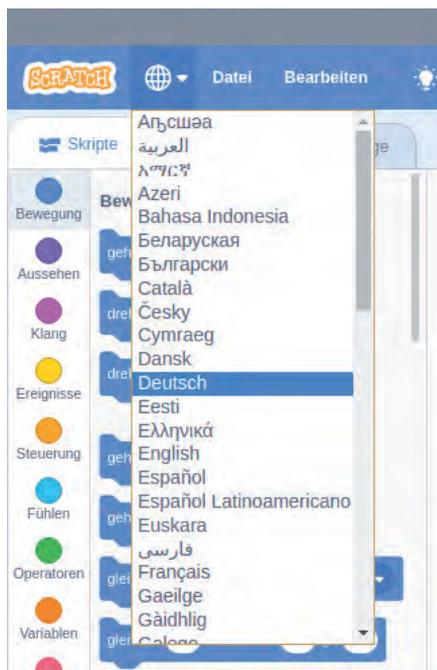
Stecken Sie die LED oben an den Stern auf der Weihnachtskrippe.

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



fritzing

Die erste LED leuchtet auf der Weihnachtskrippe.



Scratch 3 auf Deutsch umschalten.

## Scratch 3

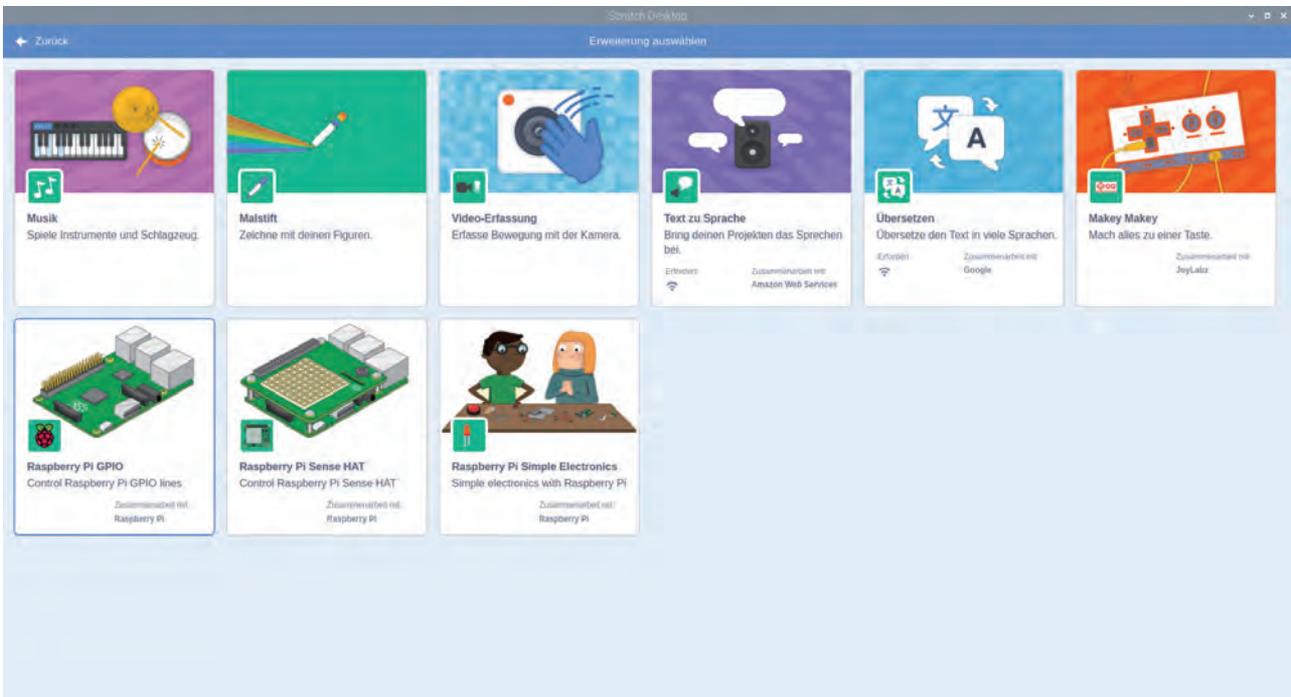


Scratch ist eine intuitive visuelle Programmierumgebung, mit der Kinder und Programmierneulinge schnell Ideen umsetzen können, ohne sich zuerst mit Programmiertheorie auseinandersetzen zu müssen. Wir verwenden für die Projekte in diesem Adventskalender das modernere Scratch 3, zu finden im Startmenü unter *Entwicklung*. Scratch 3 ist nur in der Vollversion von Raspberry Pi OS vorinstalliert, kann aber über *Recommended Software* im Menü *Einstellungen* nachinstalliert werden. Dort finden Sie auch frühere Scratch-Versionen.

Klicken Sie in Scratch 3 als Erstes oben links auf die Weltkugel, um die Scratch-Benutzeroberfläche auf Deutsch umzuschalten, wenn sie nicht bereits automatisch auf Deutsch angezeigt wird.

Scratch 3 unterstützt verschiedene Hardwarekomponenten an den GPIO-Pins, die als Zusatzmodule einmal aktiviert werden müssen:

Klicken Sie unterhalb der Blockpalette auf das Symbol *Erweiterung hinzufügen* und wählen Sie die Erweiterung *Raspberry Pi GPIO*.



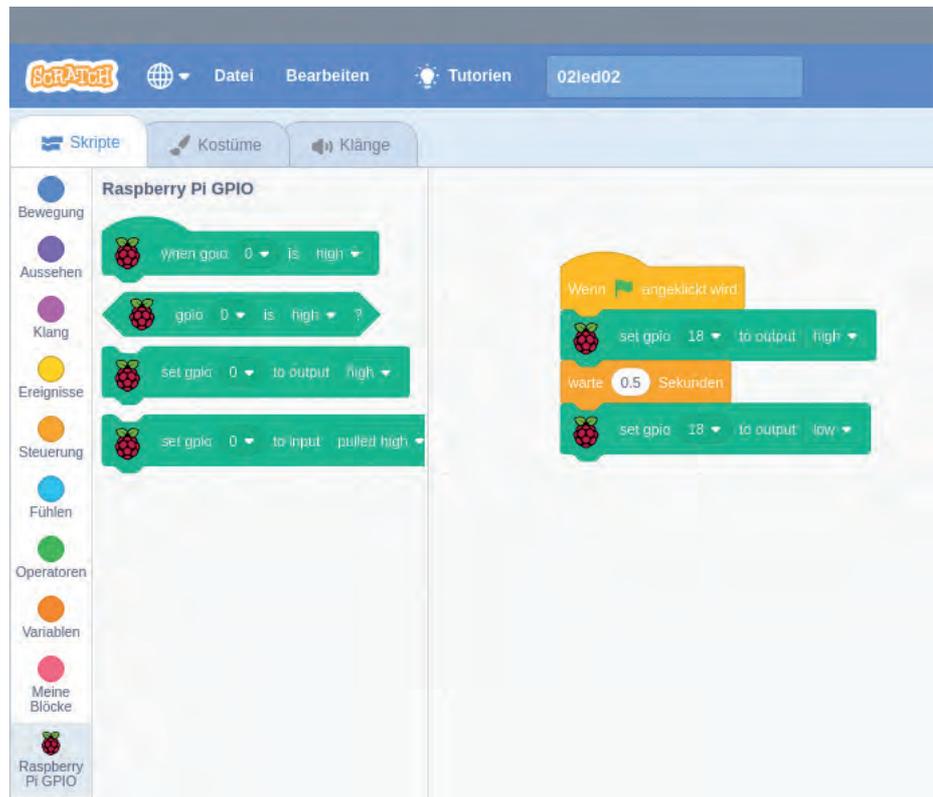
Erweiterungen in Scratch 3 installieren.

In der Blockpalette erscheinen ganz unten vier neue Scratch-Blöcke zur GPIO-Steuerung.

## Das Programm

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

Sie können das Programm auf dem Bildschirm selbst zusammenbauen oder das Programm 021ed02.sb3 aus dem Download zum Adventskalender verwenden. Wählen Sie dazu im Menü *Datei/Hochladen von deinem Computer* und klicken Sie im nächsten Dialogfeld links oben auf *Persönlicher Ordner*, um das persönliche Home-Verzeichnis auszuwählen, in dem die heruntergeladenen Programme liegen.



Das Scratch-Programm 021ed02 lässt die LED eine halbe Sekunde leuchten.

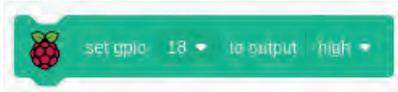
## Raspberry-Pi-Adventskalender

Möchten Sie das Programm selbst zusammenbauen, ziehen Sie die Blöcke, die Sie brauchen, einfach aus der Blockpalette in das Skriptfenster in der Mitte von Scratch.

Klicken Sie in Scratch ganz links auf das gelbe Symbol *Ereignisse*. Dann werden die entsprechenden Blöcke in der Blockpalette angezeigt.



Der Block *Wenn (grüne Fahne) 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.



Am Anfang wird der GPIO-Pin 18 mit dem Block *set gpio 18 to output high* als Ausgang definiert und eingeschaltet. Jeder GPIO-Pin kann entweder Ausgang oder Eingang sein. Wählen Sie dazu im linken Listenfeld dieses Blocks den GPIO-Pin 18 aus und im rechten Listenfeld die Option *high*, was „eingeschaltet“ bedeutet.



Danach wartet das Programm eine halbe Sekunde. Dazu bietet Scratch einen eigenen Block *warte ... Sekunden* 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 den Scratch-Block *set gpio 18 to output low* die am GPIO-Pin 18 angeschlossene LED wieder ausgeschaltet.



Das Programm startet, wenn man oben rechts im Scratch-Fenster links oberhalb der sogenannten Bühne auf das grüne Fähnchen klickt.

# Tag 3

## Heute im Adventskalender

- 1 LED rot mit Vorwiderstand

## Zwei LEDs blinken abwechselnd

Das heutige Experiment 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

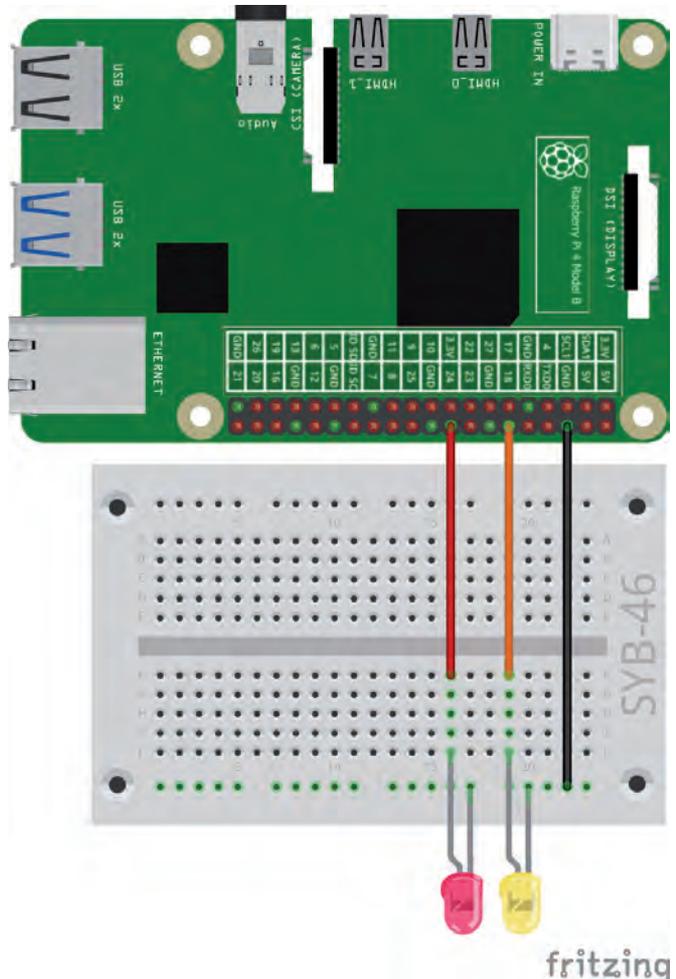
Die LEDs stecken bei diesem Experiment direkt auf dem Steckbrett mit den Kathoden in der Masseleiste.

## Das Programm

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



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



Zwei LEDs blinken am Raspberry Pi.



Das Programm 031ed03 steuert die beiden LEDs.

# Tag 4

## Heute im Adventskalender

- 3 GPIO-Verbindungskabel (lang)

## Ein Stern schaltet zwei LEDs um

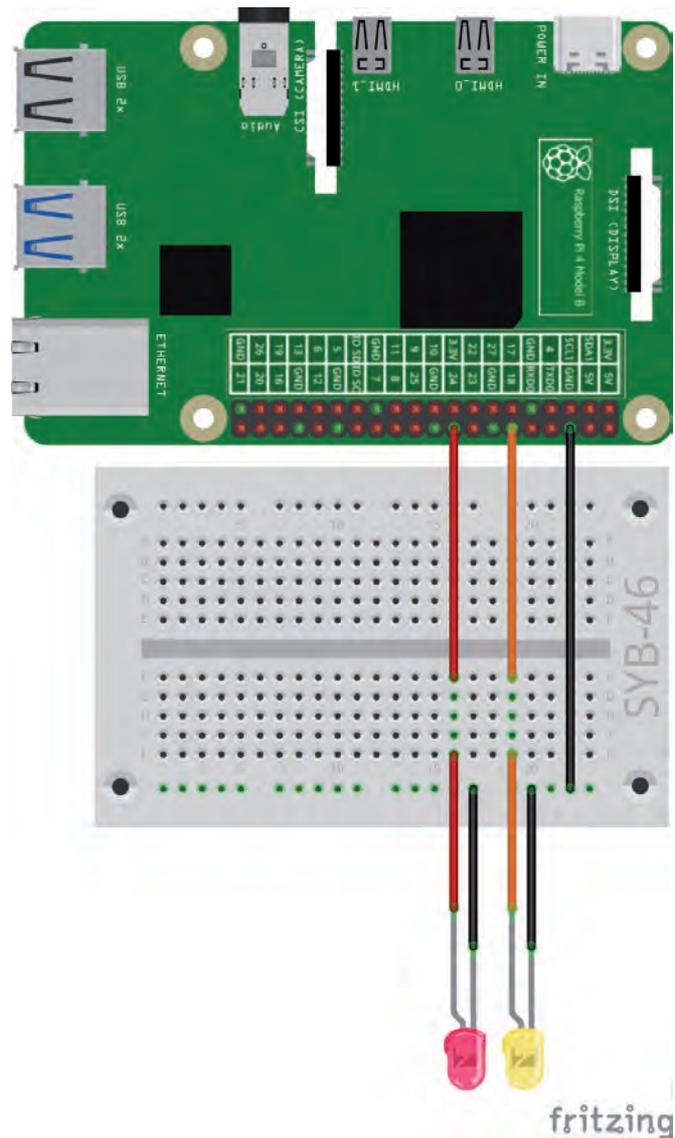
Das Experiment des 4. Tags schaltet zwei LEDs ein oder aus. Zur Steuerung wird ein Objekt in Scratch verwendet, das auf dem Bildschirm bewegt werden kann.

**Bauteile:** 1 Steckbrett SYB-46, 1 LED gelb mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 7 GPIO-Verbindungskabel

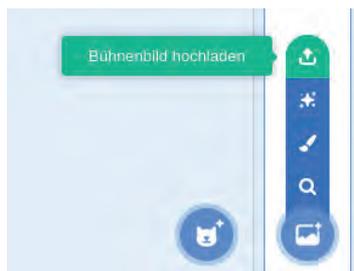
Mit den zusätzlichen GPIO-Kabeln können Sie die LEDs auf die Krippe stecken.

## Das Programm

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



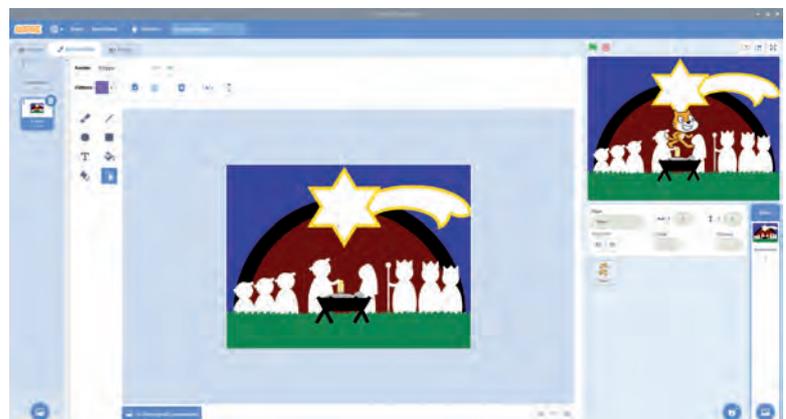
Zwei LEDs werden über ein Scratch-Programm umgeschaltet.



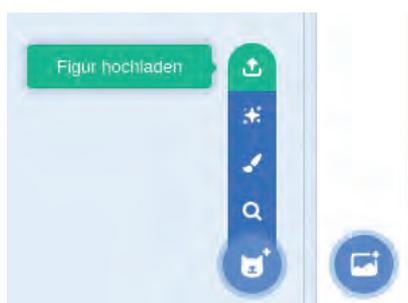
Hintergrundbild hochladen.

Fahren Sie mit der Maus unten rechts auf das Symbol *Bühnenbild wählen* und klicken Sie dann auf das obere Symbol der Symbolleiste, *Bühnenbild hochladen*. Laden Sie das Bild *Krippe.png* aus dem Download in das Programm.

Löschen Sie als Nächstes die Katze, die in diesem Programm nicht gebraucht wird. Klicken Sie dazu auf die Katze im Figurenfenster und dann auf das blaue Papierkorb-symbol.



Das neue Hintergrundbild für das Programm.



Figur hochladen.

# Raspberry-Pi-Adventskalender

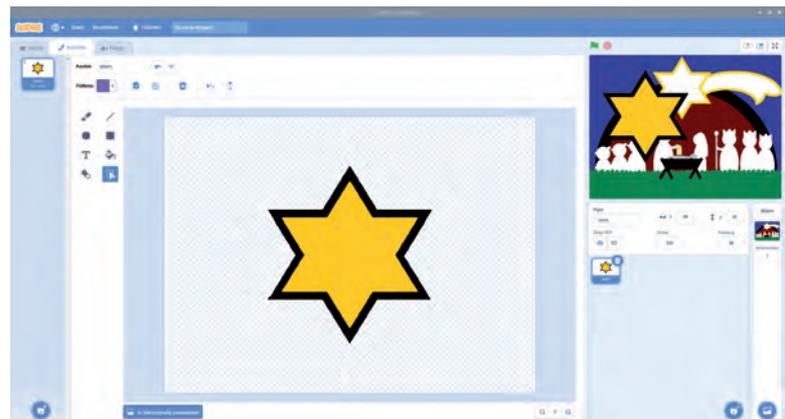
Im Programm soll eine Sternfigur die LEDs umschalten. Sie können diese Figur direkt in Scratch malen oder die Datei `stern.png` aus dem Download als neue Figur über das Symbol *Figur hochladen* hochladen.

Der Stern ist standardmäßig viel zu groß für die Szene.

Scratch enthält ein einfaches Malprogramm, mit dem Sie die Figur bearbeiten können. Schalten Sie im Skriptfenster auf die Registerkarte *Kostüme*, um dieses Malprogramm zu öffnen.

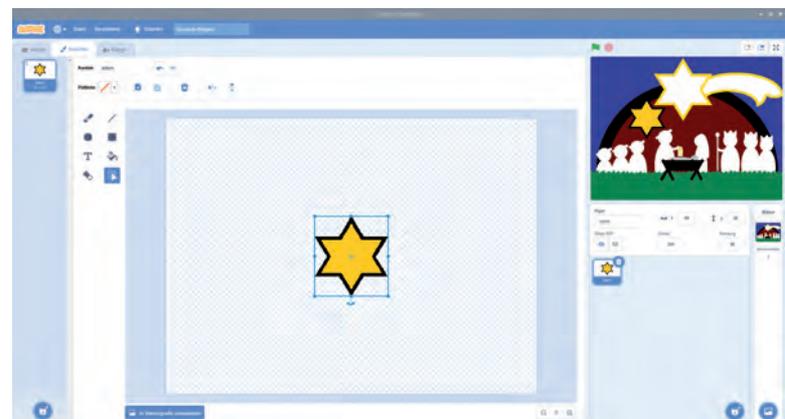
Schrumpfen Sie den Stern auf die passende Größe. Ziehen Sie dazu mit dem Symbol *Auswählen* ein Rechteck um den Stern und ziehen Sie dann an einer der Ecken, um ihn zu verkleinern.

Schieben Sie den Stern anschließend auf der Scratch-Bühne mittig auf den großen Stern des Hintergrundbilds.

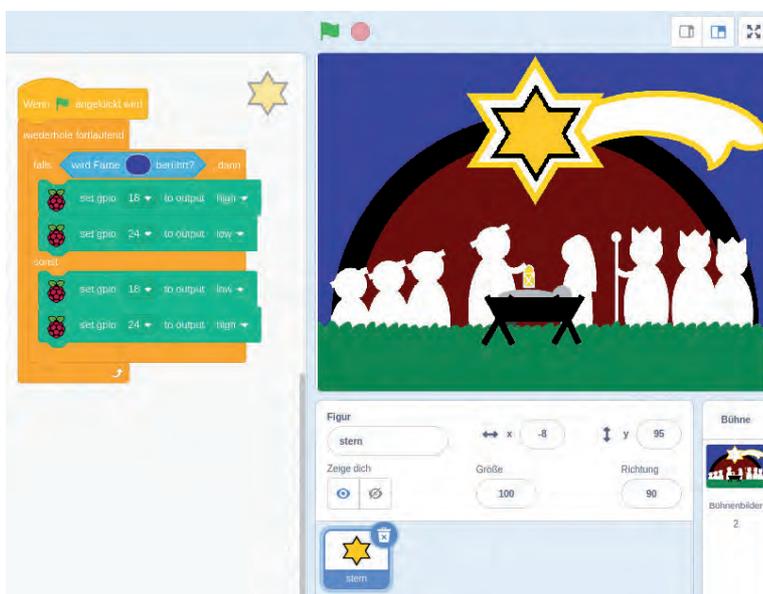


Der im Malprogramm von Scratch importierte Stern.

Figuren sind in Scratch frei verschiebbar, während das Programm läuft. Schieben Sie den Stern auf den dunkelblauen Himmel, leuchtet die andere LED. Das Programm `04krippe01` startet, nachdem die GPIO-Pins initialisiert wurden, eine Endlosschleife, die ständig prüft, ob der Stern eine blaue Fläche berührt.

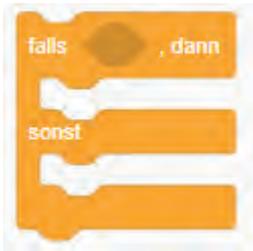


Der verkleinerte Stern.



Das Programm `04krippe01` schaltet zwei LEDs mit einem Stern um.

## Raspberry-Pi-Adventskalender



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

Setzen Sie die *set gpio ...*-Blöcke zum Umschalten der LEDs in die beiden Klammern des *falls ... dann ... sonst ...*-Blocks.

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

Tippen Sie auf das farbige Feld innerhalb dieses Blocks, erscheinen Farbregrler und ganz unten eine Pipette. Klicken Sie mit dieser auf eine blaue Fläche auf dem Bühnenbild. Damit wird diese Farbe in der Abfrage ausgewählt.



Farbe auswählen.



Starten Sie das Programm mit einem Klick auf das grüne Fähnchen. Ziehen Sie den Stern in Richtung der linken oberen Bildschirmecke, sodass er den blauen Hintergrund berührt. Das Programm schaltet automatisch auf die andere LED um.



Das Vollbildsymbol oben rechts in der Ecke des Scratch-Fensters schaltet auf den Vollbildmodus um, in dem die Scratch-Bühne das ganze Fenster füllt. Alle Bedienelemente verschwinden. Ein Klick auf dasselbe Symbol schaltet wieder zurück.



Das Programm im Vollbildsymbol.

# Tag 5

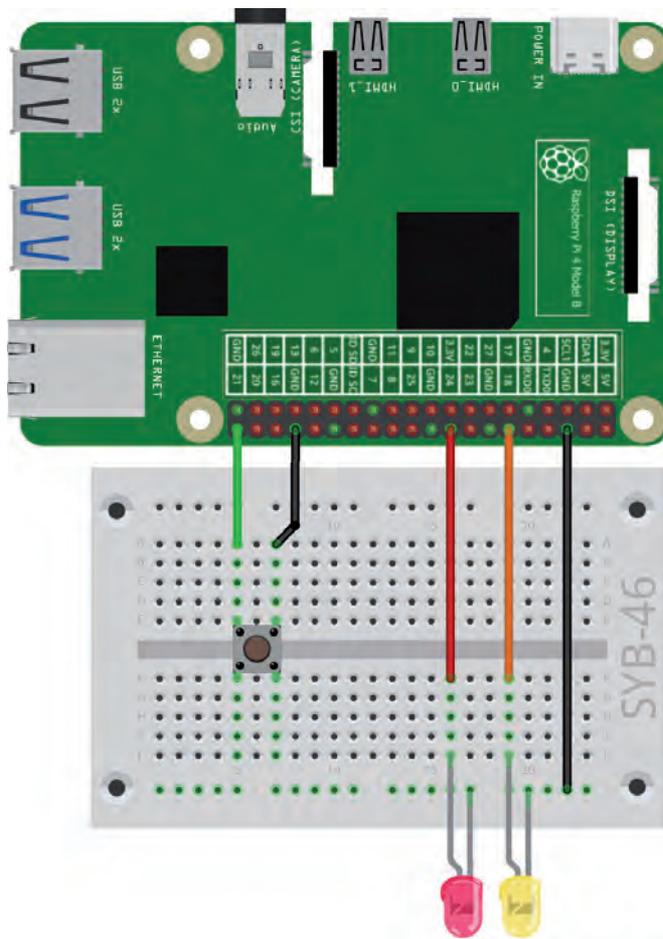
## Heute im Adventskalender

● 1 Taster

## LEDs mit Taster umschalten

Das Experiment des 5. Tags schaltet die LEDs nicht automatisch um, sondern erst wenn der Benutzer einen Taster betätigt. GPIO-Pins können nicht nur Daten ausgeben, zum Beispiel über LEDs, sie können auch zur Dateneingabe verwendet werden. Dazu müssen sie im Programm als Eingang definiert werden. Zur Eingabe verwenden wir im nächsten Projekt einen Taster, der direkt auf die Steckplatine gesteckt wird.

