



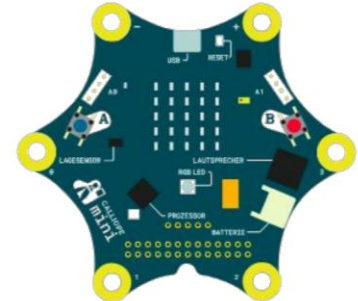
Startvideo

Einführung

Wir wollen heute einen kontaktlosen und optischen Countdown-Timer mit dem Calliope mini bauen.

Kontaktlos bedeutet, dass du den Timer starten kannst, ohne den Calliope mini zu berühren.

Mit diesem Tutorial lernst du, wie du die 25 LEDs des Calliope mini einzeln oder in Gruppen programmieren kannst.



1. LED Helligkeitsunterschiede

Aufgabe 1: Erstelle ein Programm, mit dem fünf LEDs waagrecht nebeneinander in unterschiedlichen Helligkeiten leuchten, wenn der Calliope mini angeschaltet wird.

Gib zunächst lab.open-roberta.org in deinen Browser ein und wähle dann Calliope mini als System. → Dort kannst du programmieren.

Die benötigten Blöcke findest du wie folgt:

- wähle zunächst den Expertenmodus, um weitere Code-Blöcke freizuschalten.
- Unter **Aktion->Anzeige**
 - Setze LED x, y, Helligkeit

Tipp: Auf dem Calliope mini gibt es ein quadratisches Display, das aus 5 x 5 roten LEDs besteht.

x: x-Position von links nach rechts 0-4

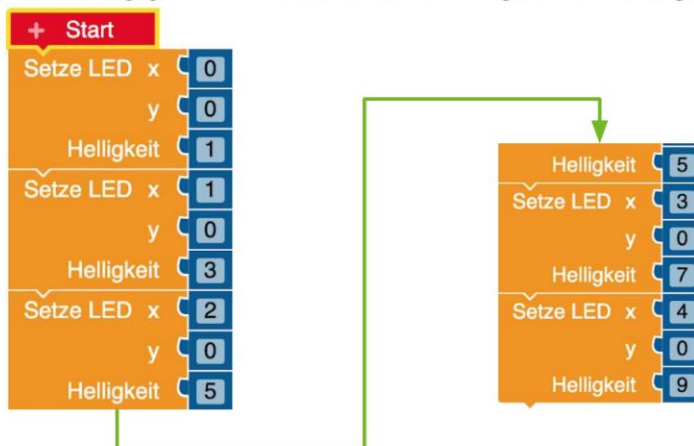
y: y-Position von oben nach unten 0-4

Helligkeit: von aus bis volle Helligkeit 0-9

Teste dein Programm erst im *Simulator* und dann auf dem Calliope mini.

Ergebnis

Wenn du alles richtig gemacht hast, sollte dein Ergebnis wie folgt aussehen:





1.2 LED - Zeile

Aufgabe 2: Erstelle nun ein neues Programm, das die fünf LEDs der oberen Zeile in aufsteigender Helligkeit nacheinander einschaltet, den Bildschirm dann löscht und die LEDs erneut der Reihe nach aufleuchten lässt.

Schritt 1: Erstellen einer Variable "Spalte" und einer Variable "Helligkeit".

In der ersten Aufgabe haben wir einen Programmierblock für jede LED gebraucht, die wir zum Leuchten gebracht haben. Diese Arbeit wollen wir uns sparen und verwenden ab jetzt Variablen, um eine spezifische LED aus dem 5 x 5 Quadrat anzusteuern.

Schau dir die folgende Infobox an und erstelle eine Variable vom **Typ Zahl** mit dem Namen *Spalte* und dem **Startwert 0** und eine zweite Variable mit dem Namen *Helligkeit* mit dem **Startwert 1**.

Info

The screenshot shows the 'Add Variable' dialog box in Scratch. It has a 'Start' button at the top left. Below it are two variable creation options. The first is 'Variable Spalte : Zahl' with a start value of 0. The second is 'Variable Helligkeit : Zahl' with a start value of 1. Annotations with arrows point to various parts: 'neue Variable hinzufügen' points to the 'Start' button; 'Name der Variable' points to the text 'Spalte'; 'Startwert der Variable' points to the value '1'; 'Typ der Variable' points to the dropdown menu 'Zahl'; and 'Variable löschen' points to the minus sign on the left of the variable list.

Schritt 2: Bringe eine spezifische LED zum Leuchten

Du benötigst zuerst eine Schleife, die fünf mal wiederholt wird. Darin setzt du den Setze LED Block und verbindest die x-Koordinate mit der Variable *Spalte*, die y-Koordinate bleibt 0 (oberste Zeile) und die Helligkeit verbindest du mit der Variable *Helligkeit*.

Hierfür benötigst du die Blöcke aus dem Bereich *Mathematik* und *Variablen*

Zwischenergebnis

Hier siehst du ein Zwischenergebnis, das bislang nur eine LED zum Leuchten bringt. Damit die anderen vier LEDs der gleichen Zeile eingeschaltet werden, musst du nun deine Variablen innerhalb der Schleife verändern.

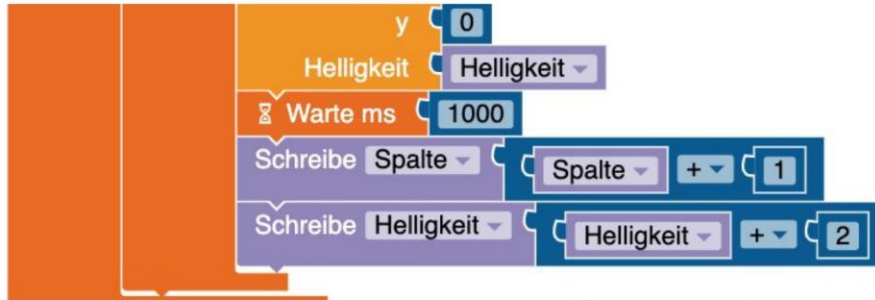
The screenshot shows a Scratch script starting with a 'Start' block. Below it are two 'Variable' blocks: 'Variable Spalte : Zahl' with a value of 0, and 'Variable Helligkeit : Zahl' with a value of 1. This is followed by a 'Wiederhole unendlich oft' loop. Inside this loop is a 'mache' block with a 'Wiederhole 5 mal' block. Inside the '5 mal' loop is a 'mache' block containing a 'Setze LED x' block with 'Spalte' selected, a 'y' block with '0' selected, and a 'Helligkeit' block with 'Helligkeit' selected.



Aufgabe 2: Erstelle ein Programm, das die fünf LEDs der oberen Zeile in aufsteigender Helligkeit nacheinander einschaltet, den Bildschirm dann löscht und die LEDs erneut aufleuchten lässt.

Schritt 3: Bringe fünf LEDs nacheinander mit unterschiedlicher Helligkeit zum Leuchten

Bei jeder Schleifen-Wiederholung soll die Variable **Spalte** um 1 und die Variable **Helligkeit** um 2 erhöht werden.