**Bauteile:** 1 Steckbrett SYB-46, 1 LED gelb mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 1 Taster, 5 GPIO-Verbindungskabel



fritzing LEDs mit Taster umschalten.

Man unterscheidet in der Digitalelektronik zwischen den Signalen *low* (= niedrig) und *high* (= hoch). Dabei gilt für Eingänge:

*low* – Der Eingang ist mit der Masse verbunden.

*high* – Am Eingang liegt eine positive Spannung an oder gar keine, wenn der Pin nicht angeschlossen ist. Damit ein nicht angeschlossener Eingang immer einen eindeutigen Zustand hat, sind im Raspberry Pi an jedem Eingangspin Pull-up-Widerstände eingebaut, die den Eingang mit +3,3V verbinden und damit auf *high* hochziehen. Ist ein Eingang mit Masse verbunden, überwiegt dieses deutlich stärkere Signal und der Eingang hat ein *low*-Signal.

# Raspberry-Pi-Adventskalender

## Anschluss des Tasters

Wird der Taster gedrückt, ist er mit Masse verbunden und liefert damit ein *low*-Signal, nicht *high*, wie man für „eingeschaltet“ erwarten würde.

## Das Programm

Das Programm 051ed05 schaltet zwei an den GPIO-Pins 18 und 24 angeschlossene LEDs auf Tastendruck um. Es verwendet mehrere voneinander unabhängige Scratch-Programmblöcke. Zur Abfrage eines Tasters wird keine Endlosschleife benötigt. In anderen Programmiersprachen, die diese Möglichkeit nicht bieten, müsste man eine Endlosschleife programmieren, die ständig darauf wartet, dass der Taster gedrückt wird.



Das Programm 051ed05 steuert die beiden LEDs über einen Taster.

Ausgänge werden bei ihrer ersten Verwendung über den Block *set gpio ... to output ...* automatisch eingerichtet. Eingänge müssen dagegen am Anfang eines Programms einmal initialisiert werden, damit sie später jederzeit im Programm abgefragt werden können. Bei dieser Initialisierung legt man auch fest, ob ein Pull-up- oder ein Pull-down-Widerstand für diesen Eingang auf dem Raspberry Pi eingeschaltet werden soll. Ist der Taster mit Masse verbunden, muss ein Pull-up-Widerstand den offenen Eingang auf *high* ziehen, wenn der Taster nicht gedrückt ist

Bei Klick auf die grüne Fahne wird Pin 21 mit dem Block *set gpio 21 to input pulled high* aus der Erweiterung **Raspberry Pi GPIO** als Eingang mit Pull-up-Widerstand initialisiert. Danach wird die LED an Pin 18 mit dem Block *set gpio 18 to output high* ein- und die LED an Pin 24 mit dem Block *set gpio 24 to output low* ausgeschaltet.

Der Block *when gpio 21 is ...* wertet den Taster als Ereignis aus.

Der linke Programmblock schaltet, wenn der Taster gedrückt wird, die LED an Pin 18 aus und die andere LED an Pin 24 ein.

Der rechte Programmblock schaltet umgekehrt, wenn der Taster losgelassen wird, die LED an Pin 18 ein und die andere LED an Pin 24 aus.

## 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 dann an passender Stelle im Programm eingefügt werden. Das Ganze funktioniert auch mit der Tastenkombination Strg+c (kopieren) und Strg+v (einfügen).

# Tag 6

## Heute im Adventskalender

- Schaltdraht

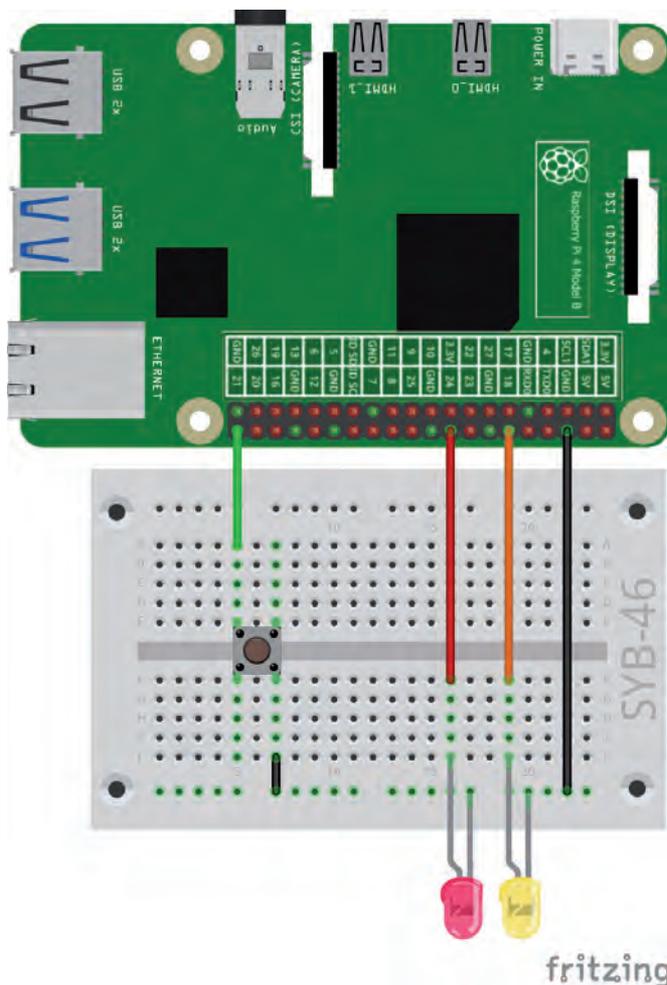
### Schaltdraht

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 durch kurzen Tastendruck umschalten

Im Programm des 6. Tags werden auch wieder zwei LEDs über einen Taster umgeschaltet. Diesmal schaltet jeder Druck des Tasters die LEDs in den jeweils anderen Zustand um. Der Taster muss also nicht gedrückt gehalten werden.

**Bauteile:** 1 Steckbrett SYB-46, 1 LED gelb mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 1 Taster, 1 Drahtbrücke, 4 GPIO-Verbindungskabel



LEDs mit Taster umschalten.

Mit einer kurzen Drahtbrücke zwischen Taster und Masseleiste spart man sich ein GPIO-Anschlusskabel zu einem weiteren GND-Pin am Raspberry Pi.

# Raspberry-Pi-Adventskalender

## Das Programm

Das Programm 061ed06 startet wieder mit einem Klick auf das grüne Fähnchen. Wie im letzten Programm wird beim Start die LED an Pin 18 ein- und die andere LED an Pin 24 ausgeschaltet.

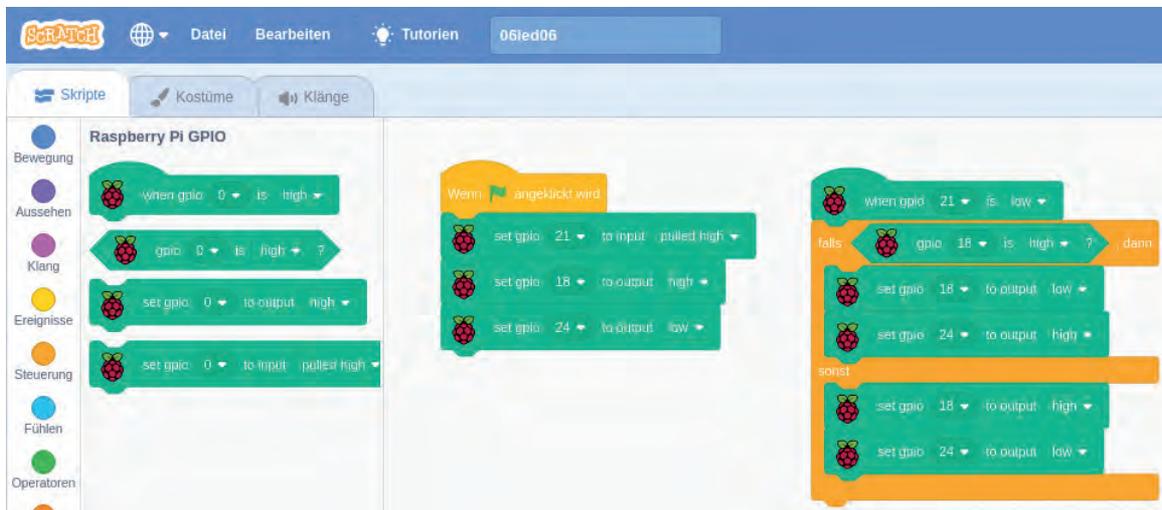
Zur Abfrage des Tasters an Pin 21 wird eine andere Programmlogik verwendet:

Der Block *when gpio 21 is ...* wertet den Taster wieder als Ereignis aus und startet diesmal eine *falls ... dann ... sonst ...*-Abfrage.

Ziehen Sie in das Abfragefeld dieses *falls ... dann ... sonst ...*-Blocks einen Block *gpio ... is ...* aus der **Raspberry Pi GPIO**-Erweiterung. Dieser Block kann nicht nur Taster, sondern jeden GPIO-Pin abfragen.

Ist die LED an Pin 18 eingeschaltet, werden die beiden Blöcke in der ersten Klammer der *falls ... dann ... sonst ...*-Abfrage ausgeführt, und damit wird die LED an Pin 18 aus- und die LED an Pin 24 eingeschaltet.

Ist die LED an Pin 18 dagegen ausgeschaltet, werden die beiden Blöcke in der zweiten Klammer der *falls ... dann ... sonst ...*-Abfrage ausgeführt, und damit wird die LED an Pin 18 ein- und die LED an Pin 24 ausgeschaltet.



Das Programm 061ed06 schaltet die beiden LEDs über einen Taster um.

## Heute im Adventskalender

- 2 GPIO-Verbindungskabel

## LEDs mit Scratch-Steuerpult umschalten

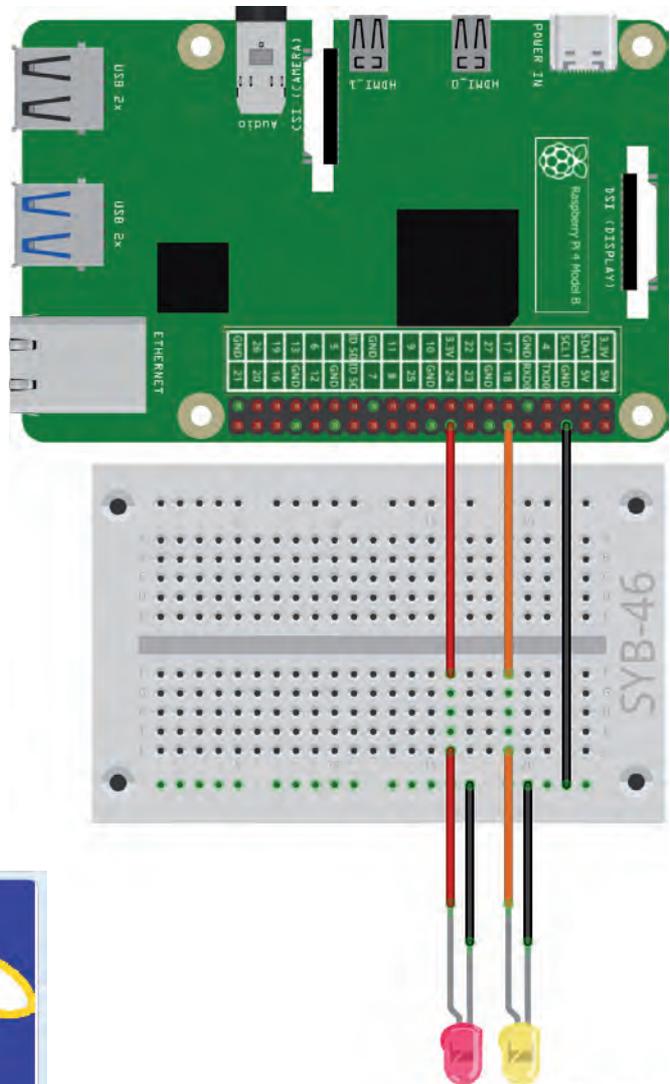
Im Programm des 7. Tags werden zwei LEDs durch Klicks in Scratch ein- und ausgeschaltet. Dabei zeigt das Scratch-Steuerpult den Zustand der LEDs an.

**Bauteile:** 1 Steckbrett SYB-46, 1 LED gelb mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 7 GPIO-Verbindungskabel

## Das Programm

Das Programm `07krippe02` verwendet das gleiche Hintergrundbild wie die Programme der letzten Tage.

Für die beiden LEDs werden auf dem Schweiß des Sterns zwei kreisförmige Figuren mit den Namen *Gelb* und *Rot* angelegt. Löschen Sie die Katze und malen Sie zunächst eine der Figuren. Sie können auch die Datei `kreis_gelb.png` aus den Downloads importieren. Die zweite Figur können Sie später mitsamt allen Einstellungen und Skriptblöcken duplizieren und müssen Sie dann nur noch geringfügig anpassen.



Zwei LEDs auf der Weihnachtskrippe umschalten.



Die beiden Figuren für die LEDs.

Jede Figur in Scratch kann über sogenannte Kostüme ihr Aussehen verändern. Wählen Sie die Figur für die gelbe LED aus und gehen Sie im Skriptfenster auf die Registerkarte *Kostüme*.

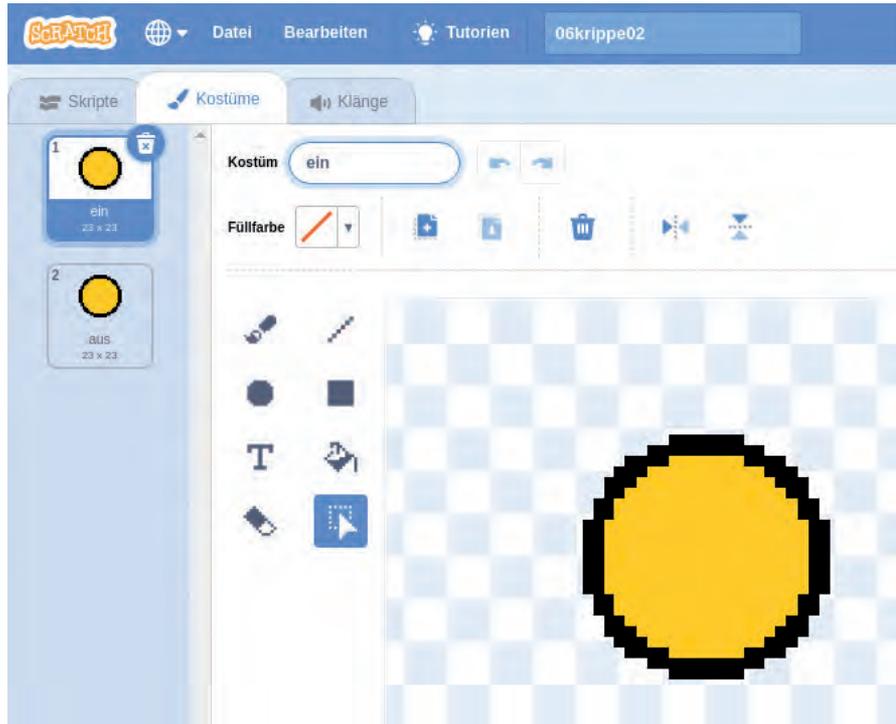
Duplizieren Sie das vorhandene Kostüm mit einem Rechtsklick auf das Kostüm links in der Leiste. Wählen Sie im Menü *Duplizieren*. Benennen Sie die beiden Kostüme um: *ein* für die eingeschaltete und *aus* für die ausgeschaltete LED.

# Raspberry-Pi-Adventskalender

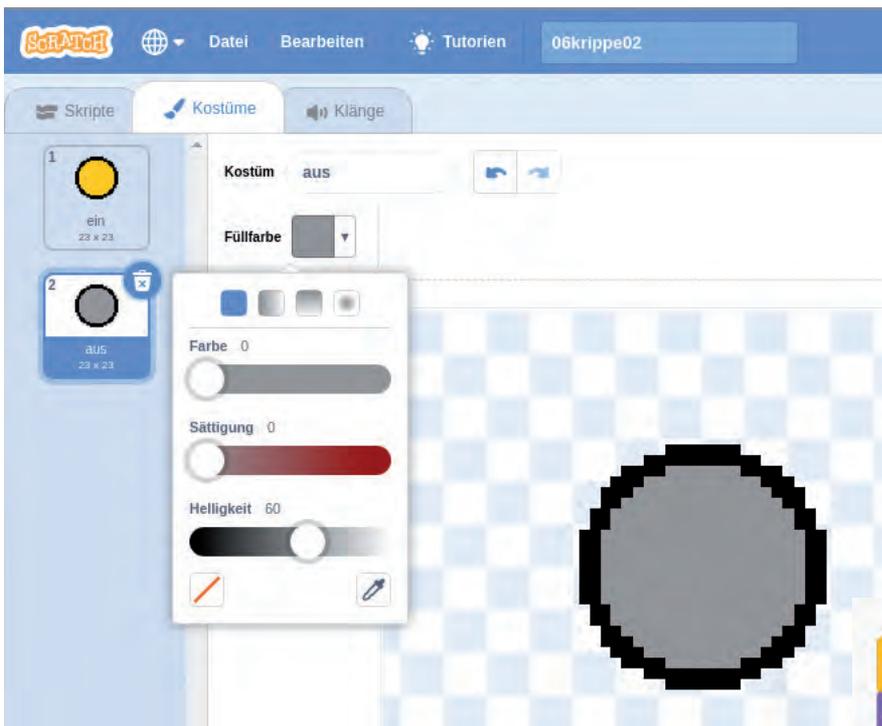
Bearbeiten Sie das Kostüm *aus* mit dem Malprogramm. Füllen Sie die farbige Fläche mit Grau, um die ausgeschaltete LED darzustellen. Legen Sie dazu über das Symbol *Füllfarbe* ein helles Grau fest und füllen Sie die gelbe Fläche im Kostüm mithilfe des Farbeimersymbols mit dieser Farbe.

Das Programm besteht aus zwei Teilen für die beiden Figuren, jeweils ein Teil pro Figur. Da sich diese sehr ähneln, stellen wir zuerst die erste Figur für die gelbe LED fertig und duplizieren sie einschließlich ihrer Programmblöcke.

Das grüne Fähnchen wird in diesem Programm nicht benötigt. Das Programm startet, wenn die Figur angeklickt wird. Auf



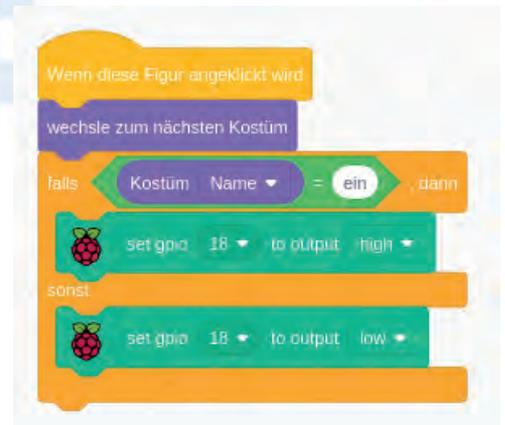
Die beiden Kostüme der Figur *Gelb*.



Kostüm für eine ausgeschaltete LED im Malprogramm.

der Blockpalette *Ereignisse* finden Sie dazu einen Block *Wenn diese Figur angeklickt wird*.

Danach schaltet der Block *wechsle zum nächsten Kostüm* von der Blockpalette *Aussehen* auf das andere Kostüm.



Programmblöcke für die gelbe LED.

Jetzt fragt eine *falls ... dann ... sonst ...*-Abfrage, welches Kostüm gerade aktiv ist. Ziehen Sie dazu den Block *... = ...* aus der Blockpalette *Operatoren* in das Abfragefeld. In diesem Block werden zwei Parameter miteinander verglichen. Ziehen Sie in das erste Feld den Block *Kostüm* aus der Blockpalette *Aussehen* und wählen Sie im Listenfeld *Name*. Tragen Sie in das zweite Feld des Gleichheitsoperators das Wort *ein* ein. Das Kostüm *ein* steht für die eingeschaltete LED.



Figur der gelben LED duplizieren.

Figuren und wählen Sie im Kontextmenü *Duplizieren*. Benennen Sie die erste Figur in *Gelb* und die neue Figur in *Rot* um. Schalten Sie oben links auf die Palette *Kostüme* und füllen Sie den Kreis im Kostüm *ein* mit roter Farbe. Die Programmblöcke werden beim Duplizieren mit dupliziert.

In der ersten Klammer der *falls ... dann ... sonst ...*-Abfrage wird die LED an GPIO-Pin 18 eingeschaltet.

In der zweiten Klammer, die ausgeführt wird, wenn das Kostüm mit dem Namen *ein* nicht aktiv ist, die LED also ausgeschaltet sein soll, wird die LED über den GPIO-Pin 18 ausgeschaltet.

Sie können das Programm direkt ausprobieren, indem Sie auf die gelbe Figur klicken. Wird sie gelb dargestellt, leuchtet die LED. Wird sie grau dargestellt, ist die LED ausgeschaltet.

Duplizieren Sie diese Figur mit einem Rechtsklick auf die Figur in der Palette



Programmblöcke für die rote LED und beide LEDs auf der Bühne.

Sie brauchen in den beiden *set gpio ...*-Blöcken die GPIO-Pinnummer nur noch von 18 auf 24 zu ändern und die rote Figur ebenfalls in den Schweif des Sterns zu ziehen. Jetzt können Sie die beiden Figuren anklicken und damit die LEDs ein- und ausschalten.

# Tag 8

## Heute im Adventskalender

- 1 LED grün mit Vorwiderstand

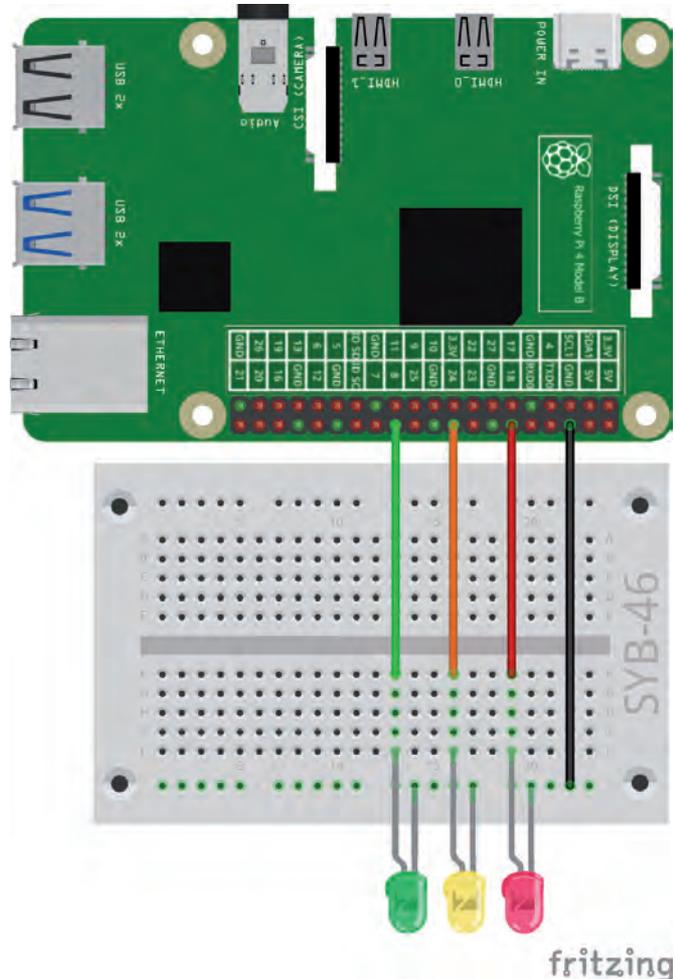
## Lauflicht mit drei LEDs

Das Programm des 8. Tags lässt drei LEDs abwechselnd leuchten. Der Lauflichteffekt entsteht dadurch, dass eine LED eine kurze Zeit leuchtet und, sobald sie ausgeschaltet wird, 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

## Das Programm

Das Programm 081ed08 lässt drei LEDs als Lauflicht leuchten. Die Geschwindigkeit lässt sich interaktiv einstellen, während das Programm läuft. Dazu verwendet das Programm eine Variable *pause*, die die Pause zwischen zwei Schaltvorgängen angibt.



Lauflicht mit drei LEDs.



Das Programm 081ed08 lässt drei LEDs als Lauflicht leuchten.

## Variablen in Scratch 3

Variablen sind kleine Speicherplätze, in denen sich ein Programm eine Zahl oder irgendetwas anderes merkt. Wenn das Programm beendet wird, werden diese Variablenspeicher automatisch geleert. Variablen müssen in Scratch auf der Blockpalette *Variablen* mit einem Klick auf *Neue Variable* erst einmal 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.



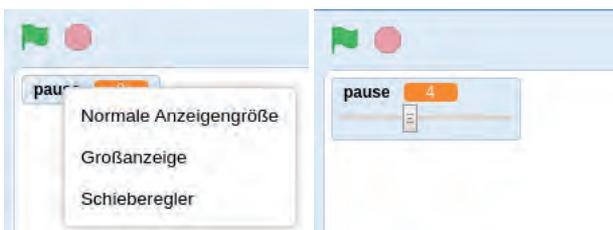
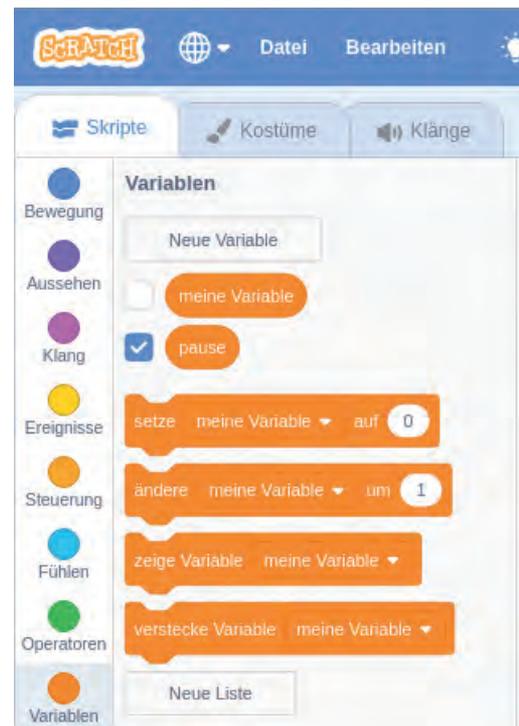
Nach Klick auf das grüne Fähnchen werden als Erstes die drei GPIO-Pins 18, 24 und 8 für die LEDs als Ausgänge definiert, und es werden alle ausgeschaltet, um bei ausgeschalteten LEDs mit dem Lauflicht zu beginnen.