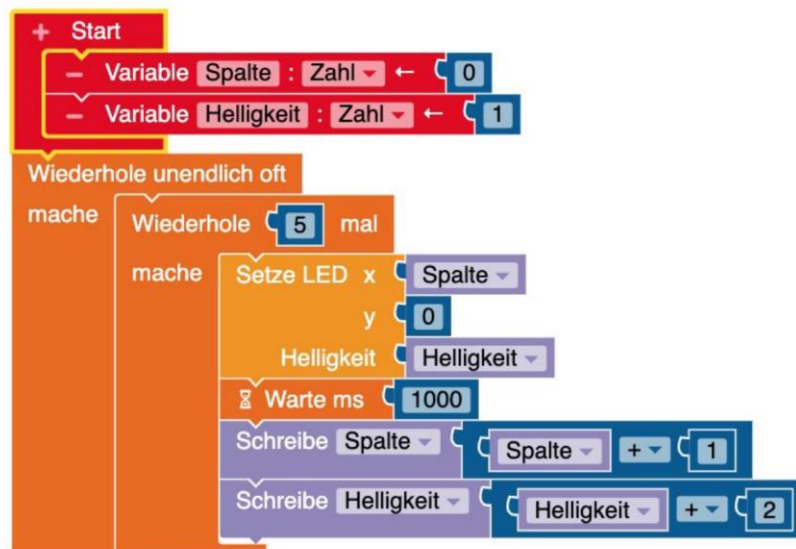


Füge einen **Warte-Block** ein, um den Wechsel zwischen den LEDs zu kontrollieren.

Teste dein Programm erst im *Simulator* und dann auf dem Calliope mini.

Ergebnis

Wenn du alles richtig gemacht hast, sollte dein Ergebnis wie folgt aussehen:



Welche Blöcke musst du hinzufügen, wenn du die fünf LEDs erneut aufleuchten lassen willst?

Tipp: Lösche den Bildschirm, setze die Variable **Spalte** wieder auf 0 und die Variable **Helligkeit** auf 1. Füge zuletzt noch einen **Warte-Block** hinzu. Diese Blöcke musst du unterhalb der **Wiederhole 5 mal** Schleife in den Code einfügen.

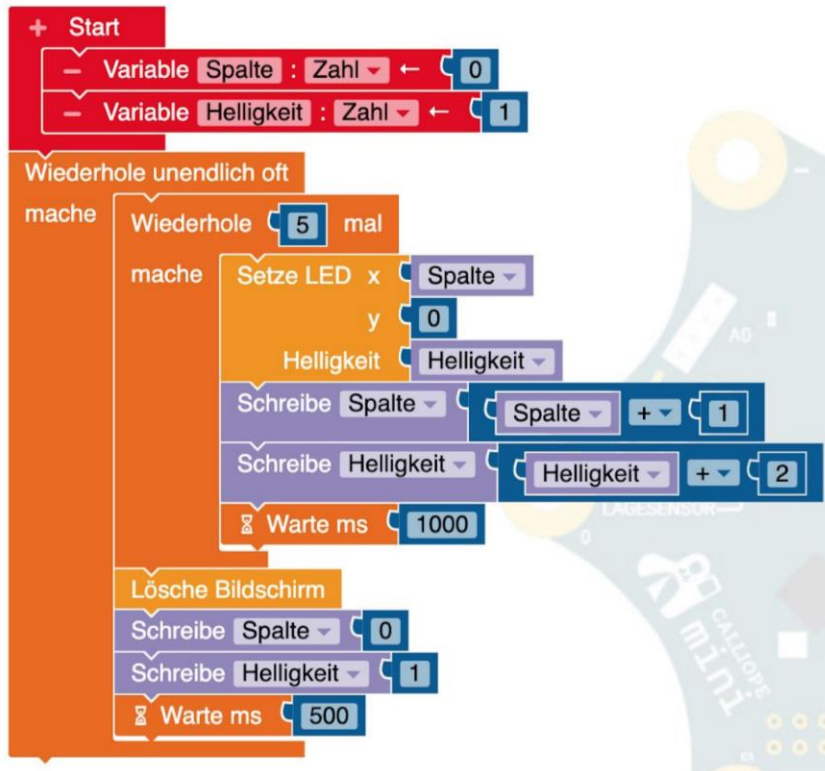
Hinweis: Versuche zunächst deine eigene Lösung zu erstellen. Wenn du nicht weiter kommst, findest du auf der nächsten Seite ein mögliches Ergebnis.