Ein Block *setze pause auf 5* von der Blockpalette *Variablen* setzt die Pause auf 5, was im Programm einer halben Sekunde entspricht. Da die Schieberegler in Scratch 3 immer nur ganze Zahlen einstellen, wählen Sie hier Zehntelsekunden. Im Programm werden die eingestellten Variablenwerte durch 10 geteilt, um die tatsächlichen Zeiten in Sekunden einzustellen.

Anschließend startet eine Endlosschleife, die die LEDs nacheinander 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, aktivieren Sie das Häkchen links neben dem Variablensymbol auf der Blockpalette *Variablen*. Damit erscheint die Variable auf der Bühne und zeigt dort immer den aktuellen Wert.

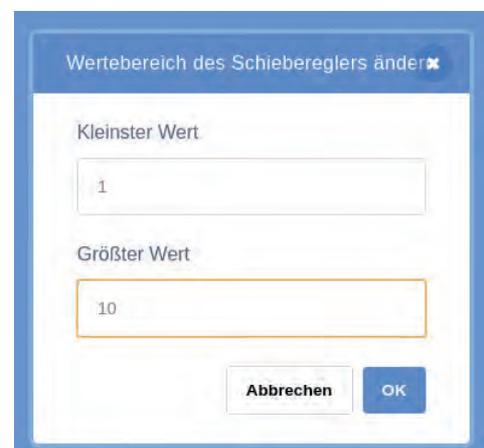
Klicken Sie mit der rechten Maustaste auf die Variable auf der Bühne und wählen Sie im Kontextmenü *Schieberegler*. Damit erscheint ein Regler, mit dem der Variablenwert eingestellt werden kann.



Schieberegler für eine Variable auf der Scratch-Bühne anlegen.

Klicken Sie dann noch auf *Wertebereich des Schiebereglers ändern* und setzen Sie den Minimalwert auf 1 und 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 der Variablen *pause* auf der Bühne die Geschwindigkeit während des Programmlaufs interaktiv einstellen. Variablen auf der Bühne werden auch im Präsentationsmodus angezeigt.



Wertebereich für den Schieberegler einer Variablen festlegen.

# Tag 9

## Heute im Adventskalender

- 2 GPIO-Verbindungskabel

## Lauflicht auf der Krippe

Das Programm des 9. Tags setzt das Programm vom Vortag fort, zeigt aber die leuchtenden LEDs zusätzlich auch auf der Darstellung der Krippe auf der Scratch-Bühne.

**Bauteile:** 1 Steckbrett SYB-46, 1 LED grün mit Vorwiderstand, 1 LED gelb mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 10 GPIO-Verbindungskabel

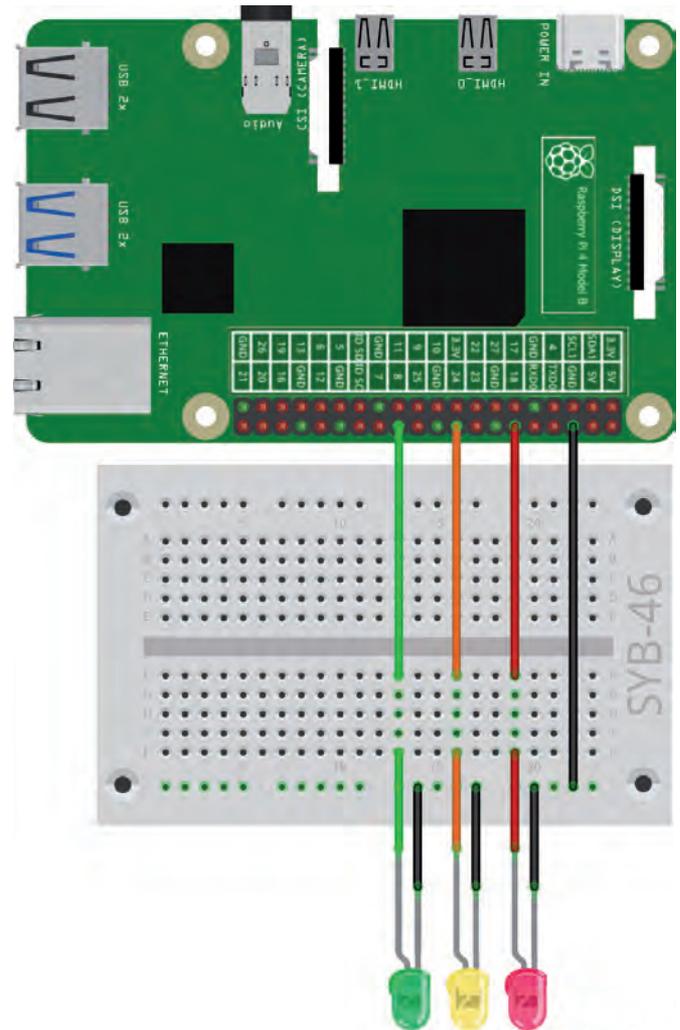
Die zusätzlichen GPIO-Kabel heute machen es möglich, die drei LEDs auf die Krippe zu stecken und sie dort gleichzeitig mit der Scratch-Grafik leuchten zu lassen.

## Das Programm

Das Programm 09krippe03 lässt wieder drei LEDs als Lauflicht leuchten. Zusätzlich wird die Grafik auf der Bühne animiert. Auch hier lässt sich die Geschwindigkeit interaktiv einstellen, während das Programm läuft.

Das Programm können Sie leicht erstellen, indem Sie das Programm 081ed08 vom Vortag dafür umbauen.

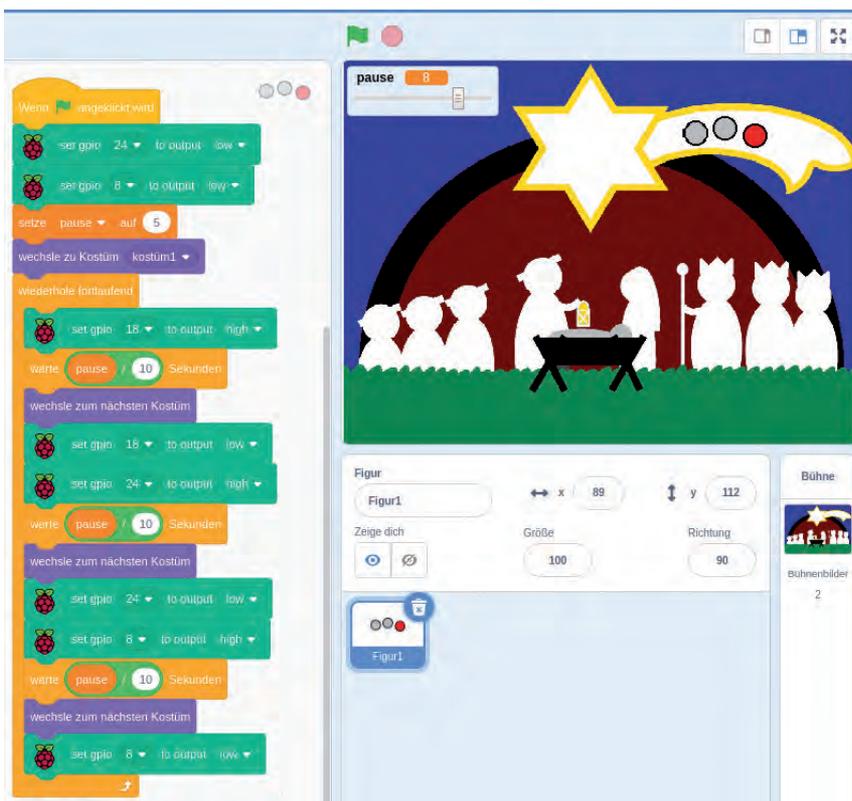
Laden Sie über das Symbol *Bühnenbild hochladen* das bereits bekannte Bild *krippe.png* als Hintergrund in das Programm.



fritzing

Lauflicht mit drei LEDs.

Das Programm 09krippe03 lässt drei LEDs als Lauflicht leuchten.



Neues Kostüm in eine Figur hochladen.

Die Programmblöcke sind in der Katze, der Scratch-Standardfigur, gespeichert. Ändern Sie deren Aus-

sehen in drei farbige Punkte, können Sie die Blöcke direkt übernehmen. Die Katze wird in dem Programm ohnehin nicht benötigt.

Klicken Sie in der *Figuren*-Palette auf die Katze und schalten Sie oben links auf die Registerkarte *Kostüme*. Fahren Sie mit der Maus unten links auf das Symbol *Wähle ein Kostüm* und klicken Sie dann in der Symbolleiste auf *Kostüm hochladen*. Laden Sie hier die Grafik *kreis\_gelb.png* hoch. Daraus wird ein neues Kostüm erstellt.



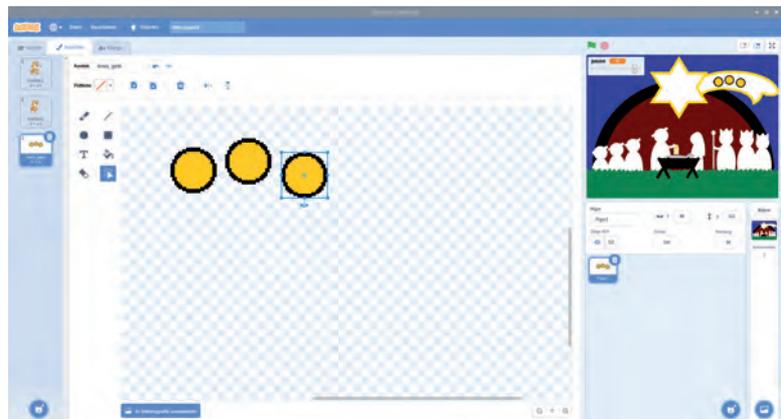
Klicken Sie auf das Symbol *Auswählen* und wählen Sie den Kreis aus, indem Sie ein Quadrat darum herum aufziehen.



Jetzt können Sie diesen Kreis mit dem Symbol *Kopieren* in die Zwischenablage kopieren und daneben zwei weitere Kreise einfügen. Die Kreise werden bogenförmig angeordnet, damit sie auf dem Schweif des Sterns Platz finden.

Schieben Sie auf der Bühne die neue Figur gleich an die passende Stelle. Die beiden vorhandenen Kostüme der Katze können Sie mit den blauen Papierkorbsymbolen löschen.

Duplizieren Sie das Kostüm zweimal und färben Sie dann im ersten Kostüm den Kreis ganz rechts mit roter Farbe, im zweiten den mittleren gelb und im dritten Kostüm den linken Kreis grün. Die übrigen Kreise bekommen einen grauen Hintergrund, da sie ausgeschaltete LEDs darstellen. Die Figur hat jetzt drei Kostüme. Die Programmblöcke wurden aus dem vorherigen Programm übernommen.

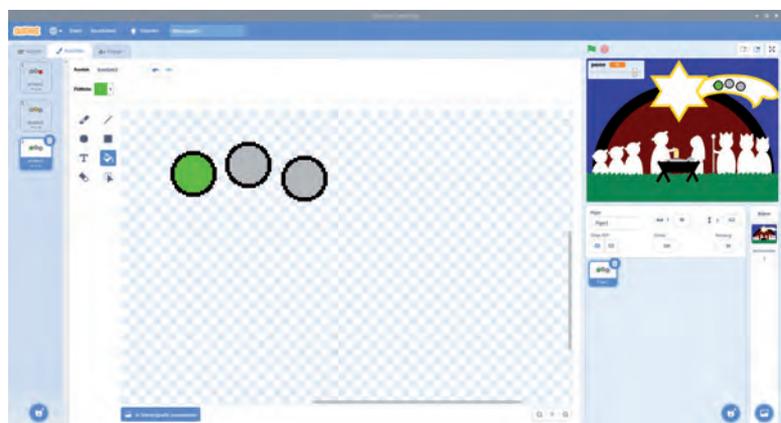


Weitere Kreise kopieren.

Im nächsten Schritt wird das Programm noch so verändert, dass nicht nur die LEDs leuchten, sondern auch die farbigen Punkte in der Scratch-Grafik dargestellt werden.

Ziehen Sie als Erstes direkt vor der *wiederhole fortlaufend*-Schleife einen Block *wechsle zu Kostüm* in das Programm und wählen Sie im Listenfeld *Kostüm 1*. Damit wird der rote Punkt eingeschaltet, und das Programm startet mit der roten LED.

Fügen Sie nach den drei Pausen jeweils einen Block *wechsle zum nächsten Kostüm* ein. Er schaltet, bevor die entsprechenden LEDs umgeschaltet werden, auf das nächste Kostüm um. Nach dem roten Punkt folgen der gelbe und zuletzt der grüne Punkt. Nach der letzten Pause in der *wiederhole fortlaufend*-Schleife schaltet das Lauflicht wieder auf die rote LED. Auch hier kann ein Block *wechsle zum nächsten Kostüm* verwendet werden, da nach dem letzten Kostüm in der Reihe automatisch wieder das erste ausgewählt wird.



Farben der Kreise ändern.



Starten Sie das Programm mit einem Klick auf das grüne Fähnchen. Gleichzeitig mit dem Lauflicht leuchten auch die farbigen Punkte auf der Scratch-Bühne auf.

# Tag 10

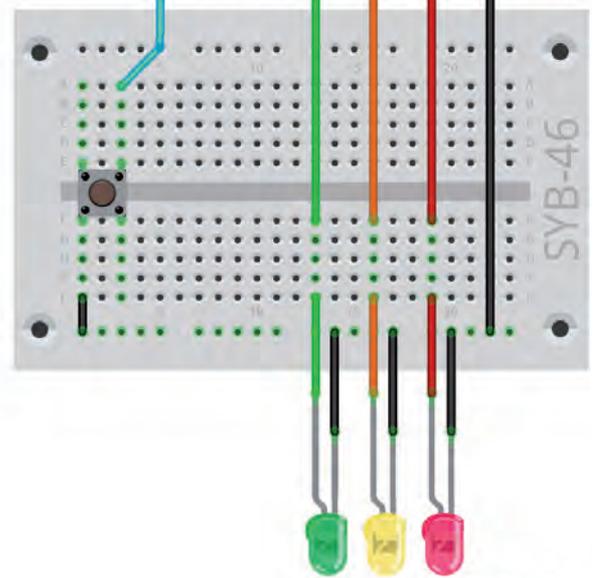
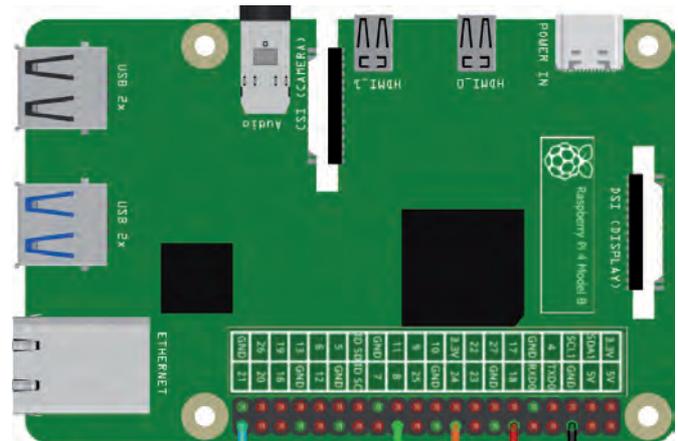
## Heute im Adventskalender

- 2 GPIO-Verbindungskabel

## Lauflicht mit Taster

Das Programm des 10. Tags ist wieder ein Lauflicht, diesmal leuchten die LEDs aber nur, wenn ein Taster gedrückt ist.

**Bauteile:** 1 Steckbrett SYB-46, 1 LED grün mit Vorwiderstand, 1 LED gelb mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 1 Taster, 1 Drahtbrücke, 11 GPIO-Verbindungskabel



fritzing

Lauflicht mit drei LEDs und Taster.



Das Programm 10krippe04 steuert ein Lauflicht mit einem Taster.

## Das Programm

Das Programm 10krippe04 basiert auf dem Programm des vorherigen Tags. Zusätzlich wird Pin 21 als Eingang mit Pull-up-Widerstand initialisiert.

Nach der Initialisierung fragt eine Endlosschleife ständig ab, ob an Pin 21 das Signal *low* anliegt, der Taster also gedrückt ist. Hier wird zur Abfrage der Abfrageblock *gpio ... is ...* mit den spitzen Enden innerhalb einer *falls ... dann*-Abfrage genutzt. Der bisher verwendete Block *when gpio ... is ...* kann hier nicht eingesetzt werden, da er nur aktiviert wird, wenn der Taster von „nicht gedrückt“ (*high*) auf „gedrückt“ (*low*) wechselt. Ein ständig gedrückter Taster würde das Lauflicht nach einem Durchgang beenden und nicht weiterlaufen lassen.

# Tag 11

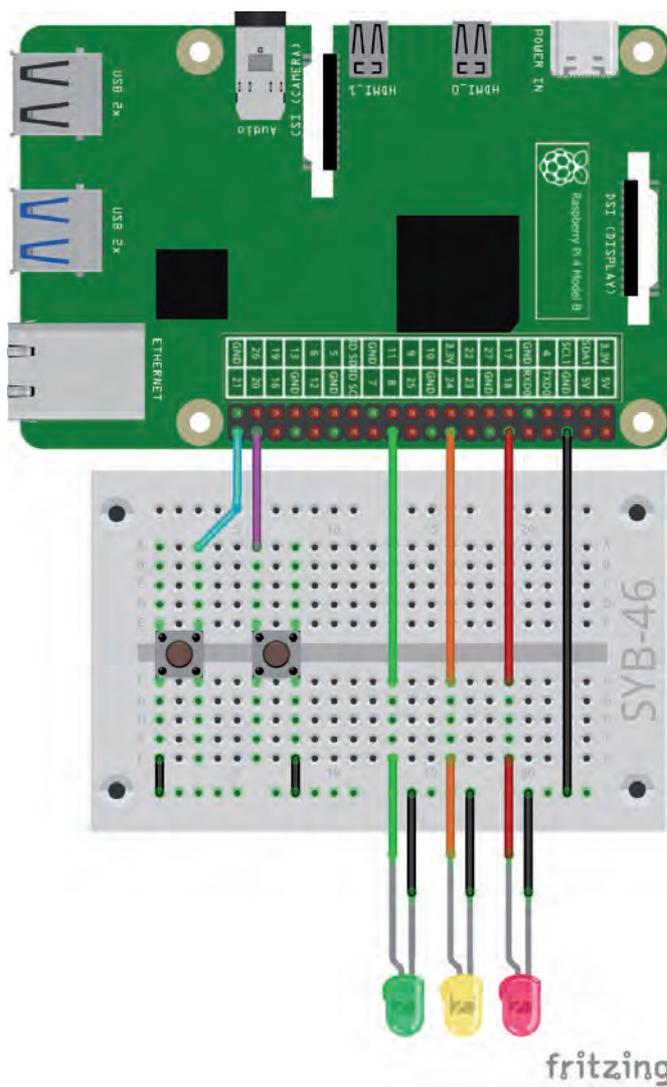
## Heute im Adventskalender

- 1 Taster

## Lauflicht mit zwei Tastern umschalten

Über zwei Taster kann die Richtung des Lauflichts gewählt werden. Der linke Taster lässt das Lauflicht nach links laufen, der rechte nach rechts.

**Bauteile:** 1 Steckbrett SYB-46, 1 LED grün mit Vorwiderstand, 1 LED gelb mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 2 Taster, 2 Drahtbrücken, 12 GPIO-Verbindungskabel



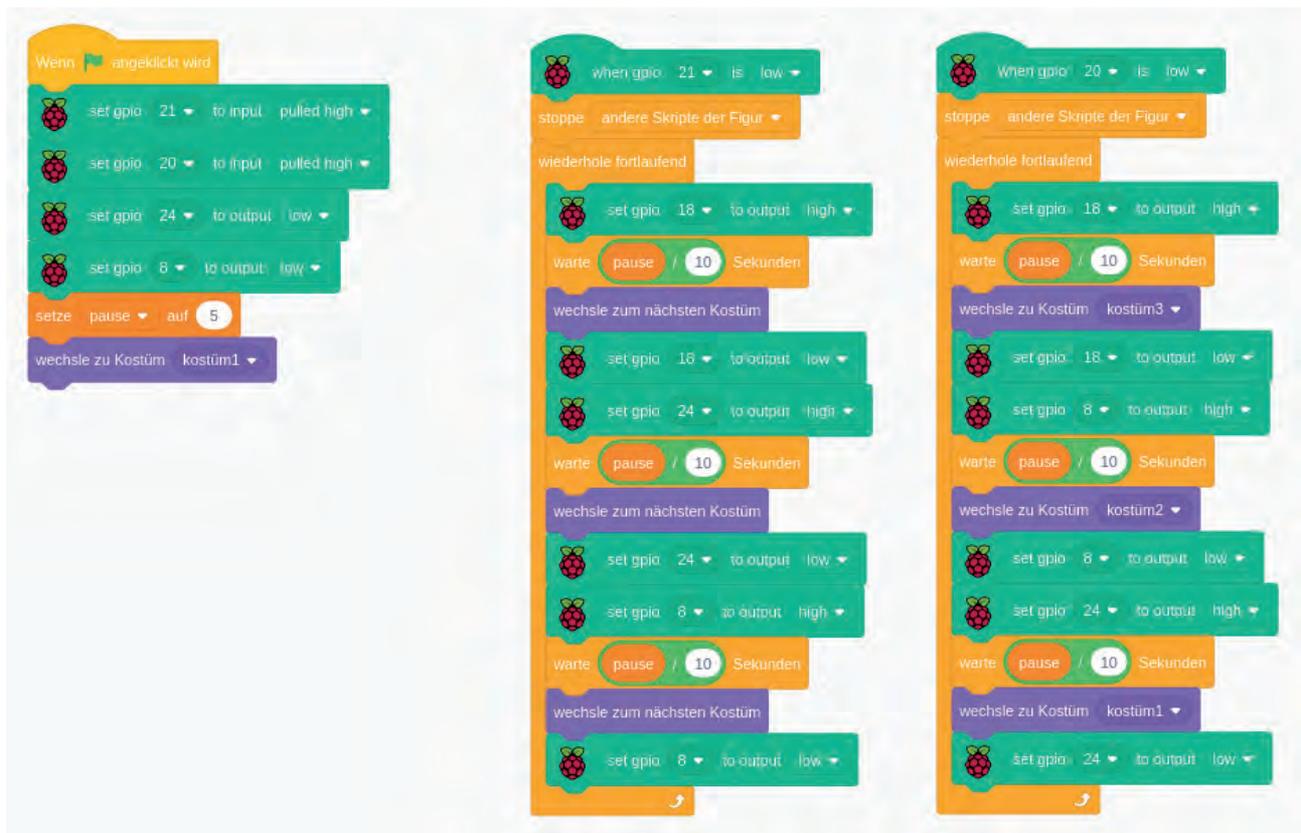
fritzing

Drei LEDs und zwei Taster.

## Das Programm

Das Programm 11krippe05 lässt drei LEDs als Lauflicht leuchten, wobei die Richtung über zwei Taster umgeschaltet wird.

# Raspberry-Pi-Adventskalender



Das Programm 11krippe05 lässt drei LEDs als Lauflicht in verschiedene Richtungen laufen.

Bei Klick auf das grüne Fähnchen werden die beiden Pins 20 und 21 für die Taster als Eingänge mit Pull-up-Widerständen initialisiert, die gelbe und die grüne LED werden ausgeschaltet, und das erste Kostüm wird gewählt.

Die Steuerung der Lauflichteffekte erfolgt über zwei sehr ähnliche Programmblöcke, die starten, wenn einer der Taster gedrückt wird.

Um problemlos auf die andere Richtung für das Lauflicht umschalten zu können, muss ein laufendes Lauflicht zuerst ausgeschaltet werden. Dazu verwenden wir den Block *stoppe ...* von der Blockpalette *Steuerung*. Wählen Sie hier im Listenfeld die Option *andere Skripte der Figur*. Damit werden alle laufenden Skripte gestoppt – auch die Endlosschleifen dieser Figur, der einzigen im Programm. Anschließend startet das Lauflicht.



Das Lauflicht in der Richtung, die bisher verwendet wurde, läuft mit den bereits bekannten Programmblöcken. Für die Gegenrichtung, die mit dem Taster an Pin 20 gestartet wird, sind ein paar Änderungen nötig:

- Das Lauflicht startet weiterhin mit der LED an Pin 18, die Reihenfolge der weiteren LEDs wird aber umgekehrt.
- Der Block *wechsle zum nächsten Kostüm* kann hier wegen der umgekehrten Reihenfolge nicht verwendet werden. Da es aber keinen Block gibt, der zum vorherigen Kostüm in der Liste wechselt, verwenden wir *wechsle zu Kostüm ...*-Blöcke, die zu einer bestimmten Kostümnummer umschalten.

# Tag 12

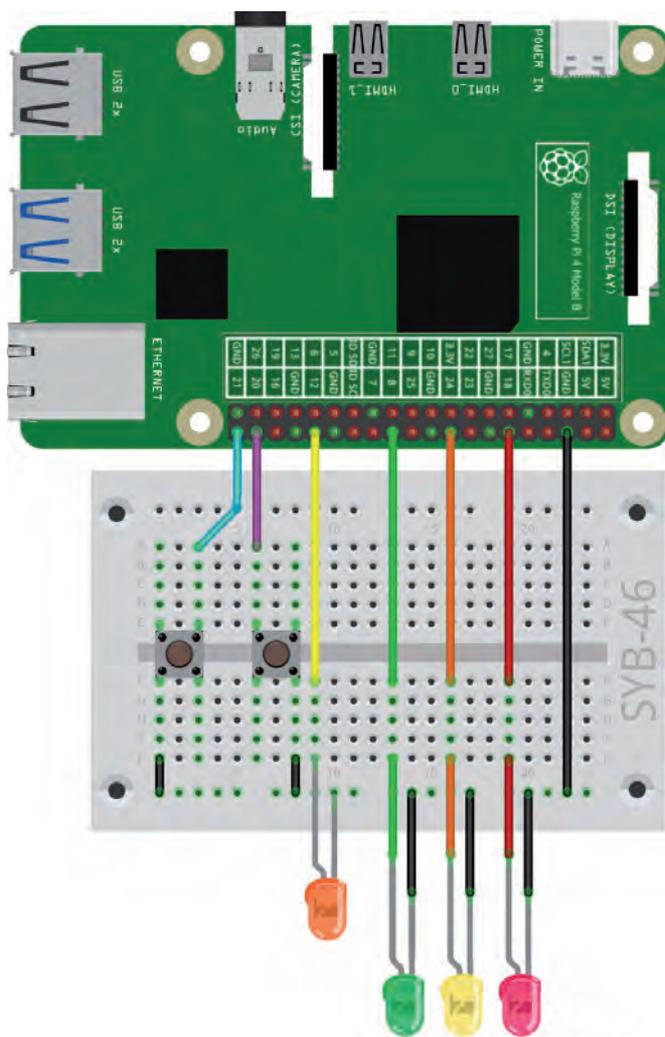
## Heute im Adventskalender

- 1 LED orange mit Vorwiderstand

## Lauflicht und zusätzliche Beleuchtung

Eine zusätzliche LED wird über einen Taster wie mit einem Lichtschalter ein- und ausgeschaltet. Der erste Tastendruck schaltet die LED ein, der zweite wieder aus. Der zweite Taster schaltet auf ähnliche Weise das Lauflicht ein und 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, 2 Taster, 2 Drahtbrücken, 13 GPIO-Verbindungskabel

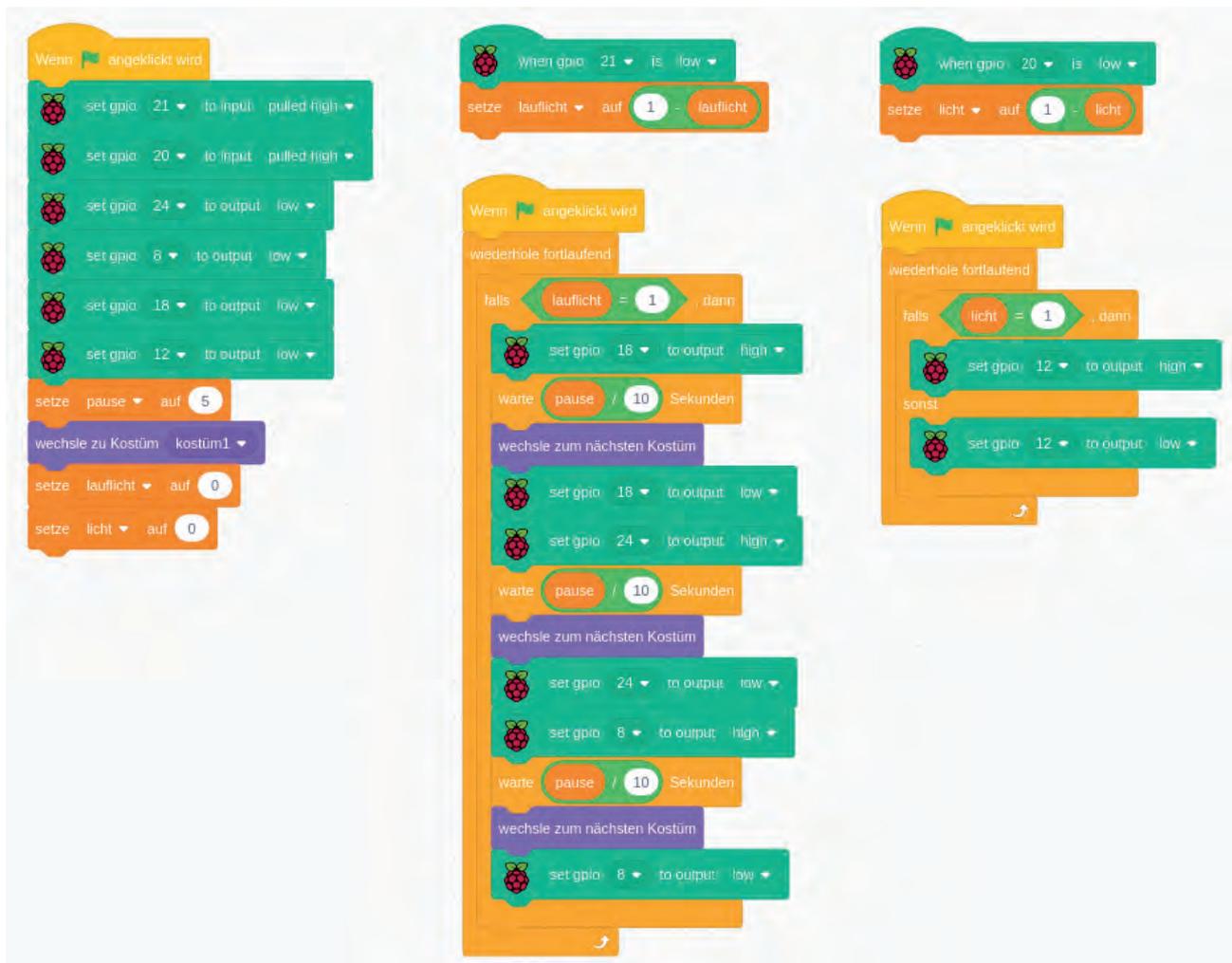


fritzing Vier LEDs und zwei Taster.

## Das Programm

Das Programm 12krippe06 basiert auf dem Programm des vorherigen Tags, enthält aber auch einige neue Programmblöcke.

# Raspberry-Pi-Adventskalender



Das Programm 12krippe06 lässt drei LEDs als Lauflicht und eine einzelne LED über zwei Taster leuchten.

Bei Klick auf das grüne Fähnchen werden alle Pins für die Taster und LEDs initialisiert, alle LEDs sind zunächst ausgeschaltet.

Das Programm enthält zwei neue Variablen, *licht* und *lauflicht*, die die Schaltzustände für das einzelne Licht und das Lauflicht festlegen. Dabei bedeutet 0 „ausgeschaltet“ und 1 „eingeschaltet“. Beide Variablen werden am Anfang auf 0 gesetzt.

Drückt man nun einen der Taster, wird die entsprechende Variable *licht* oder *lauflicht* umgeschaltet. Der Block *setze lauflicht auf 1-lauflicht* verwendet eine simple mathematische Formel, die aus einer 0 eine 1 sowie aus einer 1 eine 0 macht.

Eine Endlosschleife fragt in jedem Durchlauf den Wert der Variablen *lauflicht* ab. Dieser Programmblock startet erneut mit Klick auf das grüne Fähnchen. Hier braucht man keine Verzögerung, um auf die Initialisierung der GPIO-Pins zu warten, da diese sehr schnell geht und das Lauflicht erst startet, nachdem ein Taster gedrückt wurde. Hat die Variable *lauflicht* den Wert 1 – wurde also der Taster mindestens ein erstes Mal gedrückt –, startet das Lauflicht einen Zyklus, bei dem am Ende wieder alle drei LEDs ausgeschaltet sind. Solange die Variable *lauflicht* immer noch auf 1 steht, läuft das Lauflicht weiter. Wurde der Taster zwischenzeitlich ein zweites Mal gedrückt und *lauflicht* wieder auf 0 gesetzt, passiert nichts. Die LEDs bleiben ausgeschaltet.

Ein Druck auf den anderen Taster setzt die Variable *licht* auf 1. Auch hier fragt eine Endlosschleife ständig den Wert ab. Steht diese Variable auf 1, wird die LED an Pin 12 eingeschaltet, wenn nicht, wird sie ausgeschaltet. Hier müssen beide Zustände beachtet werden, da die LED im Gegensatz zum Lauflicht nicht von allein ausgeht.

# Tag 13

## Heute im Adventskalender

- 2 GPIO-Verbindungskabel

## Lauflicht mit Variablen

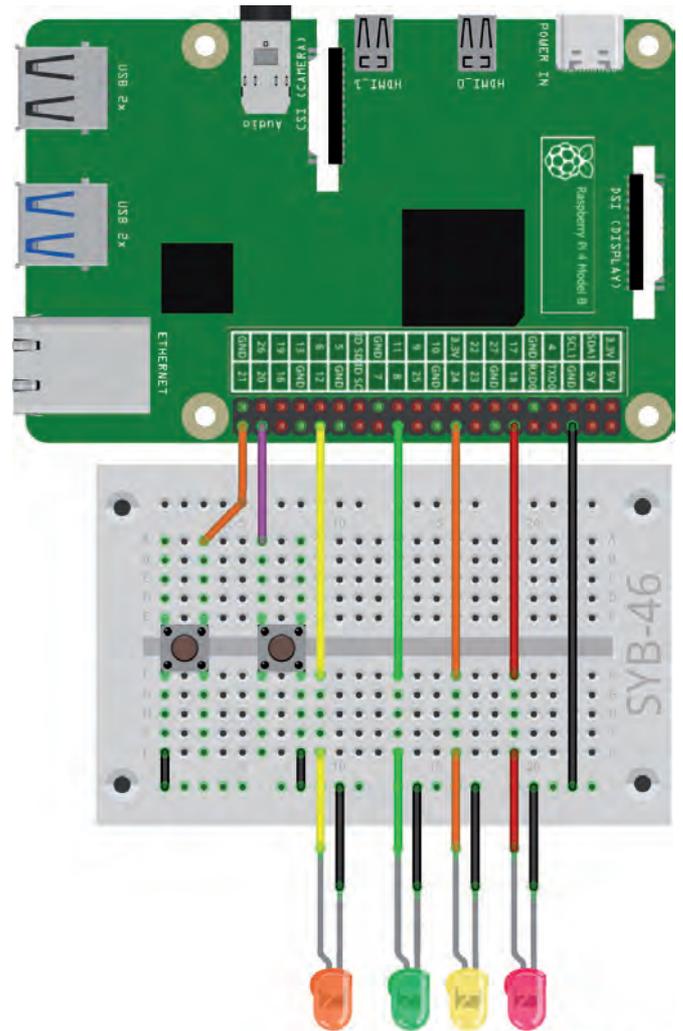
Die orangefarbene LED beleuchtet die Stalllaterne. Mit den zusätzlichen GPIO-Verbindungskabeln kann diese auf die Krippe gesteckt und mit dem Steckbrett verbunden werden.

**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, 2 Taster, 2 Drahtbrücken, 15 GPIO-Verbindungskabel

## Das Programm

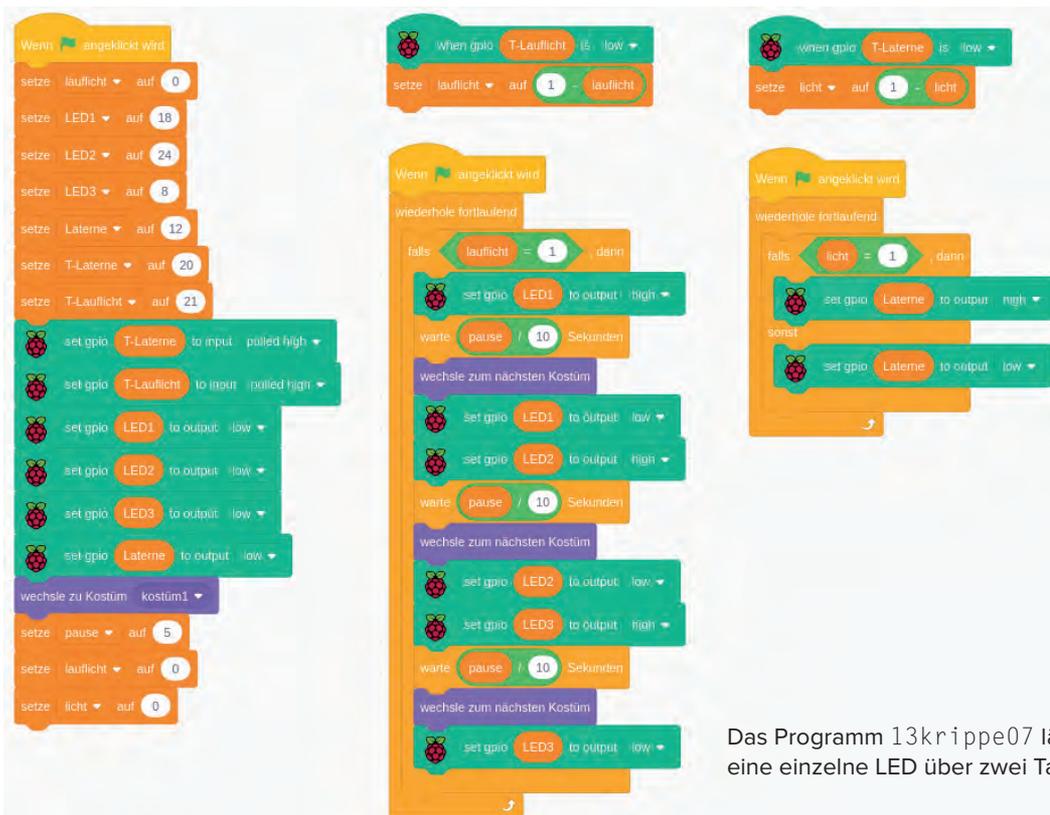
Das Programm 13krippe07 funktioniert genau so wie das Programm des vorherigen Tags, jedoch mit einem wichtigen Unterschied: Um Fehler bei der Nummerierung der GPIO-Pins zu vermeiden und das Programm übersichtlicher und auch flexibler für Umbauten zu machen, werden die Pinnummern in Variablen gespeichert. Die drei Variablen *LED1*, *LED2* und *LED3* bezeichnen die LEDs auf dem Lauflicht, die Variable *Laterne* die LED auf der Laterne im Stall. Die beiden Taster werden als *T-Lauflicht* und *T-Laterne* bezeichnet.

Das Programm funktioniert wie das vom vorherigen Tag. Sollen die LEDs oder Taster an anderen GPIO-Pins angeschlossen werden, brauchen Sie nur eine andere Pinnummer in den Variablenzuweisungen am Anfang des Programms einzutragen.



fritzing

Vier LEDs und zwei Taster.



Das Programm 13krippe07 lässt drei LEDs als Lauflicht und eine einzelne LED über zwei Taster leuchten.

# Tag 14

## Heute im Adventskalender

- 1 GPIO-Verbindungskabel

## Listen in Scratch

Das Programm für das Lauflicht ist bei drei LEDs noch übersichtlich. Jede weitere LED macht es länger und erfordert zusätzlichen Aufwand, obwohl sich das Grundprinzip immer wiederholt: LED einschalten, warten, Kostüm wechseln, LED ausschalten, nächste LED ...

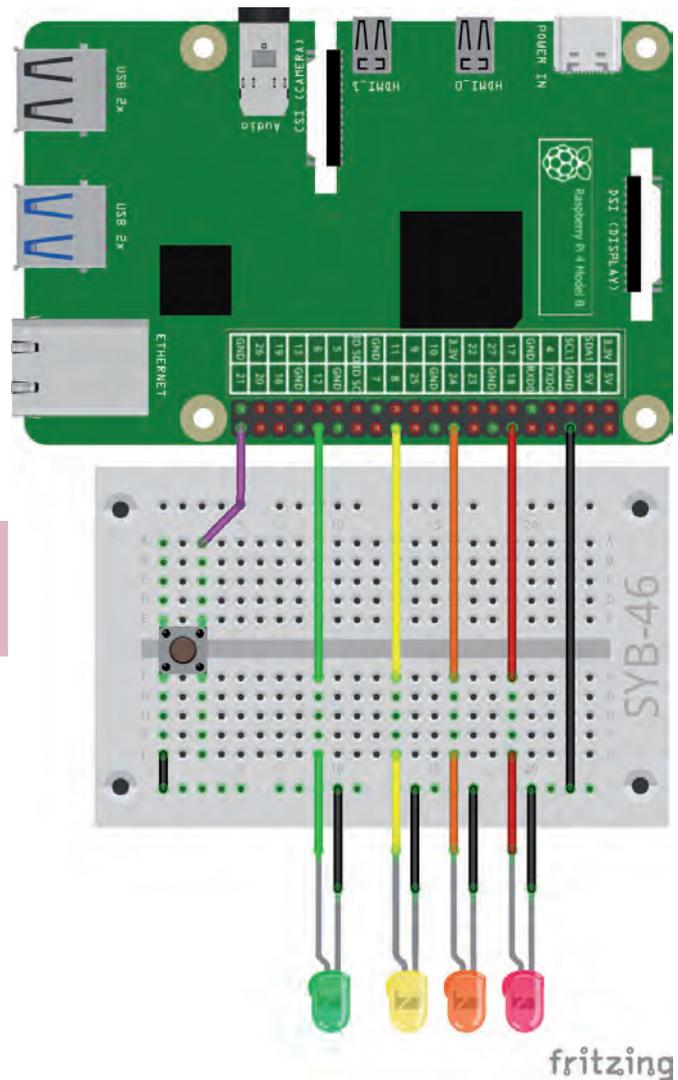
Listen in Scratch erleichtern derartige sich wiederholende Aufgaben. In solchen Listen können mehrere Werte gespeichert und über die Position innerhalb der Liste adressiert werden.

**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 Taster, 1 Drahtbrücke, 14 GPIO-Verbindungskabel

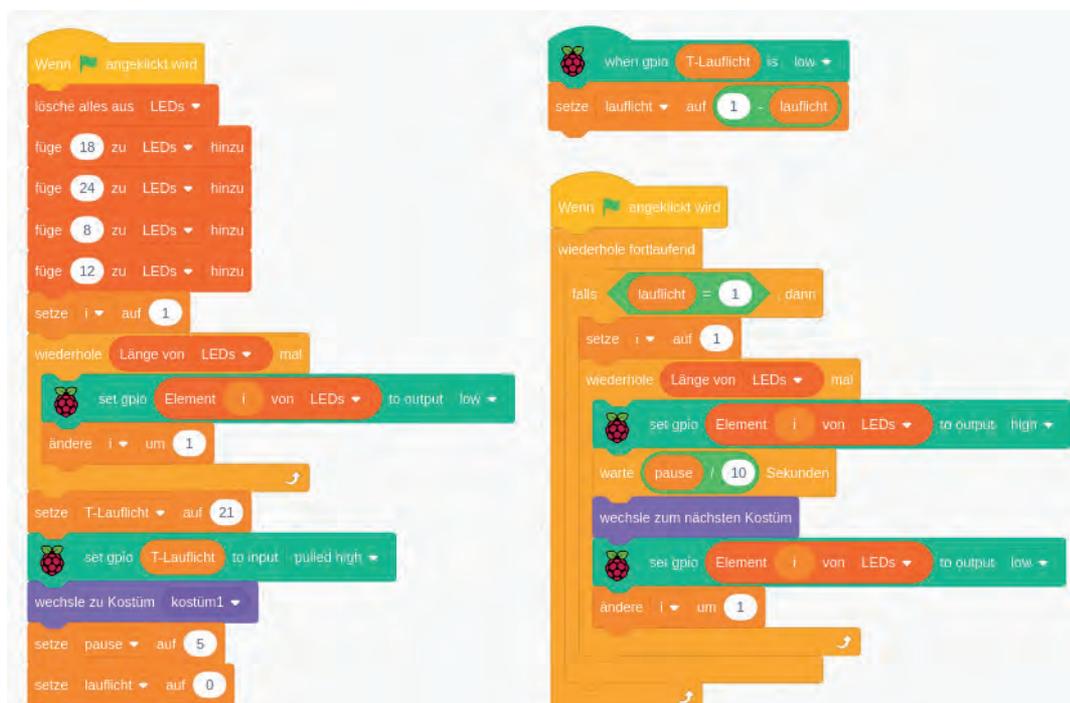
## Das Programm

Das Programm 14krippe08 verwendet eine Liste für die vier LEDs, die sich leicht auch auf mehr LEDs erweitern lässt. Die Stalllaterne fehlt, da diese LED für das Lauflicht verwendet wird.

Erstellen Sie für dieses Programm eine Liste mit dem Namen *LEDs*. Klicken Sie dazu in der Blockpalette *Variablen* auf *Neue Liste*. Beim Anlegen der ersten Liste werden die Blöcke zur Verarbeitung von Listen freigeschaltet.



Lauflicht mit vier LEDs und Taster.



Das Programm 14krippe08 lässt vier LEDs als Lauflicht leuchten.

Zusätzlich wird eine neue Variable *i* angelegt, ein einfacher Zähler, der an verschiedenen Stellen im Programm verwendet wird.



Das Programm startet mit einem Klick auf das grüne Fähnchen. Im ersten Schritt werden alle Elemente der Liste entfernt, falls aus einem früheren Programmlauf dort noch Werte vorhanden sind. Anschließend werden die Nummern der für die vier LEDs verwendeten GPIO-Pins in den ersten vier Listenpositionen gespeichert.

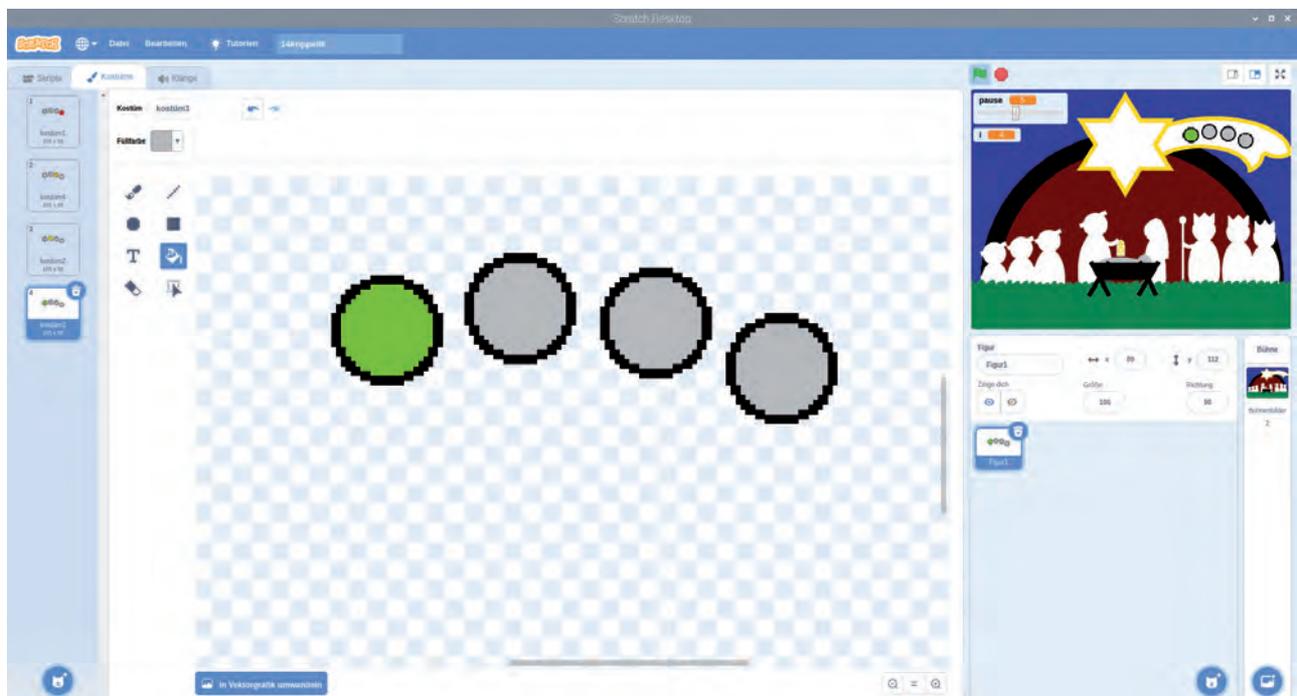
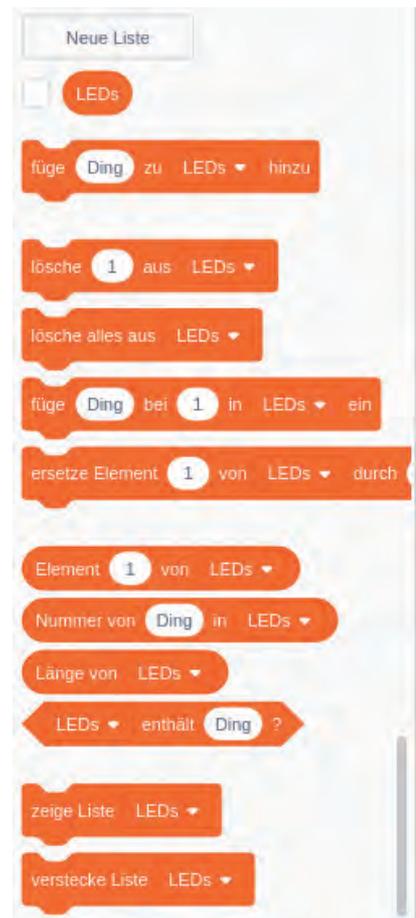
Eine Schleife läuft viermal durch. Dabei adressiert sie die vier GPIO-Pins für die LEDs und schaltet sie aus. Durch Verwendung des Blocks *Länge von ...* anstatt der Zahl 4 ist das Programm sehr leicht für längere Listen anpassbar.

Die Variable *i* steht für eine Zahl im Bereich von 1 bis 4. Somit gibt der Block *Element i von LEDs* die Nummer des jeweiligen GPIO-Pins an, die nacheinander als Ausgänge initialisiert und ausgeschaltet werden.

Die Initialisierung und die Abfrage des Tasters laufen wie im Programm des vorherigen Tags ab.