1.3 LED Reihe (Teil 3)

Ergebnis:

Wenn du alles richtig gemacht hast, sollte dein Ergebnis wie folgt aussehen:



1.4 Optischer Countdown-Timer

Aufgabe 3: Nun sollst du alle 25 LEDs so programmieren, dass sie zu Beginn alle leuchten und dann nacheinander mit einer Sekunde Versatz ausgehen. So entsteht ein optischer Countdown-Timer mit 25 Sekunden Laufzeit.

Schritt 1: Erstelle eine neue Variable "Zeile".

Die Variable *Helligkeit* benötigst du nun nicht mehr. Erstelle stattdessen eine Variable *Zeile* mit dem **Startwert 0**.





1.4. Optischer Countdown Timer Teil 2

Schritt 2: Verbinde die Variable *Spalte* mit der **Y-Koordinate** des LED-Blocks und setze die Helligkeit auf 0. Denn das Ziel soll sein, dass die LEDs ausgeschaltet werden.

Tip: Starte mit 25 leuchtenden LEDs. Klicke dafür in die leeren Felder. Die # bedeutet, dass diese LED aktiviert ist und leuchtet.

Tip: Du benötigst nun eine weitere **Wiederhole ...mal Schleife**, weil du nicht nur LEDs in einer eindimensionalen **Zeile** programmierst, sondern alle 25 LEDs der quadratischen **Fläche (5x5)**.

Schritt 3: In der neuen **Wiederhole ... Schleife** programmierst du nun den Zeilenwechsel, indem du die Variable *Spalte* wieder auf 0 setzt und die Variable *Zeile* um +1 erhöhst.

Teste dein Programm erst im *Simulator* und dann auf dem Calliope mini.

Fertig ist der optische Countdown-Timer. Erweitere dein Programm nun so, dass du deinen optischen Countdown-Timer ohne Berührung des Calliope mini starten kannst.

Hinweis: Der Calliope mini bietet verschiedene Möglichkeiten einer kontaktlosen Bedienung. Versuche zunächst deine eigene Lösung zu erstellen. Wenn du nicht weiter kommst, findest du auf der nächsten Seite ein mögliches Ergebnis.



1.5. Der kontaktlose optische Countdown-Timer

Aufgabe 4: Verwende das eingebaute Mikrofon des Calliope mini, um deinen Countdown-Timer zu starten.

Schritt 1: Füge eine weitere Variable *Mic* hinzu und verwende sie in Verbindung mit einer **bedingten Anweisung wenn... dann...** Sobald die gemessene Lautstärke einen bestimmten Pegel überschreitet, soll der Countdown Timer starten.

The code starts with three variables: 'spalte' (column), 'zeile' (row), and 'mic' (microphone), all set to 0. An infinite loop 'Wiederhole unendlich oft' contains a 'Zeige Bild' block showing a 5x5 grid of '#' characters. Below this, a 'wenn mic > 30' block triggers a 'mache' loop. Inside, a 'Wiederhole 5 mal' loop contains another 'mache' loop with 'Wiederhole 5 mal' iterations. This inner loop sets the LED x and y coordinates to 'spalte' and 'zeile' respectively, sets brightness to 0, increments 'spalte' by 1, waits 1000ms, increments 'zeile' by 1, and resets 'spalte' to 0. A tip box points to the 'gib Geräusch % Mikrofon' block, stating: 'Tipp: Der Calliope misst die Lautstärke in %. Probiere, welcher Wert für euer Anwendungsbeispiel am besten funktioniert.'

1.6. Wie geht es weiter?

Lasse deiner Kreativität freien Raum. Überlege dir, wie du das Programm erweitern kannst.

Hier zwei mögliche Ideen:

- Lass nach Ablauf des Countdown-Timers ein Symbol auf dem Display 3x blinken.
- Füge eine Melodie hinzu, die nach Ablauf des Timers gespielt wird.