Ist das Lauflicht über die Variable *lauflicht* eingeschaltet, läuft wieder eine Schleife, die in jedem Durchlauf eine LED einschaltet, die festgelegte Pausenzeit wartet, das Kostüm wechselt und die LED wieder ausschaltet. Der Zähler *i* wählt die zu schaltende LED aus.

Nach der letzten LED leuchtet erneut die erste, da *i* vor dem Beginn der Schleife wieder auf 1 gesetzt wird.



Das Programm verwendet vier Kostüme für die vier LEDs.

Wenn Sie das Programm selbst erstellen, löschen Sie alle Kostüme der einzigen Figur bis auf das erste. Fügen Sie eine weitere LED hinzu und duplizieren Sie das Kostüm dreimal. Legen Sie dann noch die vier LED-Farben fest.

## Heute im Adventskalender

- 1 RGB-LED mit Vorwiderständen

### RGB-LEDs

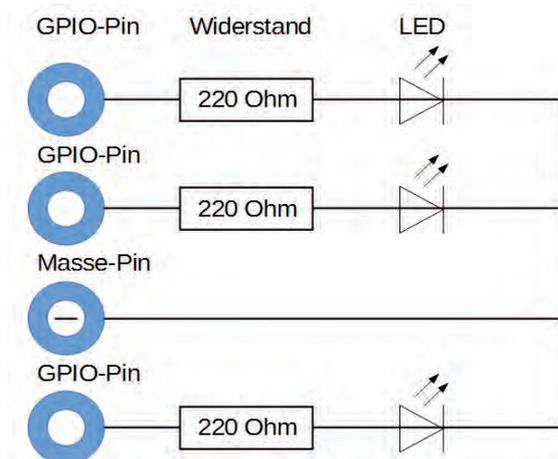
Eine normale LED leuchtet immer nur in einer Farbe. Die im Adventskalender verwendeten RGB-LEDs können wahlweise in mehreren Farben leuchten. Bei ihnen sind im Prinzip drei verschiedenfarbige LEDs in einem transparenten Gehäuse eingebaut. Jede dieser drei LEDs hat eine eigene Anode, über die sie mit einem GPIO-Pin verbunden wird. Die Kathode, die mit der Masseleitung verbunden wird, ist nur einmal vorhanden. Deshalb hat eine RGB-LED vier Anschlussdrähte.



Anschlusspins einer RGB-LED.

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

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



Interner Schaltplan für eine RGB-LED mit drei Vorwiderständen.

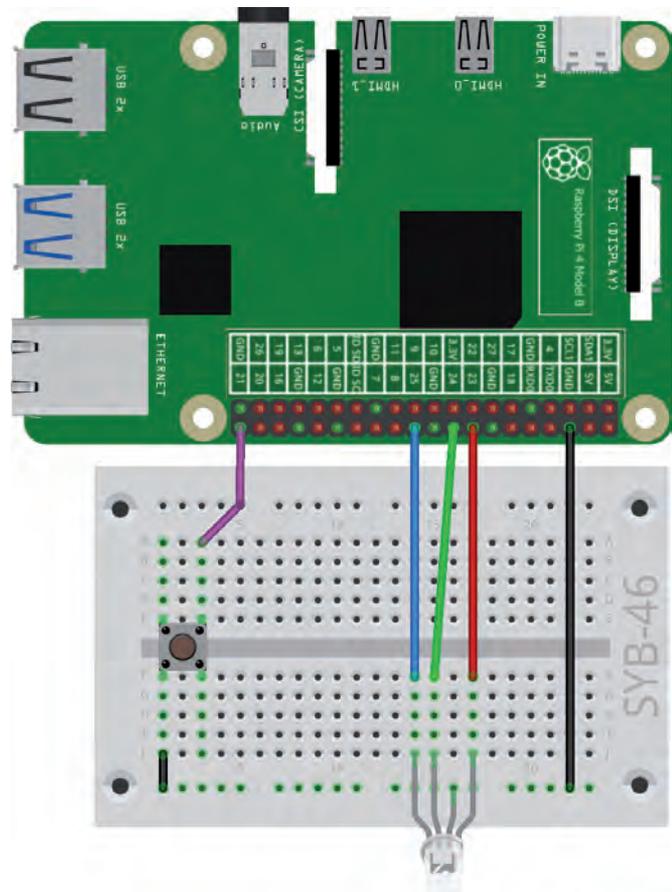
### RGB-Farbspiele

Das Experiment von heute lässt eine RGB-LED automatisch nacheinander in verschiedenen Farben aufleuchten.

**Bauteile:** 1 Steckbrett SYB-46, 1 RGB-LED mit Vorwiderständen, 1 Taster, 1 Drahtbrücke, 5 GPIO-Verbindungskabel

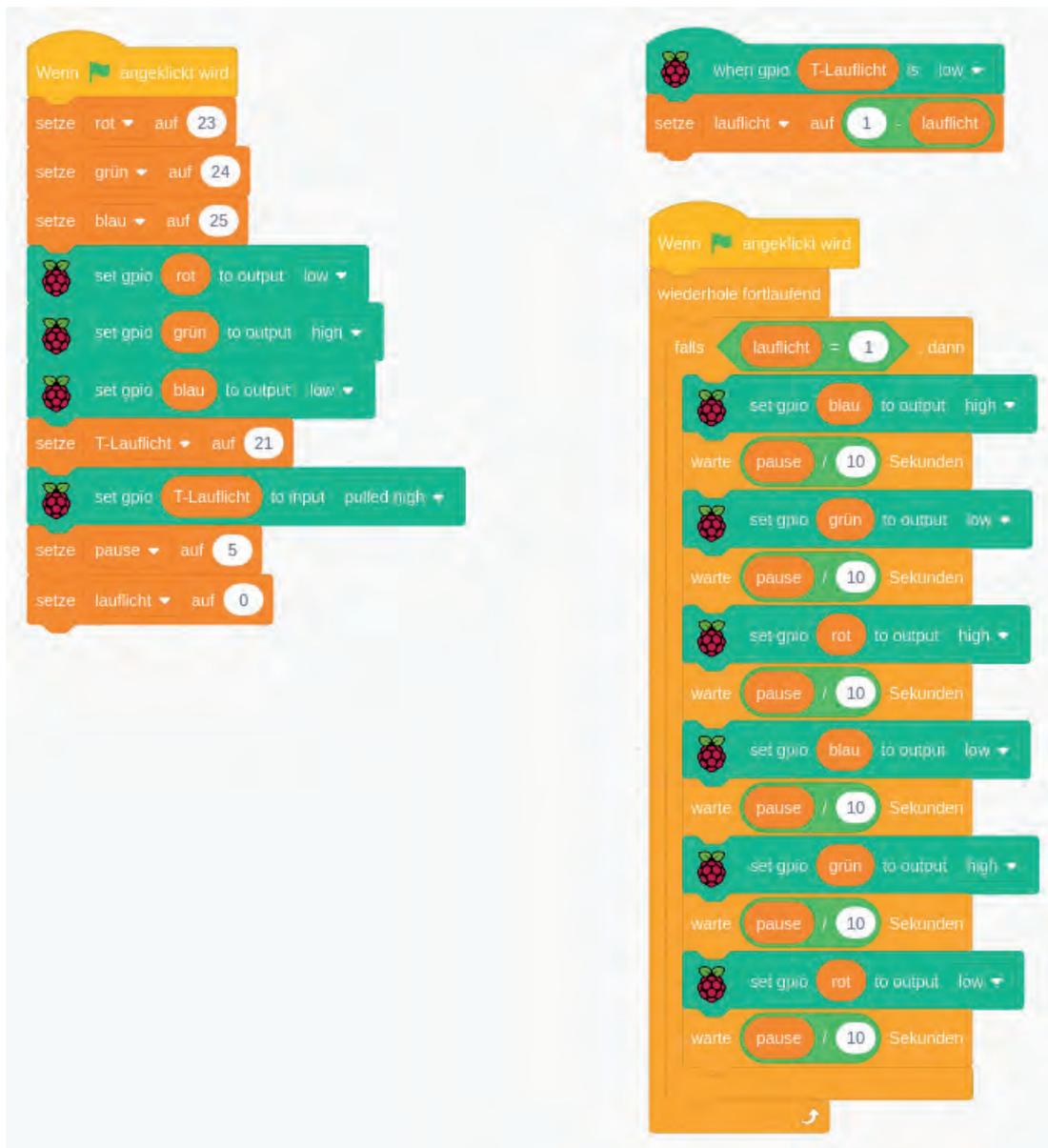
### Das Programm

Das Programm 15rgbled01 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 drei unterschiedlichen Mischfarben.



fritzing

RGB-LED an den GPIO-Pins 23, 24, 25. Der Masse-Pin ist mit der Masseleiste des Steckbretts verbunden.



Das Programm 15rgb1ed01 steuert eine RGB-LED.

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 über die Variablen *rot*, *grün* und *blau* initialisiert wurden, werden der Reihe nach verschiedene Pins ein- und ausgeschaltet. Die Wartezeit zwischen den Schaltvorgängen kann vom Benutzer interaktiv über den Regler auf der Scratch-Bühne verändert werden.

## Heute im Adventskalender

- 1 GPIO-Verbindungskabel

## RGB-LED blinkt zufällig

Im Experiment von heute blinkt eine RGB-LED in zufälligen bunten Farben, nachdem der Taster gedrückt wurde. Ein zweites Drücken schaltet das Blinken wieder aus.

Der Schaltungsaufbau entspricht dem des vorherigen Tags.

**Bauteile:** 1 Steckbrett SYB-46, 1 RGB-LED mit Vorwiderständen, 1 Taster, 1 Drahtbrücke, 5 GPIO-Verbindungskabel

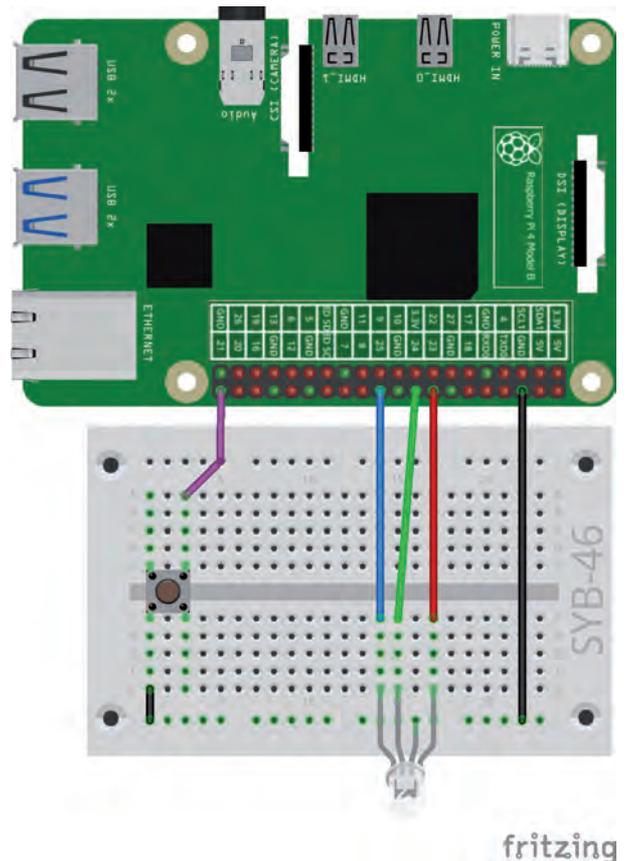
## Das Programm

Das Programm 16rgb1ed02 verwendet eine Liste für die drei GPIO-Pins der RGB-LED. Eine Schleife initialisiert die Pins als Eingänge und schaltet alle aus. Anschließend wird noch der Taster, wie bereits bekannt, initialisiert.

Ein Tastendruck soll das Blinken der RGB-LED einschalten, ein weiterer Tastendruck schaltet es wieder aus. Dazu nutzen wir die gleiche Technik wie in den vorhergehenden Programmen.

In jedem Schleifendurchlauf wird eine zufällig gewählte Farbkomponente der RGB-LED eingeschaltet. Dazu wird ein Block *Zufallszahl von 1 bis 3* verwendet, der im Zahlenfeld eines Blocks *Element ... von LEDs* liegt und damit ein zufälliges Element aus der Liste *LEDs* aussucht.

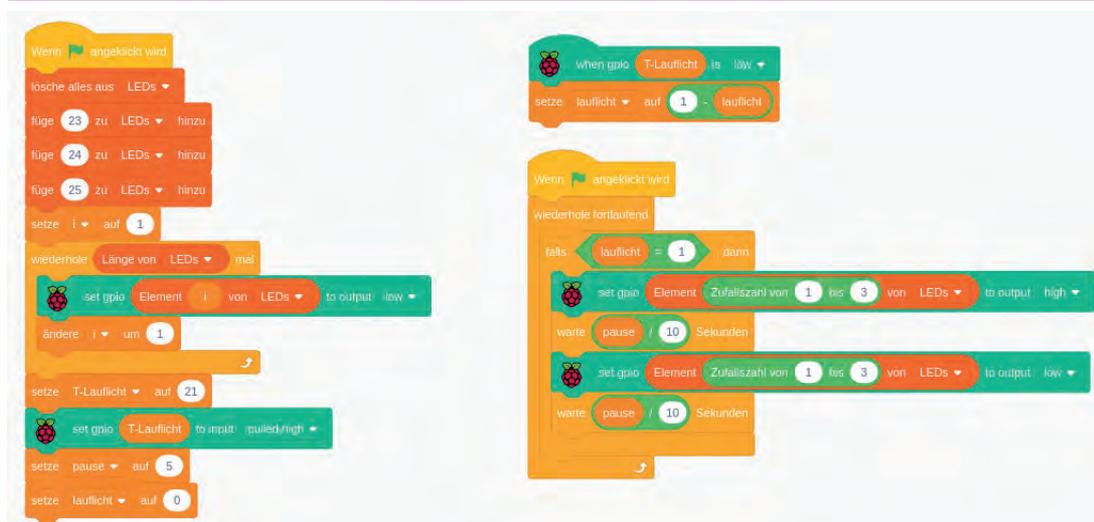
Nach einer wieder über einen Regler einstellbaren Pause wird eine zufällig gewählte Farbkomponente der RGB-LED ausgeschaltet. Durch den verwendeten Algorithmus kann es passieren, dass die RGB-LED kurzfristig ganz ausgeschaltet wird oder zweimal hintereinander die gleiche Farbe zeigt.



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

## 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.



Das Programm 16rgb1ed02 steuert eine RGB-LED über einen Taster.

## Heute im Adventskalender

- 1 GPIO-Verbindungskabel

## Scratch 3 OneGPIO-Erweiterung

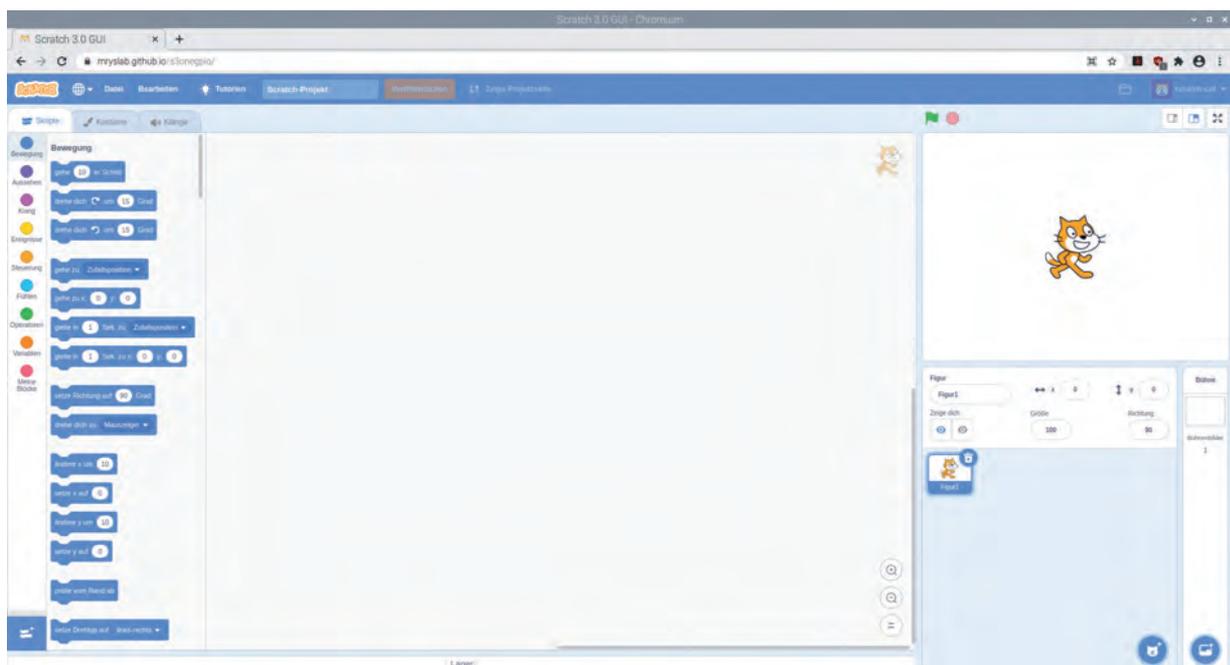
Die Möglichkeiten der mitgelieferten GPIO-Erweiterung für Scratch sind begrenzt. Dabei bietet die GPIO-Schnittstelle auf dem Raspberry Pi deutlich mehr. Um weitere Funktionen in Scratch 3 zu nutzen, sind andere Erweiterungen erforderlich. Wir verwenden für die folgenden Projekte die **Scratch 3 OneGPIO**-Erweiterung, die neben dem Raspberry Pi auch einige andere Hardwareplattformen unterstützt.



- 1 Die Erweiterung **Scratch 3 OneGPIO** wird über die Python-Paketverwaltung `pip` installiert. Prüfen Sie als Erstes, ob eine unterstützte `pip`-Version installiert ist. Öffnen Sie dazu ein Kommandozeilenfenster und geben Sie dort ein: `pip3 --version`
- 2 Wird hier eine Version älter als `pip 20.x` angezeigt, starten Sie ein Update:  
`sudo pip3 install --upgrade pip`
- 3 Installieren Sie anschließend die Erweiterung **Scratch 3 OneGPIO**: `sudo pip3 install s3r-extend`
- 4 Jedes Mal, wenn Sie Scratch mit dieser Erweiterung nutzen möchten, starten Sie zuerst den **OneGPIO-Server**: `s3r`
- 5 Lassen Sie das Kommandozeilenfenster geöffnet, solange Sie die Erweiterung **Scratch 3 OneGPIO** nutzen. Erst wenn Sie Scratch beenden, können Sie den **OneGPIO-Server** mit der Tastenkombination `Strg+C` beenden und dann das geöffnete Kommandozeilenfenster schließen.
- 6 Die Erweiterung **Scratch 3 OneGPIO** läuft nicht mit dem offline auf dem Raspberry Pi installierten Scratch, sondern nur mit einer speziell erweiterten Onlineversion. Starten Sie dazu den auf dem Raspberry Pi installierten Chromium-Browser und besuchen Sie diese Seite: [mryslab.github.io/s3onegpio](https://mryslab.github.io/s3onegpio)
- 7 Scratch 3 funktioniert im Browser genau so wie das installierte Scratch auf dem Raspberry Pi.

```
pi@raspberrypi: ~/pigpio-master
Datei Bearbeiten Reiter Hilfe
pi@raspberrypi:~/pigpio-master $ s3r
Only run this script on a Raspberry Pi!
backplane started
Websocket Gateway started
RPi Gateway started
To exit this program, press Control-c
```

`s3r` in einem Kommandozeilenfenster.

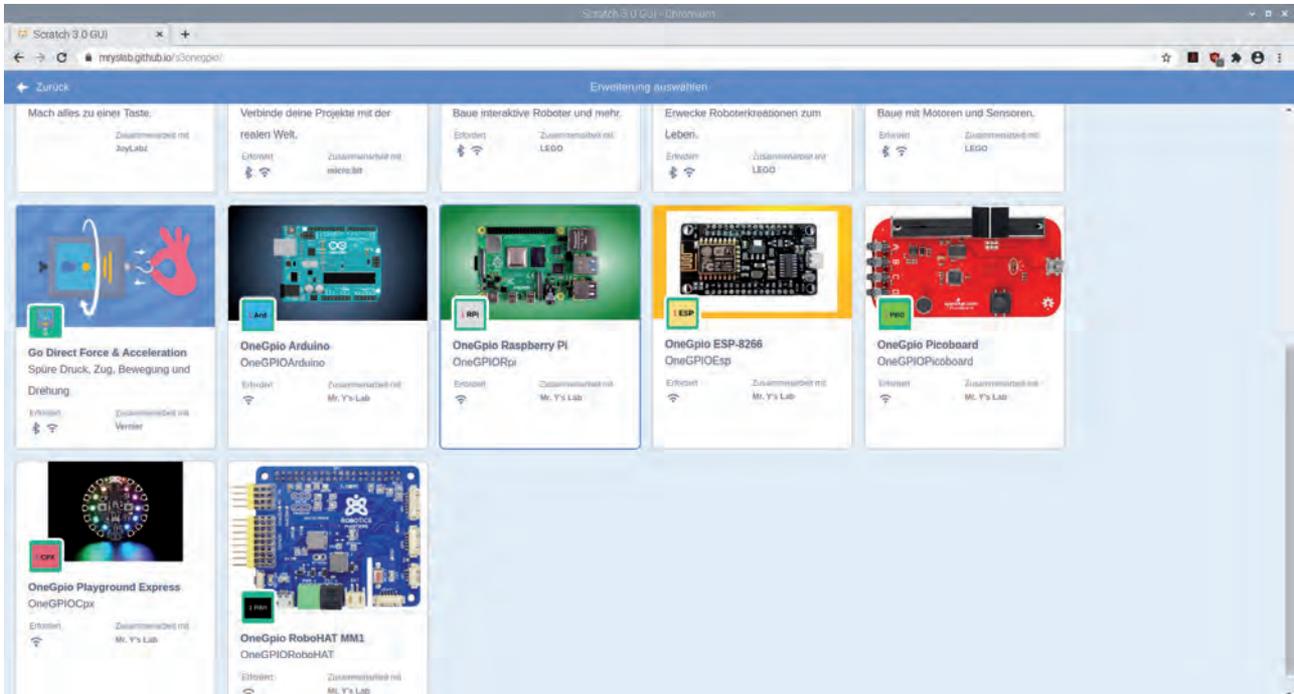


Scratch 3 im Browser auf dem Raspberry Pi.

# Raspberry-Pi-Adventskalender



- 8 Klicken Sie unterhalb der Blockpalette auf das bekannte Symbol *Erweiterung hinzufügen* und wählen Sie die Erweiterung **OneGPIO Raspberry Pi**. Hier werden mehr Erweiterungen angeboten als in der vorinstallierten Offlineversion von Scratch.



Die Erweiterung OneGPIO Raspberry Pi installieren.

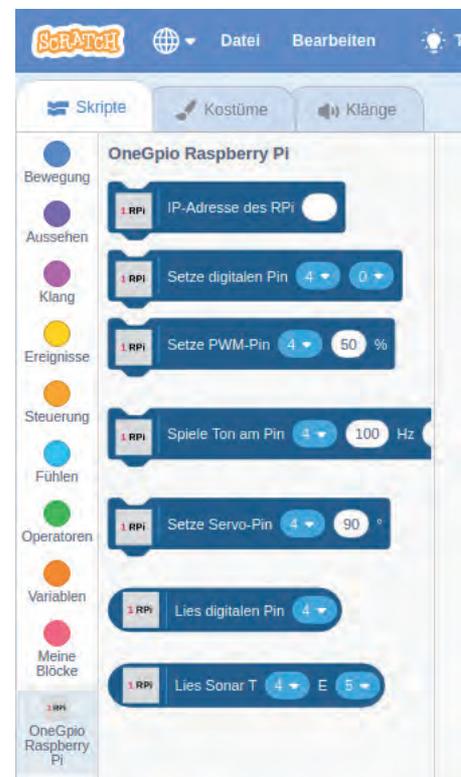
Die Erweiterung liefert neue Blöcke, mit denen sich alle grundlegenden GPIO-Funktionen nutzen lassen, aber auch verschiedene weitere. Sie brauchen die Standard-GPIO-Erweiterung nicht zusätzlich, da sich alle Funktionen daraus auch mit der Erweiterung **OneGPIO Raspberry Pi** programmieren lassen. Allerdings heißen die Blöcke etwas anders und verhalten sich auch leicht unterschiedlich.

## Scratch 3-OneGPIO-Server im Menü

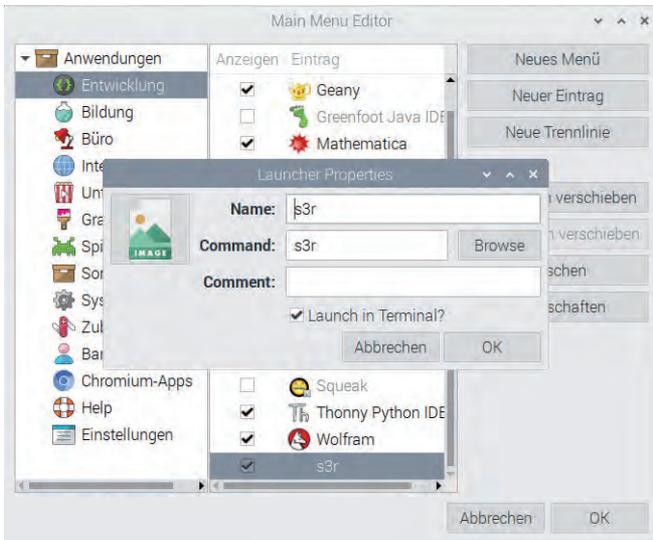
Um nicht jedes Mal die Erweiterung **Scratch 3 OneGPIO** vor dem Aufruf von Scratch über ein Kommandozeilenfenster starten zu müssen, können Sie sich einen Menüpunkt dafür anlegen.



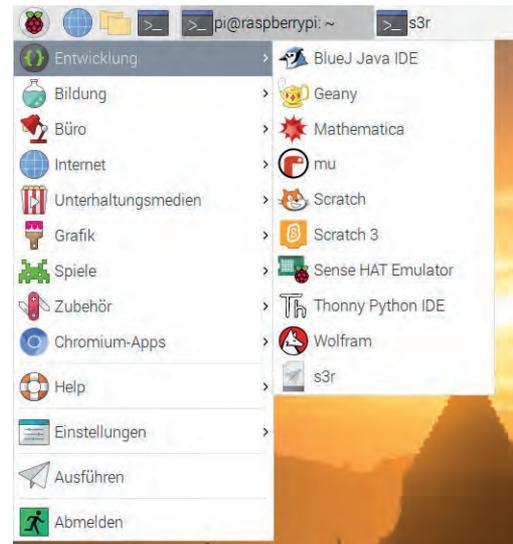
- 1 Der **OneGPIO-Server** für die Erweiterung **Scratch 3 OneGPIO** wird über den Befehl `s3r` gestartet. Wählen Sie im Startmenü unter *Einstellungen* den *Main Menu Editor*.
- 2 Gehen Sie hier in der linken Liste in den Bereich *Entwicklung*. In der rechten Liste werden alle Menüpunkte dieses Untermenüs angezeigt.
- 3 Klicken Sie auf *Neuer Eintrag* und geben Sie im nächsten Fenster in die beiden Felder *Name* sowie *Command* jeweils `s3r` ein. Aktivieren Sie dann noch das Häkchen bei *Launch in Terminal?*.



Die neuen Blöcke der Erweiterung OneGPIO Raspberry Pi.



Menüpunkt für s3r anlegen.



s3r im Startmenü.

- 4 Schließen Sie beide Fenster mit *OK*. Jetzt erscheint *s3r* als Menüpunkt im Startmenü unter *Entwicklung*.

## Lauflicht mit der Scratch 3-OneGPIO-Erweiterung

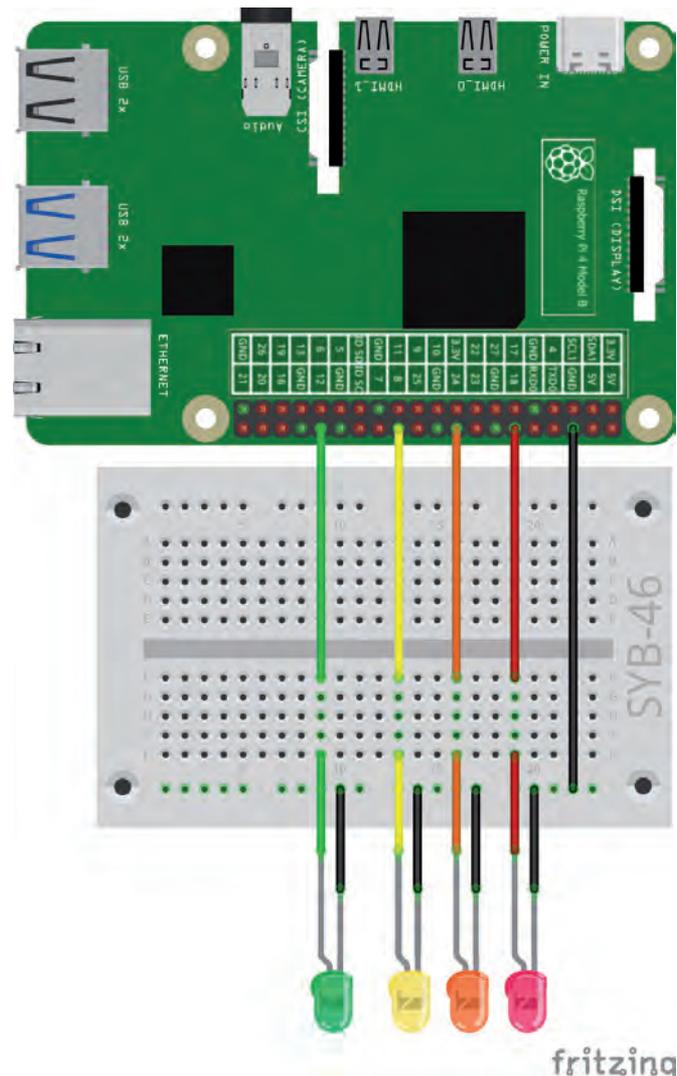
Zum einfachen Umgewöhnen programmieren wir ein bereits bekanntes Lauflicht heute mit der Erweiterung **OneGPIO Raspberry Pi** noch einmal.

**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, 13 GPIO-Verbindungskabel

### Das erste Programm

Das Programm 171auflicht01 lässt die vier LEDs nacheinander aufleuchten. In einer Endlosschleife werden die LEDs nacheinander einzeln ein- und wieder ausgeschaltet. Die Pausenzeit lässt sich auch hier mit einem Schieberegler einstellen.

Der Block *Setze digitalen Pin ...* enthält zwei Auswahlfelder. Im linken Feld wählen Sie die Nummer des GPIO-Pins, im rechten Feld *0* oder *1*, um den Pin aus- oder einzuschalten.



Lauflicht mit vier LEDs.

# Raspberry-Pi-Adventskalender

## Das zweite Programm

Das Programm 171auflicht02 zeigt, wie auch mit der Erweiterung **OneGPIO Raspberry Pi** Listen für GPIO-Nummern verwendet werden können. Die Liste *LEDs* wird beim Programmstart mit den GPIO-Nummern der vier LEDs gefüllt. Eine Schleife lässt die LEDs gemäß der Liste nacheinander blinken. Ziehen Sie dazu den Block *Element ... von LEDs* auf das erste Auswahlfeld im Block *Setze digitalen Pin ...* Dieses ersetzt dann das dort vorgegebene Auswahlfeld.



Das Programm 171auflicht02 lässt vier LEDs mithilfe einer Liste als Lauflicht laufen.



Das Programm 171auflicht01 lässt vier LEDs als Lauflicht laufen.

## Heute im Adventskalender

- 1 GPIO-Verbindungskabel

### LEDs dimmen

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

Mit einem Trick erreicht man es dennoch, die Helligkeit einer LED an einem GPIO-Pin 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 *Aus* (0 V) und *Ein* (+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.



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

#### Warum 50 Hertz die ideale Frequenz für das Dimmen von LEDs mit PWM ist

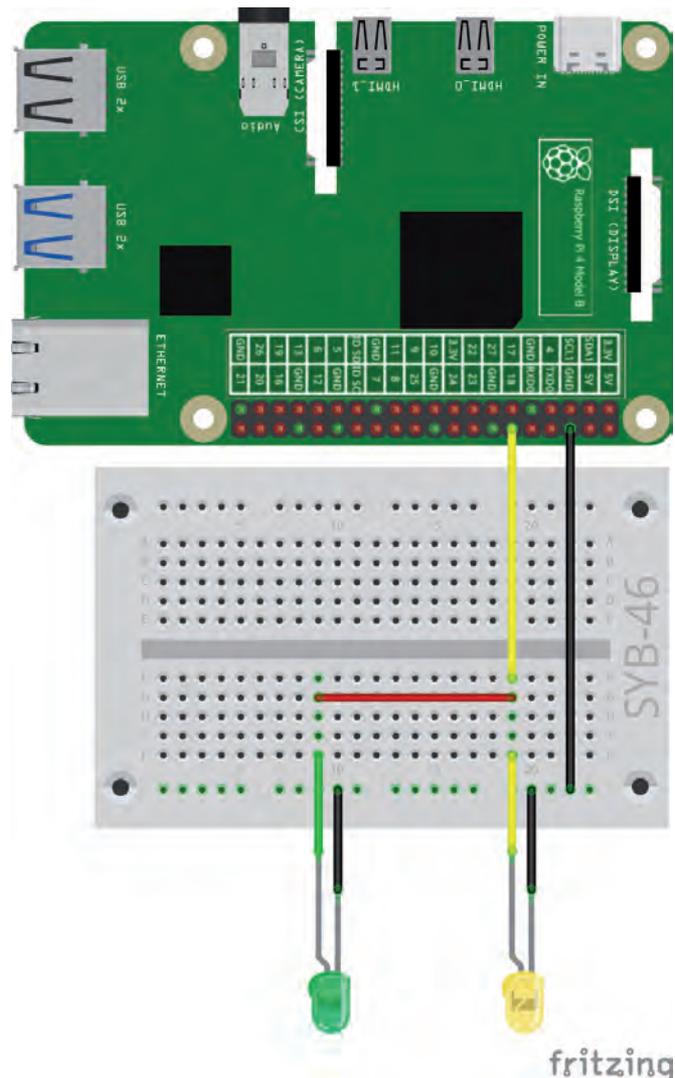
Das menschliche Auge nimmt Lichtwechsel schneller als 20 Hertz nicht mehr wahr. Da das Wechselstromnetz in Europa eine Frequenz von 50 Hertz nutzt, blinken viele Beleuchtungskörper mit dieser Frequenz, die vom Auge nicht wahrgenommen wird. Blinkt eine LED mit mehr als 20 Hertz, aber weniger als 50 Hertz,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

**Bauteile:** 1 Steckbrett SYB-46, 1 LED grün mit Vorwiderstand, 1 LED gelb mit Vorwiderstand, 6 GPIO-Verbindungskabel, 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 Drahtbrücke aus Schaltdraht verbunden.

### Das Programm

Das Programm `18_pwm` dimmt die beiden LEDs anhand eines interaktiv einstellbaren Werts.



Zwei LEDs werden über einen GPIO-Pin gedimmt.

## Raspberry-Pi-Adventskalender



Das Programm 18pwm dimmt die beiden LEDs.

Der Block *Setze PWM-Pin ... %* bietet eine Funktion zur Steuerung eines PWM-Signals an.

Eine Endlosschleife fragt die interaktiv über einen Schieberegler einstellbare Variable *pwm* ab und setzt den GPIO-Pin 18 anschließend auf den neuen Wert.

Stellen Sie den Reglerbereich der Variablen auf *Min: 0* und *Max: 100*. In diesem Bereich erwartet die Erweiterung **OneGPIO Raspberry Pi** die Werte für PWM.

Die LEDs leuchten mit der eingestellten Helligkeit. Anschließend wartet das Programm 0,1 Sekunden, da es bei zu schnellen Abfragen des bei dieser Erweiterung im Hintergrund immer mitlaufenden OneGPIO-Websocket-Gateways, das für die Kommunikation zwischen Scratch im Browser und den GPIO-Pins des Raspberry Pi zuständig ist, zu Kommunikationsproblemen kommen kann und sich das Programm „überschlägt“. Die Browserversion von Scratch kann auch mit Kommazahlen arbeiten, was bei der vorinstallierten Scratch-3-Variante im Raspberry Pi OS nur sehr eingeschränkt funktioniert. Danach startet die Endlosschleife neu und fragt den neuen Wert der Variablen *pwm* ab. So lässt sich die Helligkeit der beiden LEDs interaktiv einstellen, während das Programm läuft.

## Heute im Adventskalender

- 1 GPIO-Verbindungskabel

## Farbmischung mit RGB-LED

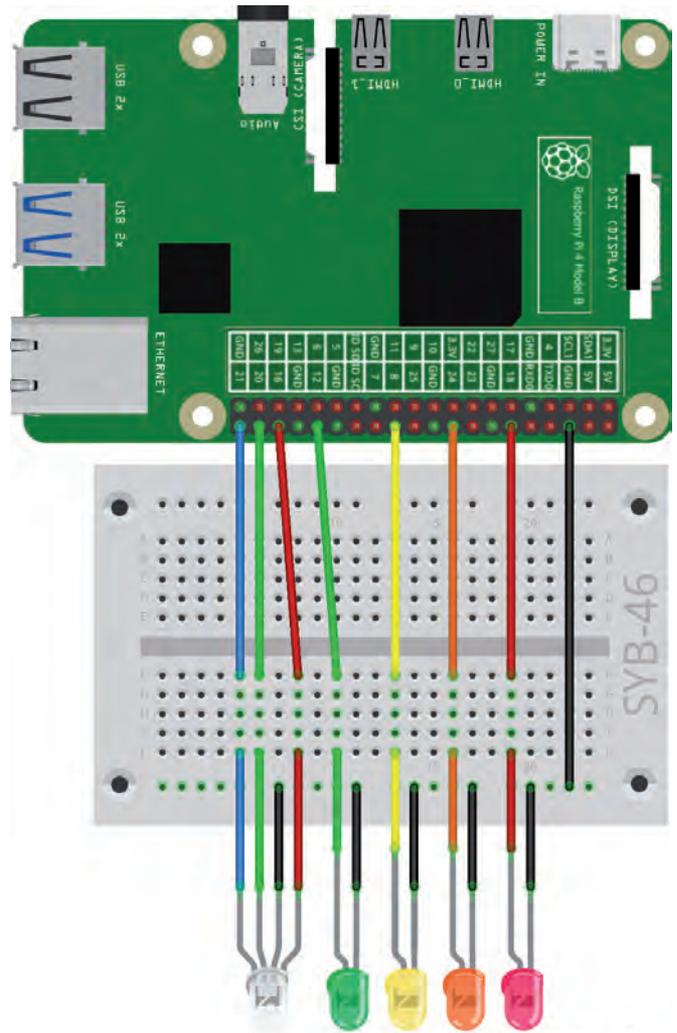
Das Experiment von heute mischt auf einer RGB-LED über PWM verschiedene Farben. Zusätzlich leuchten vier LEDs, die per PWM ebenfalls unabhängig voneinander einstellbar sind.

**Bauteile:** 1 Steckbrett SYB-46, 1 RGB-LED mit Vorwiderständen, 1 LED grün mit Vorwiderstand, 1 LED gelb mit Vorwiderstand, 1 LED orange mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 20 GPIO-Verbindungskabel

## Das Programm

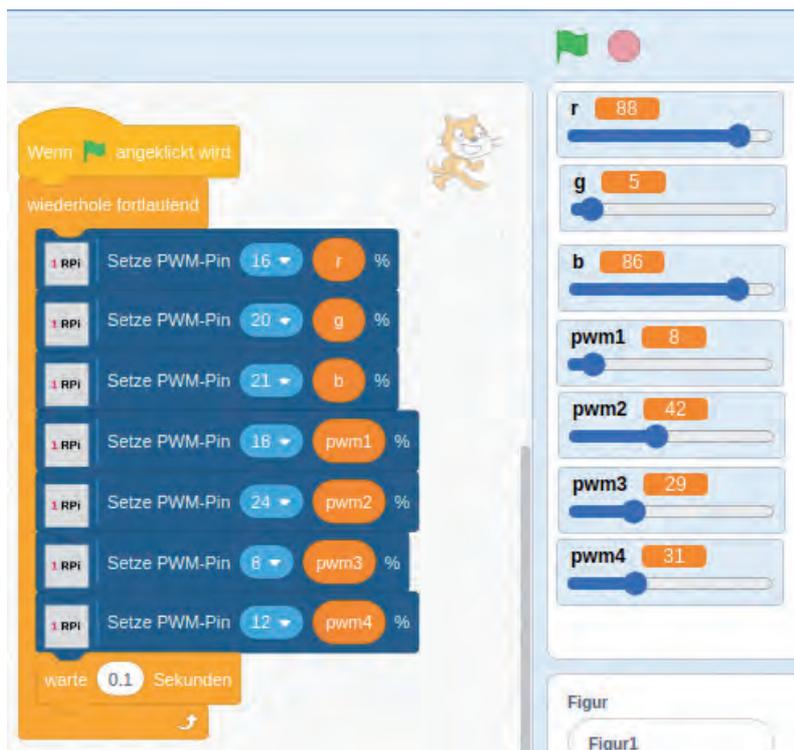
Jede Farbkomponente einer RGB-LED lässt sich über PWM dimmen. Dadurch sind beliebige Mischfarben möglich. Wenn zum Beispiel Rot zu 100% leuchtet und Grün nur zu 50%, ergibt das Orange als Mischfarbe. Das Programm `19rgbpwm` enthält sieben Variablen, die über Regler eingestellt werden können und die drei Farbkomponenten der RGB-LED sowie die vier einfarbigen LEDs unabhängig steuern.

Auch diesmal fragt eine Endlosschleife ständig die eingestellten Werte der Variablen  $r$ ,  $g$ ,  $b$  für die RGB-LED sowie  $pwm1$ ,  $pwm2$ ,  $pwm3$  und  $pwm4$  ab und setzt die PWM-Signale für die LEDs entsprechend.



fritzing

RGB-LED und vier einzelne LEDs auf der Weihnachtskrippe.



Sieben einstellbare Variablen mischen Farben auf der RGB-LED und den einfarbigen LEDs.

## Heute im Adventskalender

- 1 Servo mit Doppelarmgelenk und Befestigungsschraube
- 1 Pinleiste 3-polig für Servo

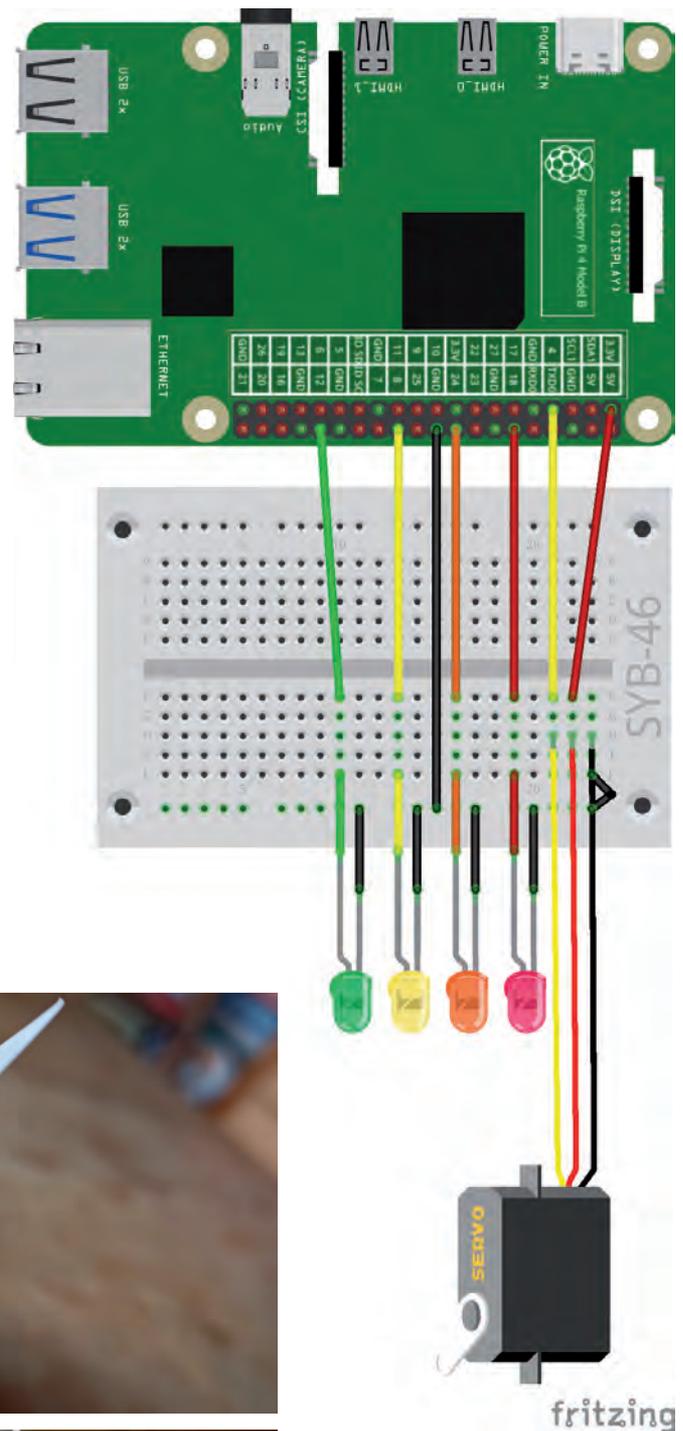
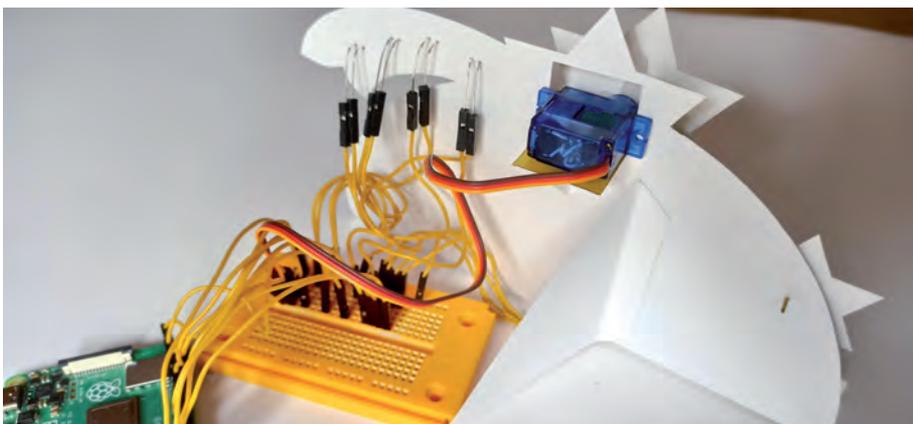
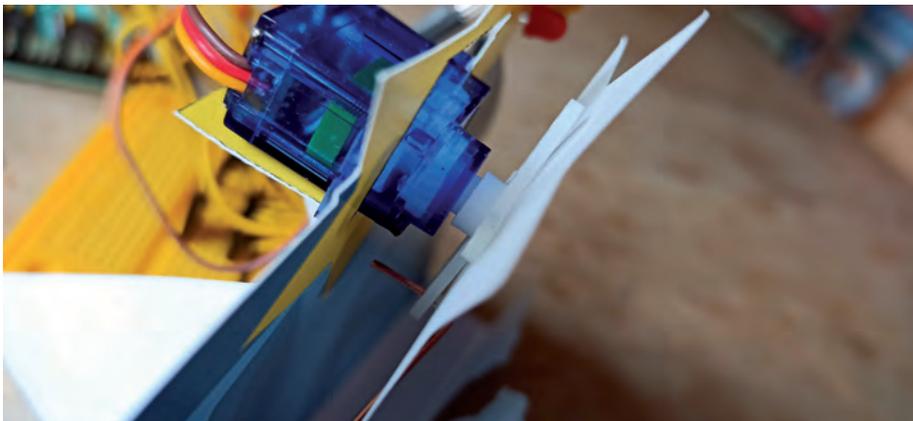
## Servo

Ein Servo ist ein kleiner Motor mit einem Arm, an dem Gegenstände befestigt werden können. Mit einem Steuerungssignal kann er um einen bestimmten Winkel gedreht werden. Die meisten Servos haben einen Drehbereich von 180 Grad, drehen also nicht, wie klassische Motoren, nacheinander viele Runden in Folge, sondern können mit dem Steuersignal in eine bestimmte Position gebracht werden.

## Servo steuern

Schrauben Sie den Doppelarm mit der kleinsten der mitgelieferten Schrauben so auf dem Servo fest, dass er im Ruhezustand in der Längsrichtung des Servos steht. Bauen Sie den Servo oben an der vorgesehenen Stelle in die Krippe ein und kleben Sie ihn mit einem Klebeband fest. Kleben Sie mit einem weiteren Stück Klebeband den Stern auf dem Arm des Servos fest.

Der Servo wird in unseren Experimenten über GPIO-Verbindungskabel und eine Pinleiste auf dem Steckbrett am Raspberry Pi angeschlossen. Diese Pin-



Vier LEDs und ein Servo. Die Drahtbrücke an der Masseleitung des Servos ist abgeknickt dargestellt. Die schwarze Masseleitung des Servos (im Bild unten rechts) ist über eine kurze Drahtbrücke mit der Masseschiene des Steckbretts verbunden. Die Drahtbrücke an der Masseleitung des Servos ist abgeknickt dargestellt, damit man sie deutlicher erkennt.

leiste passt genau in das Anschlusskabel des Servos. Servos haben drei Anschlussleitungen. Die rote und die schwarze Leitung dienen der Stromversorgung, auf der gelben Leitung empfängt der Servo die Steuersignale.

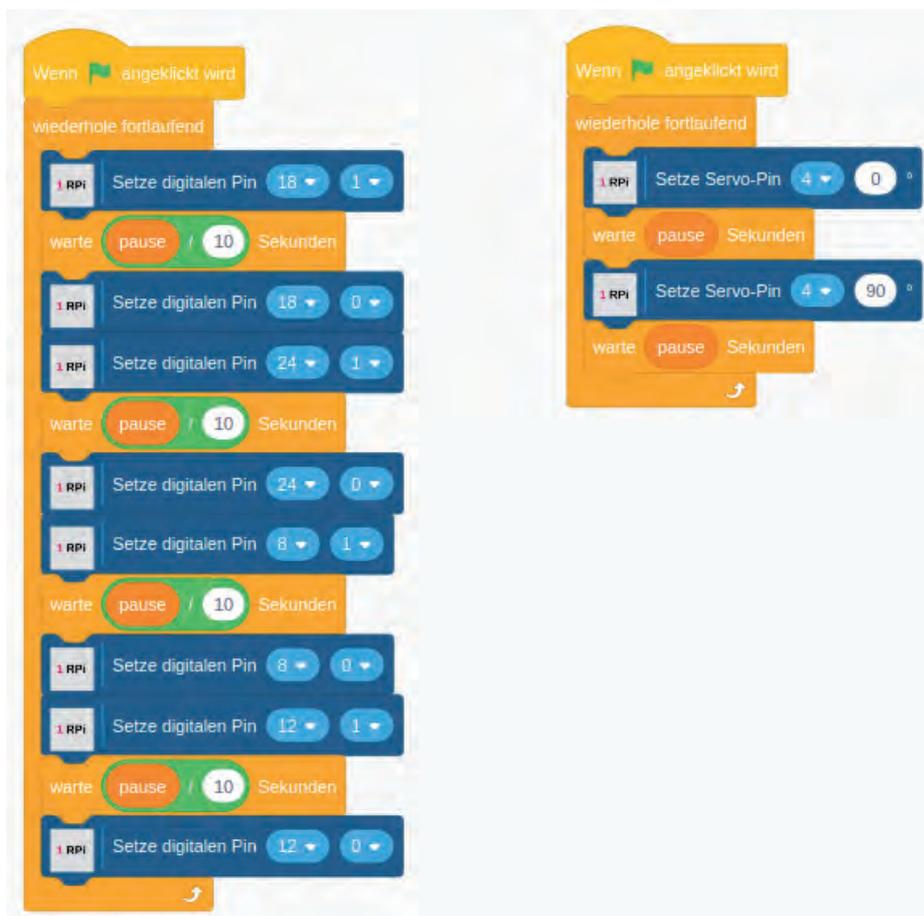
**Bauteile:** 1 Steckbrett SYB-46, 1 LED grün mit Vorwiderstand, 1 LED gelb mit Vorwiderstand, 1 LED orange mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 1 Servo, 1 Pinleiste (3 Pins), 15 GPIO-Verbindungskabel, 1 Drahtbrücke

Der Pin der Pinleiste, der mit der schwarzen Masseleitung des Servos verbunden ist, ist über eine kurze Drahtbrücke an der Masseleitung des Steckbretts angeschlossen. Um den Schaltungsaufbau übersichtlich zu halten, verwenden wir einen der anderen GND-Pins auf der GPIO-Leiste.

## Das Programm

Das Programm `20servo` besteht aus zwei unabhängigen Programmblöcken, die beide bei Klick auf das grüne Fähnchen gestartet werden. Der Programmblock, der das Lauflicht steuert, entspricht weitgehend einem bereits bekannten Programm.

Der Programmblock für den Servo wird ebenfalls bei Klick auf das grüne Fähnchen gestartet. Hier lässt eine Endlosschleife den Servo ständig zwischen 0 und 90 Grad hin- und herdrehen. Dadurch bewegt sich der Stern oben auf der Krippe. Für die Bewegung des Servos enthält die Erweiterung **OneGPIO Raspberry Pi** einen Block *Setze Servo-Pin ...*. Dieser Block enthält zwei Felder. Im linken Feld wählen Sie den Pin aus, an dem der Servo angeschlossen ist, in das rechte Feld geben Sie den Winkel ein, auf den er eingestellt werden soll. Die Pause zwischen den Servobewegungen wird auch hier mit der Variablen *pause* gesteuert. Allerdings wird deren Wert nicht durch 10 geteilt, da die Pausen nicht zu kurz sein dürfen. Der Servo benötigt einige Zeit, um die Bewegung auszuführen.



Das Programm `20servo` bewegt den Stern mit dem Servo und lässt dazu das Lauflicht leuchten.

## Heute im Adventskalender

- 1 Drahtgestänge

## Servo dreht zwei Sterne

Das Projekt von heute ist eine erweiterte Version des Projekts von gestern. Schneiden Sie den zweiten Stern aus und befestigen Sie ihn mit einer Büroklammer oder einem Reißnagel drehbar am dafür vorgesehenen zweiten Haltepunkt an der Krippe. Biegen Sie an beiden Enden des Drahtgestänges jeweils einen kleinen Haken und befestigen Sie es am Arm des Servos sowie am zweiten Stern, damit sich beide Sterne gleichzeitig bewegen können, wenn der Servo verstellt wird.



**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 Servo, 1 Pinleiste (3 Pins), 15 GPIO-Verbindungskabel, 1 Drahtbrücke

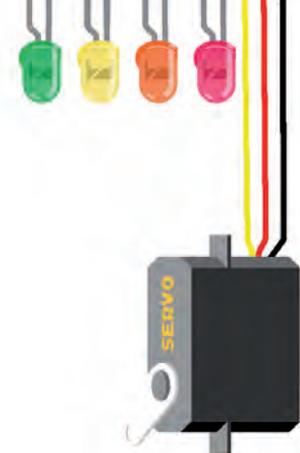
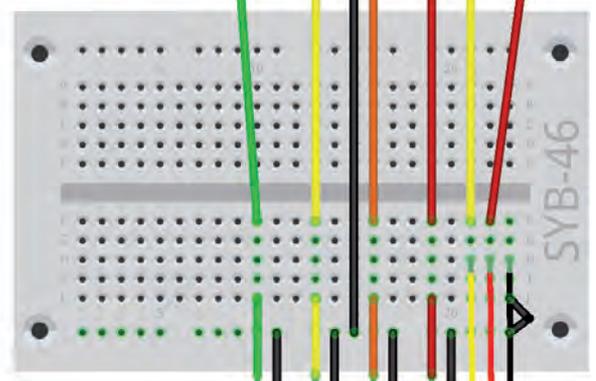
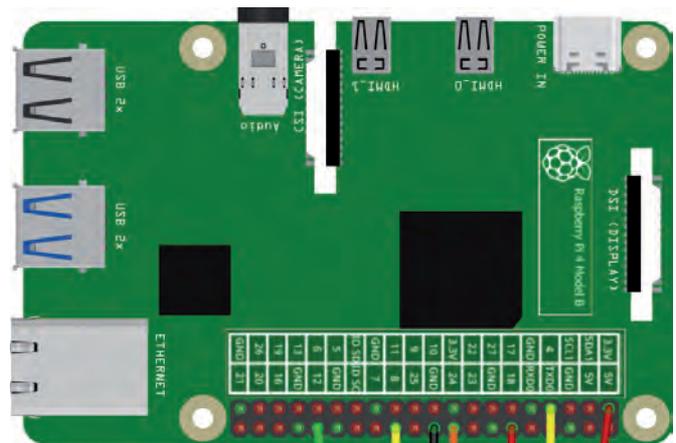
## Das Programm

Das Programm `21servo` funktioniert ähnlich wie das Programm von gestern, es bietet aber zusätzliche Funktionen:

- Die Drehgeschwindigkeit des Sterns kann unabhängig von der Blinkgeschwindigkeit des Lauflichts eingestellt werden.
- Der Stern auf dem Bühnenbild in Scratch dreht sich mit dem echten Stern am Servo mit.

Das Programm nutzt statt der Katze eine neue Figur: einen Stern. Diesen Stern können Sie in Scratch selbst zeichnen oder aus der Datei `stern.png` aus dem Download importieren.

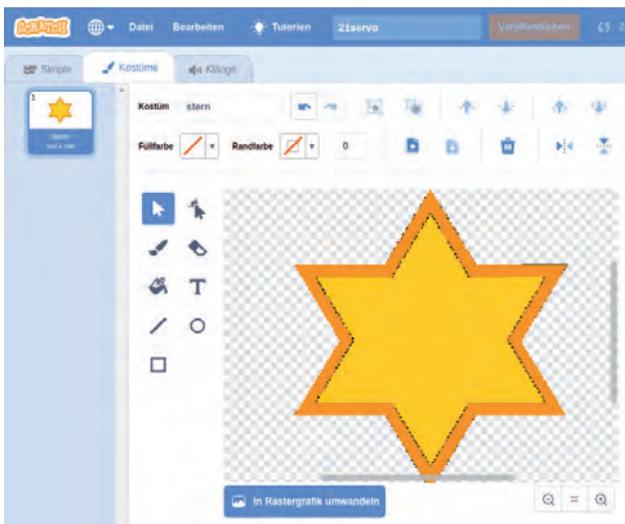
Für die Blinkgeschwindigkeit und die Drehgeschwindigkeit werden die zwei Variablen `blink` und `stern` verwendet, die beide über Schieberegler einstellbar sind. Der Wertebereich der Variablen `blink` reicht von 0 bis 10, was bei der Division durch 10 zwischen null und einer Sekunde entspricht, der Wertebereich der Variablen `stern` reicht von 1 bis 10. Diese Variable wird im Programm nicht weiter umgerechnet, sondern stellt direkt eine bis zehn Sekunden dar.



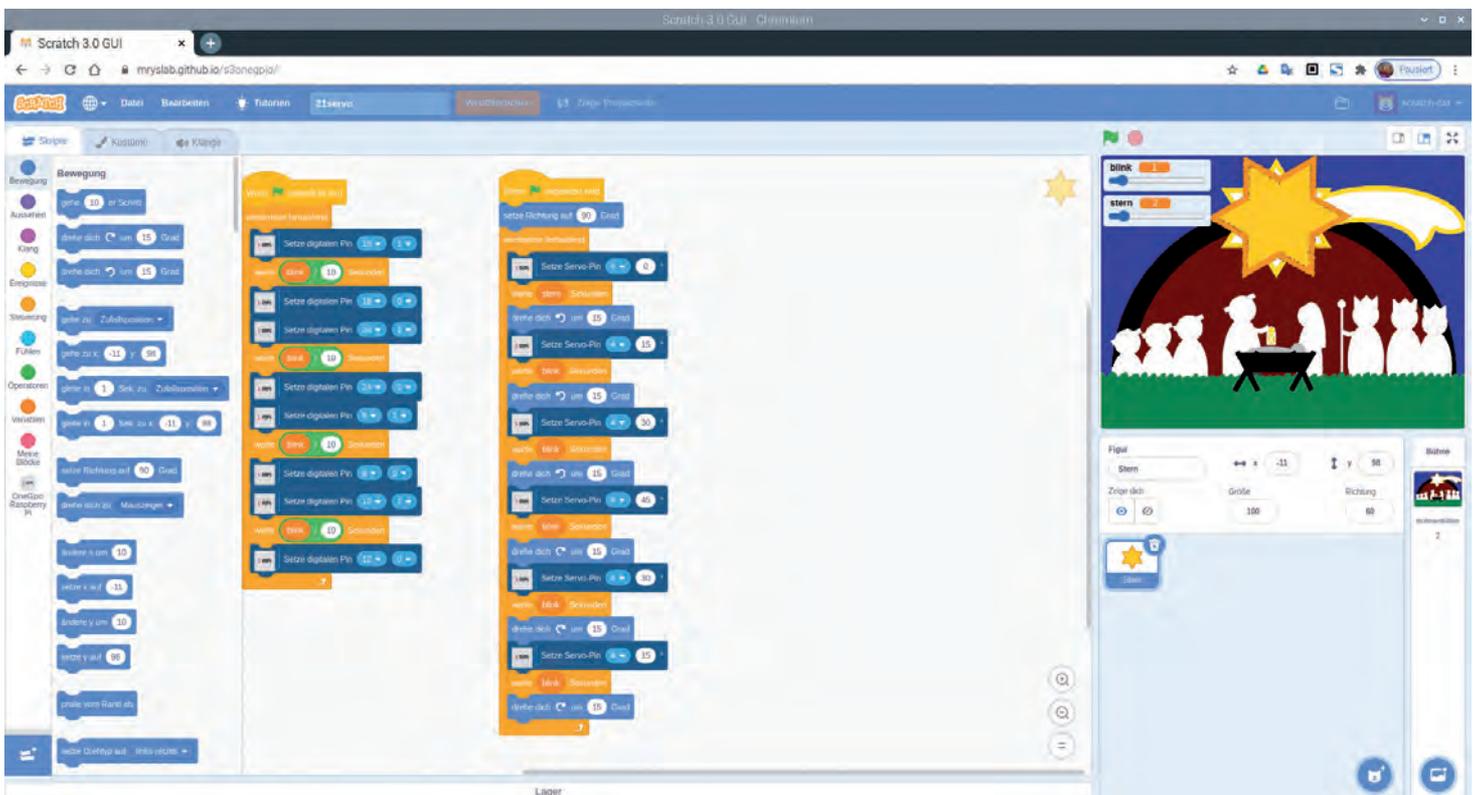
fritzing

Vier LEDs und ein Servo – der gleiche Aufbau wie gestern.

# Raspberry-Pi-Adventskalender



Die neue Figur im Programm.



Das Programm besteht wieder aus zwei Programmblöcken, die beide bei Klick auf das grüne Fähnchen starten. Der Programmblock für das Lauflicht ist bereits bekannt. Für den Stern verwenden wir neue Blöcke. Zuerst wird die Figur *Stern* mit dem Block *setze Richtung auf 90 Grad* in die Grundstellung gebracht. Die Richtung von 90 Grad entspricht der Vorgabe aus dem Kostüm.

Anschließend startet eine *wiederhole fortlaufend*-Schleife, die den Servo in 15-Grad-Schritten von 0 bis 45 Grad und wieder zurück dreht. Zwischen den Schritten wartet das Programm die in der Variablen *stern* festgelegte Zeit. Zusätzlich wird bei jeder Servobewegung die Figur um 15 Grad gedreht – in den ersten drei Schritten gegen den Uhrzeigersinn und danach wieder im Uhrzeigersinn, sodass sie der Bewegung des Servos folgt und am Ende eines Schleifendurchlaufs wieder die Grundstellung erreicht hat. Dazu bietet Scratch zwei Blöcke an, mit denen eine Figur in beliebigen Schritten gedreht werden kann.

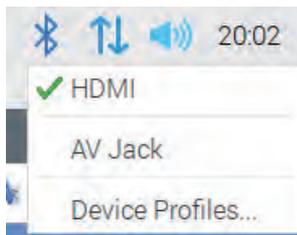
## Heute im Adventskalender

- 1 GPIO-Verbindungskabel

## Der Raspberry Pi erzeugt Töne

Der Raspberry Pi kann über einen HDMI-Monitor, einen externen Lautsprecher oder einen Kopfhörer an der analogen 3,5-mm-Klinkenbuchse Klänge 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. Der Raspberry Pi 400 hat keine analoge 3,5-mm-Klinkenbuchse. Hier kann nur ein HDMI-Audiosignal verwendet werden.

Klicken Sie mit der rechten Maustaste oben rechts auf das Lautsprechersymbol, um den Audioausgang auszuwählen.



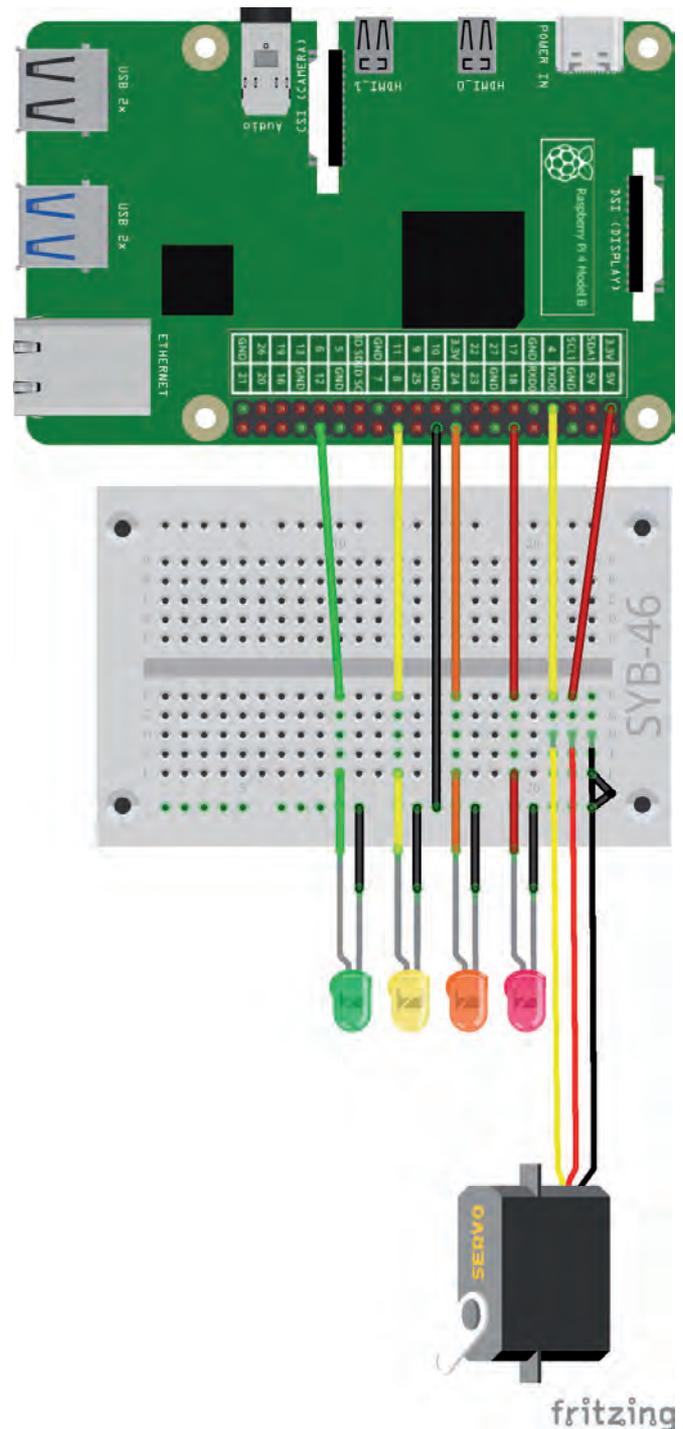
Audioausgang auswählen.

**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 Servo, 1 Pinleiste (3 Pins), 15 GPIO-Verbindungskabel, 1 Drahtbrücke

## Das Programm

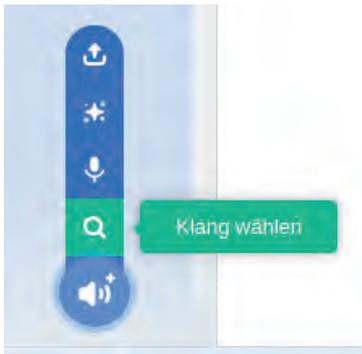
Das Programm `22servo` funktioniert ähnlich wie das Programm des vorherigen Tags. Es lässt vier LEDs in kurzen Abständen nacheinander aufblinken und dreht dazu den Stern und den Stern auf dem Bildschirm. Zu jeder LED ertönt jetzt zusätzlich ein Ton.

Die Pausen im Programm für das Lauflicht werden diesmal verlängert. Das Lauflicht soll nicht zu schnell flackern, stattdessen soll genügend Zeit bleiben, damit bei jeder aufblinkenden LED ein Ton erklingen kann. Nehmen Sie dazu die Division durch 10 heraus und nutzen Sie die eingestellte Zeit `blink` in voller Länge aus.



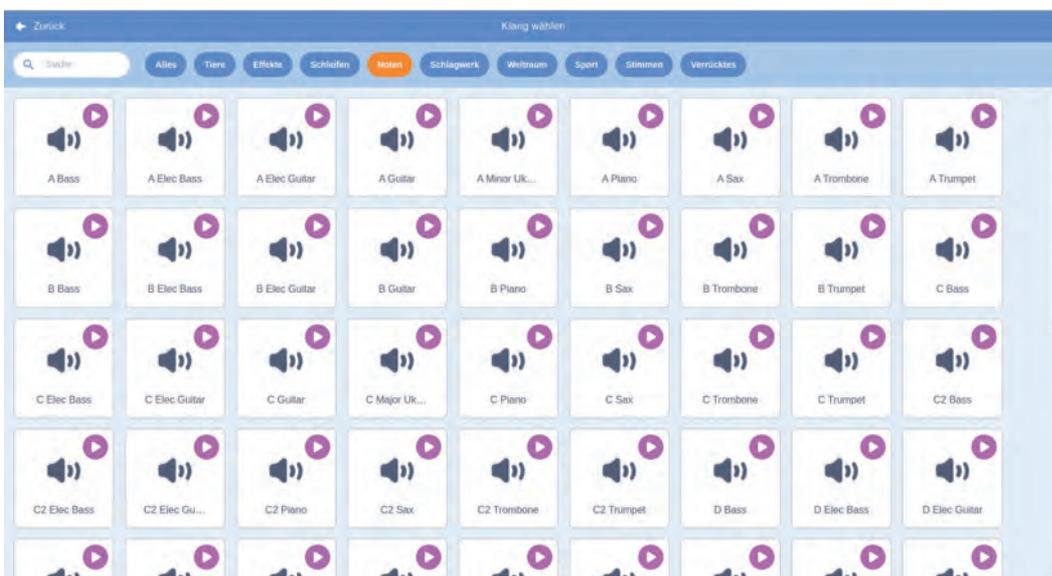
Vier LEDs und ein Servo – der gleiche Aufbau wie gestern.

Die Klänge werden über den Block *spiele Klang ...* abgespielt. Damit ein Klang verwendet werden kann, müssen Sie ihn auf der Registerkarte *Klänge* links oben zuerst in das Programm importieren. Fahren Sie dazu mit der Maus auf das Lautsprecher-symbol links unten und klicken Sie dann auf das Lupensymbol *Klang wählen*.

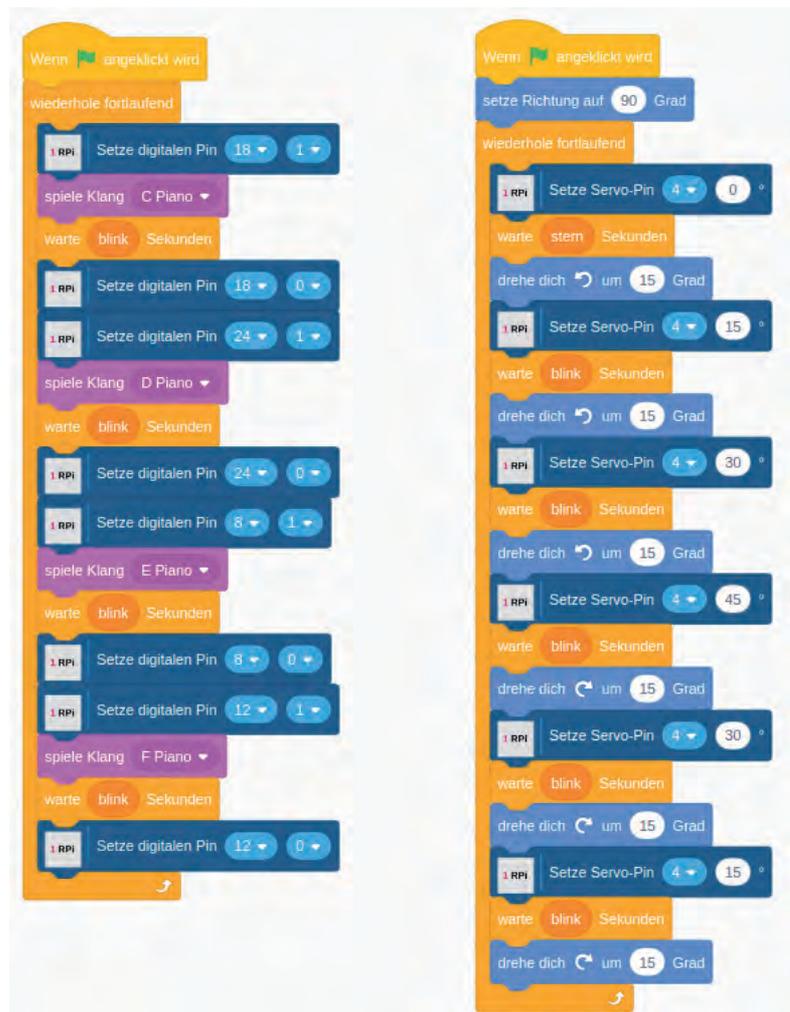


Klänge in das Programm importieren.

Jetzt werden alle in Scratch verfügbaren Klänge angezeigt. Wenn Sie mit dem Maus-pfeil über eines der violetten Abspielsym-bole fahren, können Sie den Klang hören.



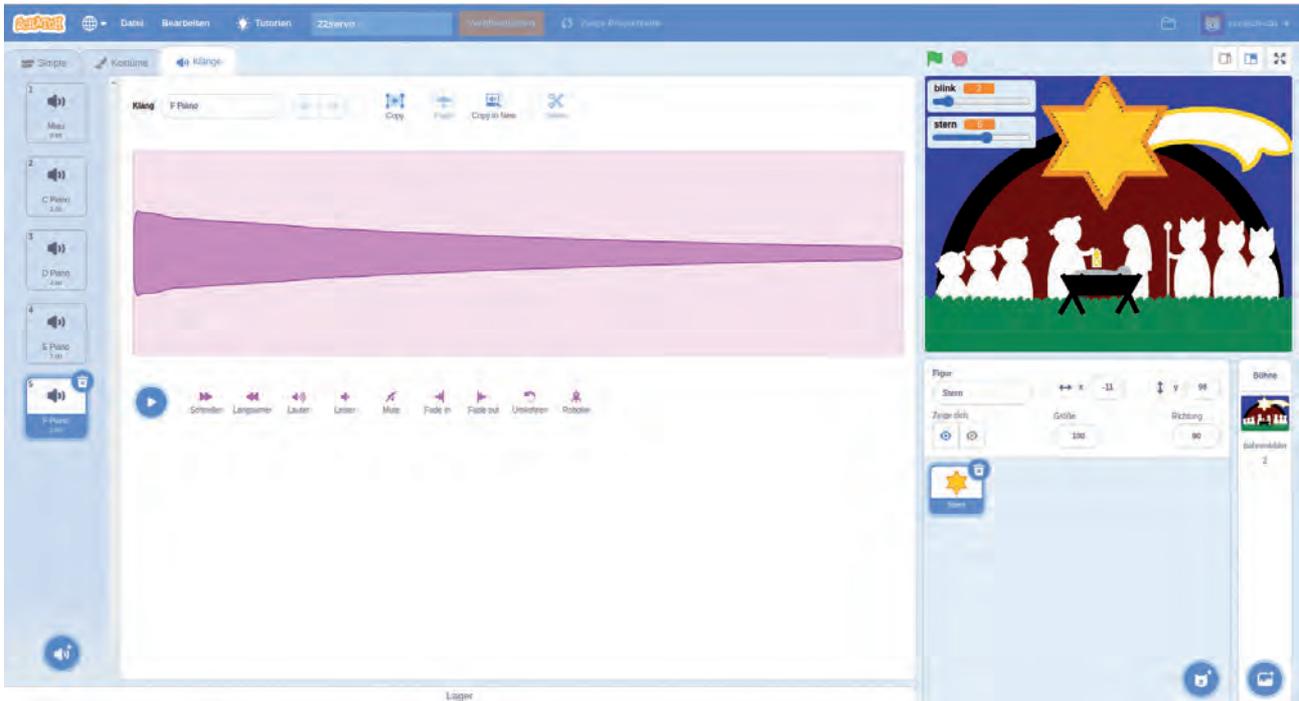
Klänge anhören und importieren.



Das Programm `22servo` lässt vier LEDs in kurzen Abständen aufblinken, dreht die Sterne und spielt dazu Töne ab.

# Raspberry-Pi-Adventskalender

Klicken Sie oben auf das Symbol *Noten*, um die Auswahl nach Musiknoten zu filtern. Importieren Sie nacheinander die Klänge *C-Piano*, *D-Piano*, *E-Piano* und *F-Piano*.



Die verwendeten Klänge auf der Registerkarte *Klänge*.

Klicken Sie anschließend wieder auf die Registerkarte *Skripte*, um die Blöcke *spiele Klang ...* in das Programm einzubauen, und zwar jedes Mal, nachdem eine LED eingeschaltet wurde. Im Listenfeld dieser Blöcke stehen alle importierten Klänge zur Auswahl zur Verfügung.



Klang im Programm auswählen.

## Heute im Adventskalender

- 1 GPIO-Verbindungskabel

## RGB-Farbspiele

Ähnlich wie das vorherige Programm zeigt das Programm heute ein Lauflicht, dreht die Sterne und spielt Töne ab. Zusätzlich leuchtet die RGB-LED in wechselnden Farben.

**Bauteile:** 1 Steckbrett SYB-46, 1 RGB-LED mit Vorwiderständen, 1 LED grün mit Vorwiderstand, 1 LED gelb mit Vorwiderstand, 1 LED orange mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 1 Servo, 1 Pinleiste (3 Pins), 22 GPIO-Verbindungskabel, 1 Drahtbrücke

## Das Programm

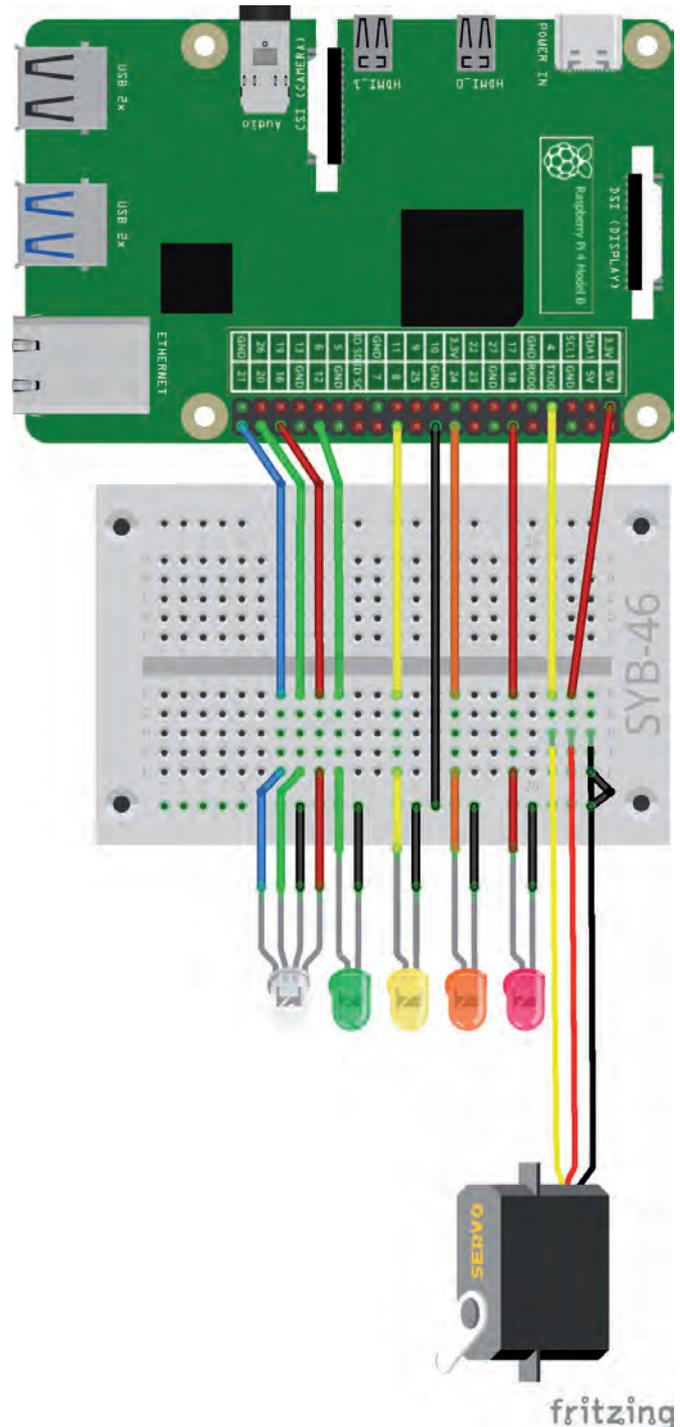
Die Programmblöcke für Lauflicht, Klänge und Servo im Programm 23rgb funktionieren wie im Programm von gestern. Neu ist der Programmblock zur Steuerung des Farbverlaufs auf der RGB-LED.

Eine Endlosschleife errechnet nacheinander die RGB-Werte für alle Farbwerte des HSV-Farbspektrums. Bei jedem Wert leuchtet die RGB-LED eine kurze Zeit lang, die über den Schieberegler *rgb* einstellbar ist, in der entsprechenden Farbe. Dazu werden die GPIO-Pins der RGB-LED auf die errechneten PWM-Werte gesetzt.

### HSV- und RGB-Farbsystem

Das RGB-Farbsystem, das bisher in allen Programmen verwendet wurde, beschreibt Farben mithilfe der drei Komponenten Rot, Grün und Blau, die miteinander gemischt werden. Es ist für einen Menschen 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).

Der H-Wert kann entsprechend den Gradzahlen auf einem Farbkreis Werte zwischen 0 und 360 annehmen. 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.

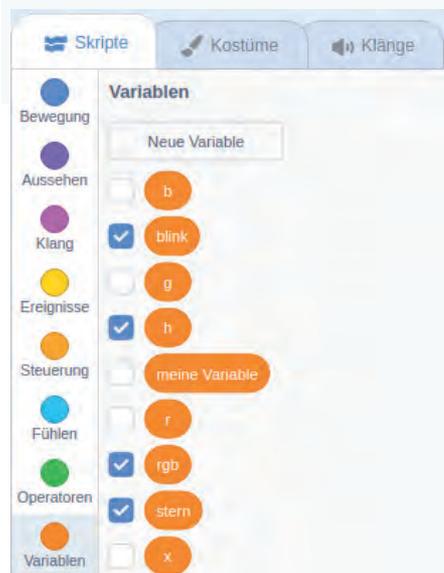
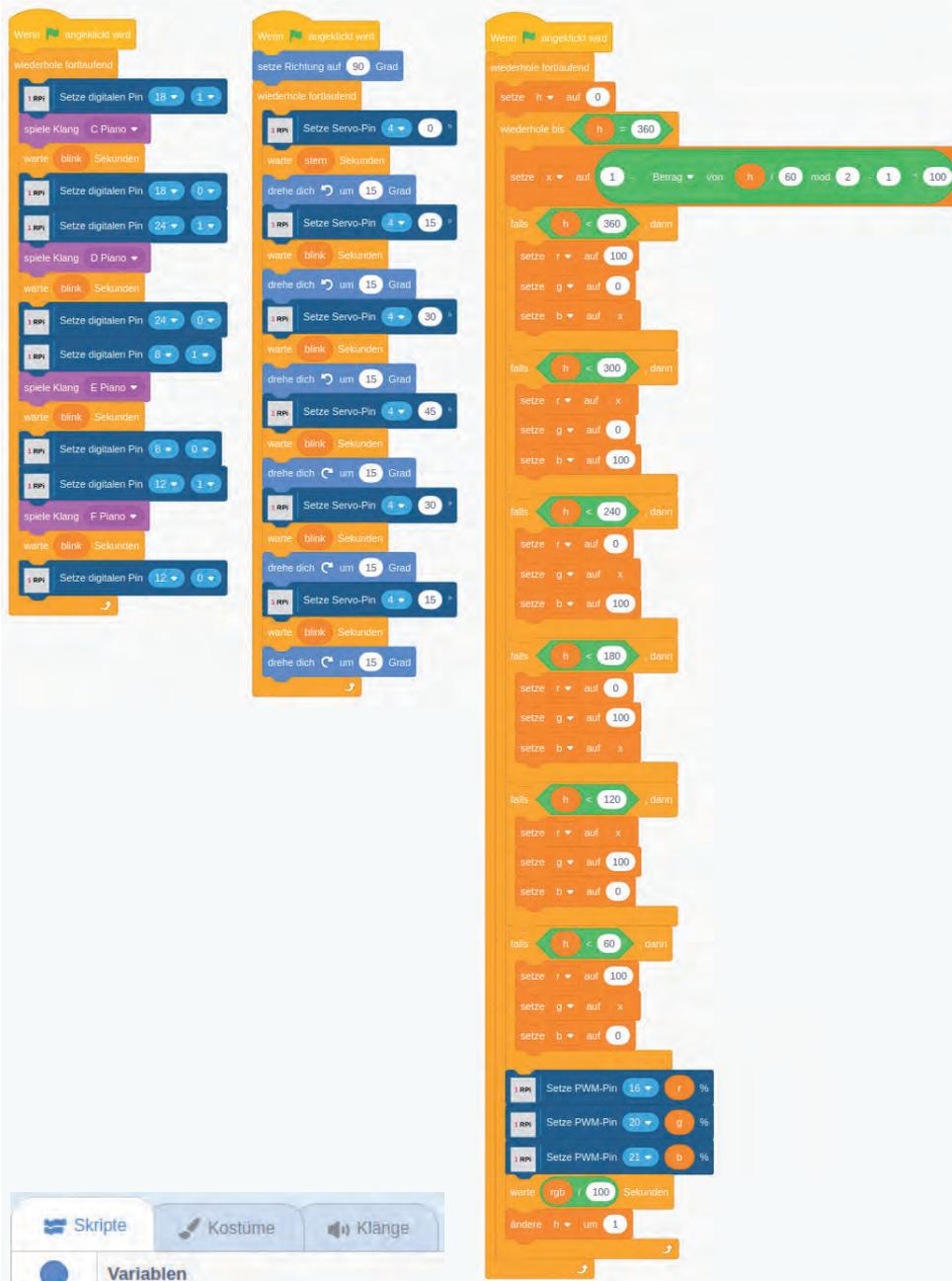


Vier LEDs, RGB-LED und ein Servo.

Zusätzlich zu den neuen Variablen *h* für den H-Wert und *x* für einen zu berechnenden Zwischenwert werden die drei Variablen *r*, *g*, *b* angelegt, in die während der Berechnung die PWM-Werte für die drei Farbkomponenten der RGB-LED eingetragen werden.

Die Schleife *wiederhole bis h=360* innerhalb der *wiederhole fortlaufend*-Schleife zählt den H-Wert entsprechend den Gradzahlen auf einem Farbkreis zwischen 0 und 360 hoch und errechnet daraus die drei Farbkomponenten R, G und B. Die Werte S = Saturation (Sättigung) und V = Value (Helligkeitswert) werden automatisch auf den Maximalwert gesetzt. Wie die Grafik zeigt, gelten für die sechs 60-Grad-Bereiche jeweils eigene lineare Verlaufskurven. In jedem dieser Bereiche wird ein Farbwert variabel errechnet, und die anderen beiden werden

# Raspberry-Pi-Adventskalender



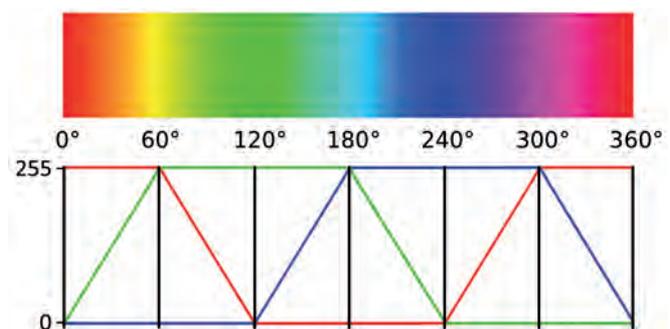
Die Variablen im Programm.

auf Minimum 0 oder Maximum 100 gesetzt.

Nachdem die RGB-Werte errechnet und in den Variablen  $r$ ,  $g$ ,  $b$  gespeichert sind, werden die PWM-Pins auf die jeweiligen Farbwerte gesetzt. Nach einer kurzen Wartezeit wird der H-Wert um 1 erhöht. Nach einem vollen Durchlauf über 360 Grad beginnt der Zyklus von Neuem bei 0.

Die *falls ... dann*-Abfragen laufen sehr schnell, wir können es uns also sparen, genaue Bereiche von ... bis ... abzufragen, und stattdessen einfach  $<...$ -Abfragen nutzen. Es werden also in vielen Fällen mehrere Abfragen direkt hintereinander ablaufen. Die drei PWM-Signale werden am Ende aber nur einmal gesetzt. Nach einer kurzen Wartezeit von wenigen Hundertstelsekunden wird die Variable  $h$  um 1 erhöht, und der nächste Schleifendurchlauf beginnt.

Das Programm 23rgb lässt vier LEDs in kurzen Abständen aufblinken, spielt Töne ab und zeigt zusätzlich einen Farbverlauf auf der RGB-LED.



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

## Heute im Adventskalender

- 1 Downloadcode für MP3-Weihnachtslieder
- 2 GPIO-Verbindungskabel

## Weihnachtslieder auf dem Raspberry Pi

Am letzten Tag wird das Programm nochmals um zwei Taster erweitert. Sie spielen zwei verschiedene Weihnachtslieder ab.

**Bauteile:** 1 Steckbrett SYB-46, 1 RGB-LED mit Vorwiderständen, 1 LED grün mit Vorwiderstand, 1 LED gelb mit Vorwiderstand, 1 LED orange mit Vorwiderstand, 1 LED rot mit Vorwiderstand, 2 Taster, 1 Servo, 1 Pinleiste (3 Pins), 24 GPIO-Verbindungskabel, 4 Drahtbrücken

Bei dieser Schaltung wird die obere waagerechte Leiste des Steckbretts für die +3,3-V-Leitung genutzt. Die Taster und der Anschluss für den Servo sind über Drahtbrücken mit dieser Leiste verbunden.

### Wichtiger Unterschied

Bei der standardmäßig mitgelieferten GPIO-Erweiterung wird ein Taster mit der Masseleitung verbunden. Drückt man diesen Taster, ist das Signal am entsprechenden GPIO-Pin 0, lässt man ihn los, wird das Signal durch den eingebauten Pull-up-Widerstand auf 1 gezogen.

Bei der Erweiterung **OneGPIO Raspberry Pi** wird ein Taster mit der +3,3-V-Leitung verbunden. Drückt man diesen Taster, ist das Signal am entsprechenden GPIO-Pin 1, lässt man ihn los, liegt am GPIO-Pin wieder 0 an.

## Das Programm

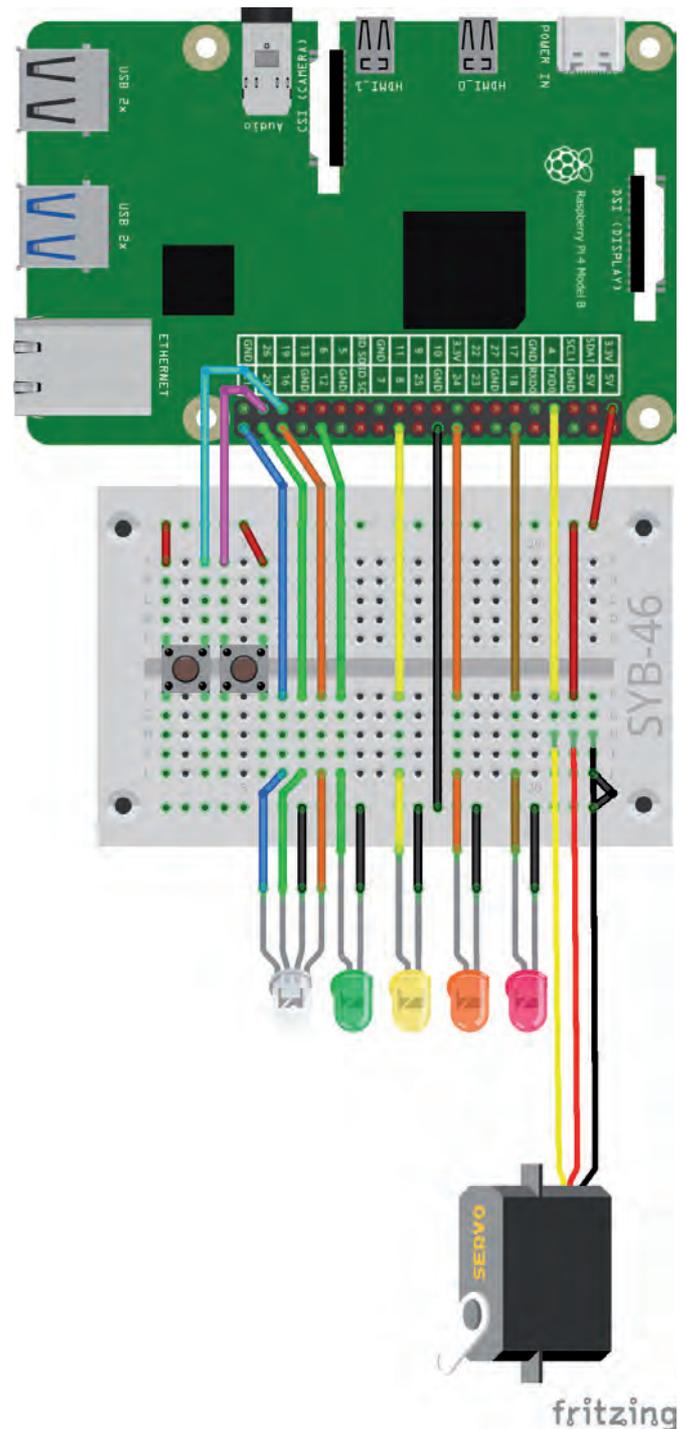
Die Programmblöcke für Lauflicht, RGB-LED und Servo im Programm 24weihnachten funktionieren wie im Programm von gestern. Im Programmblock für das Lauflicht wurden die Klänge gelöscht. Neu ist ein Programmblock, der die beiden Taster abfragt und auf Knopfdruck die Weihnachtslieder spielt.

Scratch 3 kann auf dem Raspberry Pi MP3-Dateien abspielen, die in das Programm importiert wurden. Importieren Sie dazu über das Symbol *Klang hochladen* auf der Registerkarte *Klänge* zwei Weihnachtslieder, die Sie über den Downloadcode im heutigen Türchen des Adventskalenders erhalten.

Die importierten Lieder erscheinen auf der Registerkarte *Klänge*. Hier können Sie sie anhören und auch nachbearbeiten.

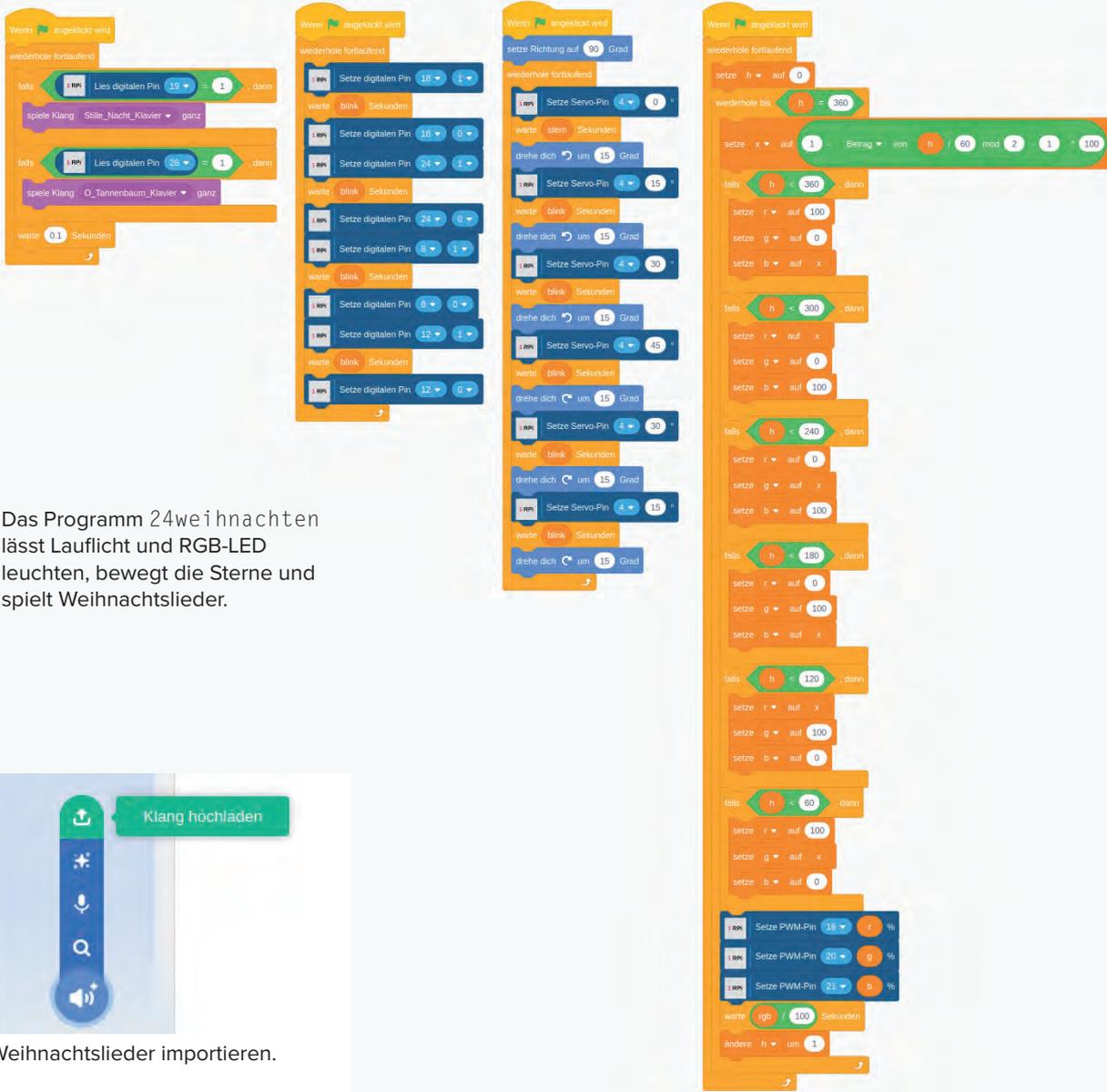
Bei Drücken eines Tasters soll das zugehörige Weihnachtslied abgespielt werden. Dazu fragt eine *wiederhole fortlaufend*-Schleife, die bei Klick auf das grüne Fähnchen gestartet wird, nacheinander die beiden GPIO-Pins 19 und 26 ab. Hat einer der Pins den Wert 1, ist der Taster gedrückt. Dann spielt ein Block *spiele Klang ... ganz* das ausgewählte Weihnachtslied ab. Dieser Block spielt einen Klang in voller Länge ab, bevor der folgende Block im Programm abgearbeitet wird. Auf diese Weise wird das Lied nicht abgebrochen, falls zwischenzeitlich der andere Taster gedrückt wird.

Am Ende der Endlosschleife ist eine kurze Wartezeit eingebaut, damit sich das Programm nicht „überschlägt“, wenn kein Taster gedrückt ist, und damit die Endlosschleife in sehr kurzer Folge ständig wiederholt wird.

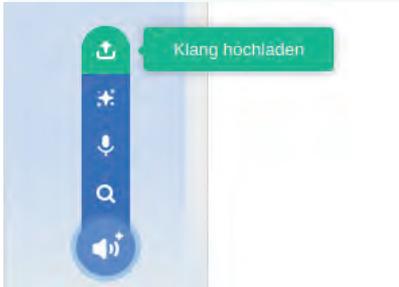


Vier LEDs, RGB-LED, zwei Taster und ein Servo.

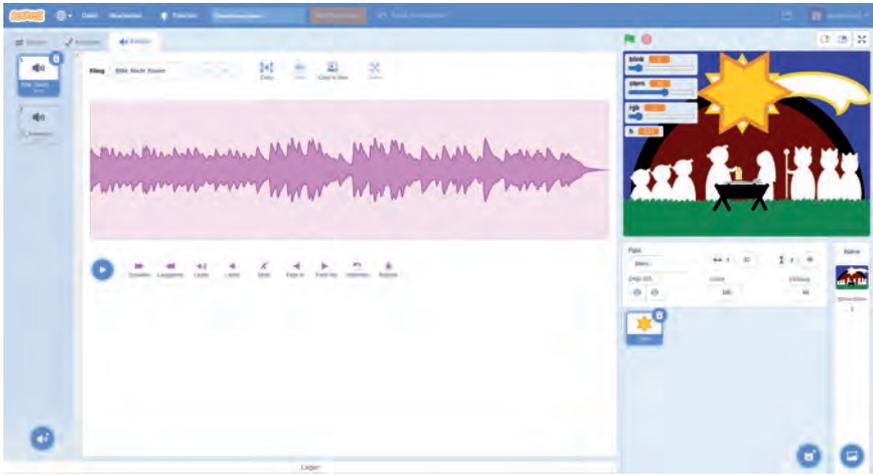
# Raspberry-Pi-Adventskalender



Das Programm 24weihnachten lässt Lauflicht und RGB-LED leuchten, bewegt die Sterne und spielt Weihnachtslieder.



Weihnachtslieder importieren.



Die importierten Weihnachtslieder.

## Übersicht

Zur Übersicht noch einmal alle Funktionen des Programms:

- Die vier LEDs laufen als Lauflicht. Der Regler *blink* regelt die Blinkgeschwindigkeit.
- Der Stern auf der Krippe und auch der in der Grafik auf der Scratch-Bühne drehen sich hin und her. Der Regler *stern* regelt die Blinkgeschwindigkeit.
- Die RGB-LED zeigt einen ständigen Farbwechsel im HSV-Farbspektrum. Der Regler *rgb* regelt die Blinkgeschwindigkeit.
- Ein Druck auf einen der beiden Taster spielt ein Weihnachtslied ab.

Frohe Weihnachten!

**FRANZIS**

Raspberry Pi is a trademark of the Raspberry Pi Foundation  
© 2021 FRANZIS Verlag GmbH, Richard-Reitzner-Allee 2, D-85540 Haar, Germany  
2021/01



WEEE-Reg.-Nr.  
DE21445697

**GTIN 4019631551030**



4 019631 551030