

Hallo



Liebe Lehrer/innen, liebe Eltern,

wir wünschen euren Schülern, Schülerinnen und Kindern viel Freude an diesem Mitmachbuch. Es ist so ausgerichtet, dass die Kinder größtenteils eigenständig mit dem Material arbeiten können - ob im Unterricht, in der Fernbeschulung oder zu Hause in den Ferien. Mitdenken, -programmieren und -basteln ist natürlich erlaubt.

Um starten zu können, benötigen die Kinder einen Calliope mini, die App **Swift-Playground** und den Playground „Calliope mini Vorlage“ oder die **Calliope mini App**, um mit den Editoren **MakeCode** bzw. **Open Roberta Lab** zu programmieren.

Solltet ihr Hilfe benötigen, nutzt das **Calliope mini Handbuch für Lehrkräfte** und die Calliope Webseite. Dort gibt es Tipps und Kniffe als erste Hilfe und ein Glossar, das die wichtigsten Begriffe aus dem Bereich Coding erklärt.

Weiteres Bastelmaterial, das benötigt wird, ist jeweils aufgelistet.

Wir wünschen Euch viel Erfolg und Spaß bei der Umsetzung!



DIE CALLIOPE MINI UMGEBUNG



CALLIOPE MINI

Alles um das Projekt, hilfreiche Tipps, Online-Editoren, Schulmaterial und viele Projekte finden sich auf der Calliope Webseite.

Um zu starten, eignen sich die [Ersten Schritte](#) und [Erste Übungen](#) oder der [Calliope mini Swift Playground](#).

[CALLIOPE.CC](https://www.calliope.cc)



EDITOREN

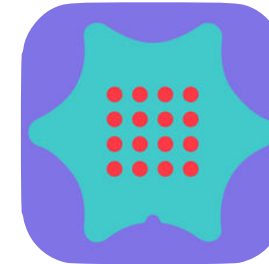
Mit den Editoren „**MakeCode**“ und „**Open Roberta Lab**“ können kreative Ideen mit dem Calliope mini umgesetzt werden.



[LAB.OPEN-ROBERTA.ORG](https://lab.open-roberta.org)

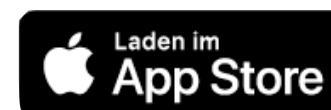


[MAKECODE.CALLIOPE.CC](https://makecode.calliope.cc)



CALLIOPE MINI APP

Die Calliope mini-App kann genutzt werden, um die benötigte Programmaktualisierung auf dem mini durchzuführen und um eigene Programme aus den Online-Editoren auf den Calliope mini zu übertragen.



SWIFT PLAYGROUNDS

Die **SwiftPlaygrounds App** wird auf jedem Gerät benötigt, um den Calliope mini Playground und die Vorlage nutzen zu können.



Das **Handbuch für Lehrkräfte** bietet einen angeleiteten Einstieg in das Programmieren mit dem Calliope mini.





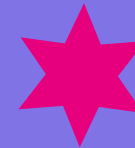
HO HO HO!

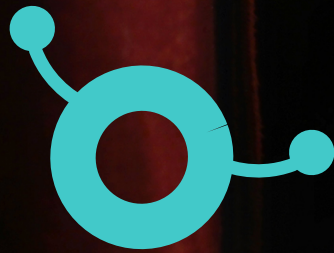
Es weihnachtet schon kräftig dort draußen und deswegen wird es Zeit für das erste Calliope mini-Mitmachbuch zu Weihnachten und Silvester!

In diesem Buch findet ihr coole Ideen für die kalte Jahreszeit.

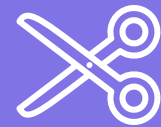


Viel Spaß und bleibt gesund!





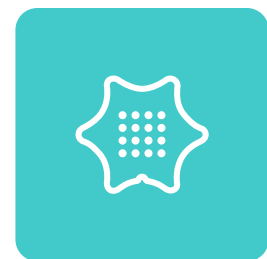
PROJEKTE



MATERIAL



Neben dem Programmieren lädt der Calliope mini auch zum Basteln von Prototypen und zum individuellen Gestalten von Projekten ein. Dazu eignen sich folgende Materialien:



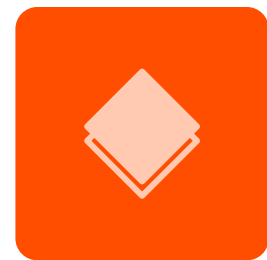
Calliope mini



Gummiband
(im Calliope mini-
Set enthalten)



Klebeband



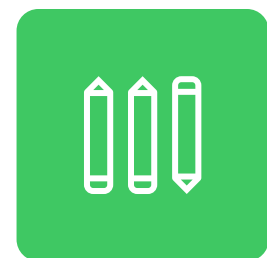
Transparentpapier



Schere



Pappe



Stifte



Papier



Acrylkugel



Aufkleber



PLAYGROUND

CODING-TAGEBUCH



PROJEKTE



Rudolph mit
der roten
Nase



0 Tannenbaum



Weihnachts-
baumkugel



Alarmanlage
für die
Keksdose



Neujahrs-
Orakel



Countdown





RUDOLPH MIT DER ROTEN NASE

MATERIAL

Tonpapier
Schere
Kleber
Buntstifte
Deko zum Verzieren

SCHWIERIGKEIT



FÜR DICH ZUM HERUNTERLADEN:

Bastelvorlage
Storyboard im Coding-
Tagebuch

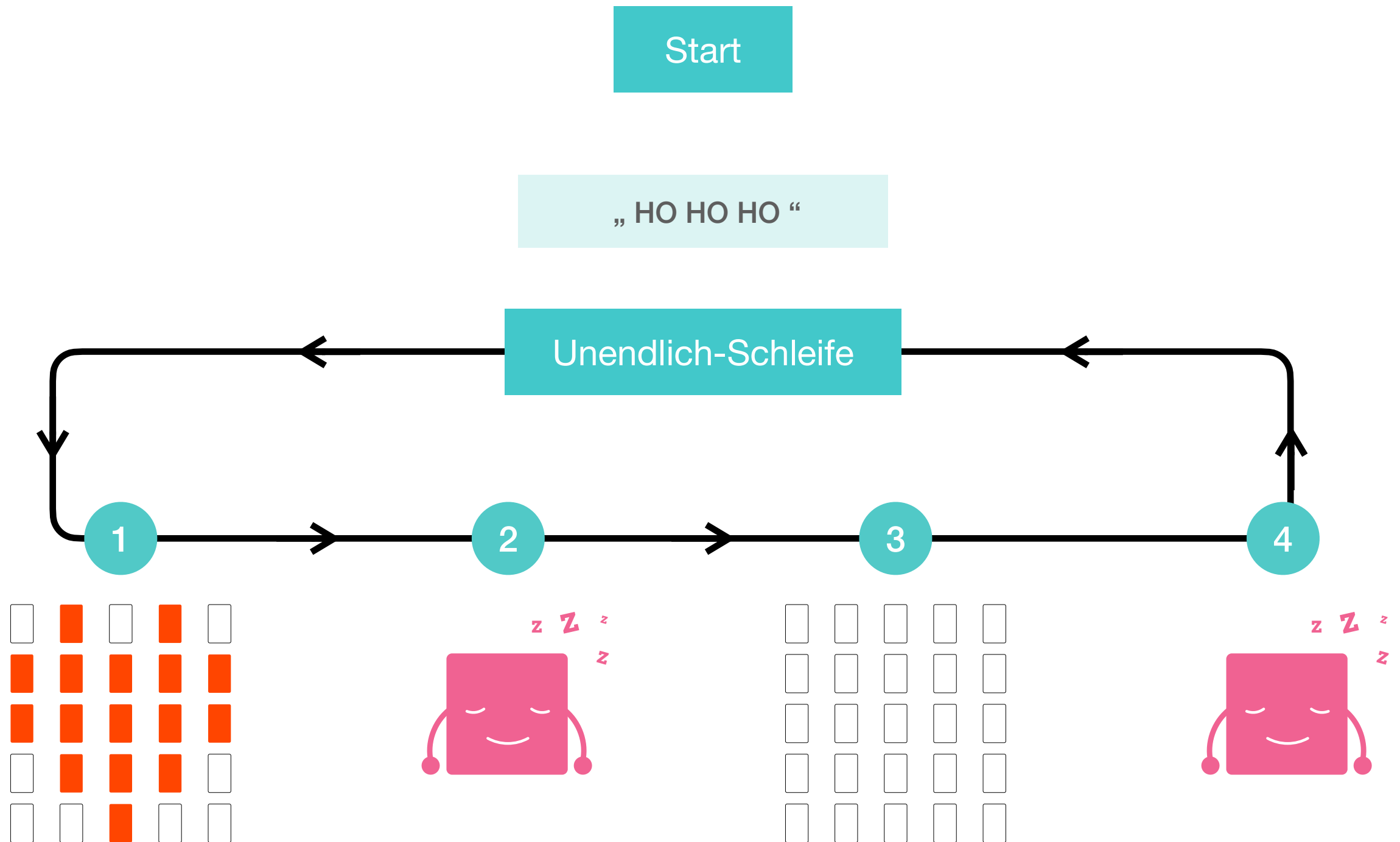


Um Himmels Willen! Es ist der 23.12.2020 und Rudolph hat einen schlimmen Schnupfen (Corona-Test ist negativ, kein Grund zur Beunruhigung)! Bei Rudolph wirkt sich eine Schnupfennase allerdings anders aus, als bei euch: Seine Nase hört auf, rot zu leuchten!

Helft dem Weihnachtsmann, das Weihnachtsfest zu retten, indem ihr für Rudolph eine blinkende rote Nase programmiert und euren Rudolph bastelt. Nehmt eine Geschichte mit Bildern in Keynote auf und beschreibt spannend, wie ihr es geschafft habt, das Weihnachtsfest gemeinsam zu retten!

Tipp: Nutzt für eure Ideen ein Storyboard, bevor ihr euch ans Aufnehmen macht!

Vielleicht habt ihr ja noch weitere Ideen, die RGB-LED und die LED-Matrix zu nutzen?





TIPP:

Der mini.sleep-Befehl sorgt dafür, dass der Text bis zum Ende angezeigt wird. Wenn dein Text länger sein sollte, dann musst du den mini.sleep verlängern. Dann wird dein ganzer Text angezeigt und läuft über das LED-Feld.

PLAYGROUND

Ausgabe:
RGB
Text
Symbole

Funktionen:
Start Funktion
Pause



PLAYGROUND

```
func start (){  
  display.show(text:"H0 H0 H0")  
  mini.sleep (6000)  
}
```

Starttext

```
func forever (){  
  display.show(image:.full)  
  mini.sleep (1000)  
  display.clear()  
  mini.sleep (1000)  
}
```

Blinken

TIPP:

Der mini.sleep-Befehl sorgt dafür, dass der Text bis zum Ende angezeigt wird. Wenn dein Text länger sein sollte, dann musst du den mini.sleep verlängern. Dann wird dein ganzer Text angezeigt und läuft über das LED-Feld.

PROGRAMM:

mini-Rudolph.xml

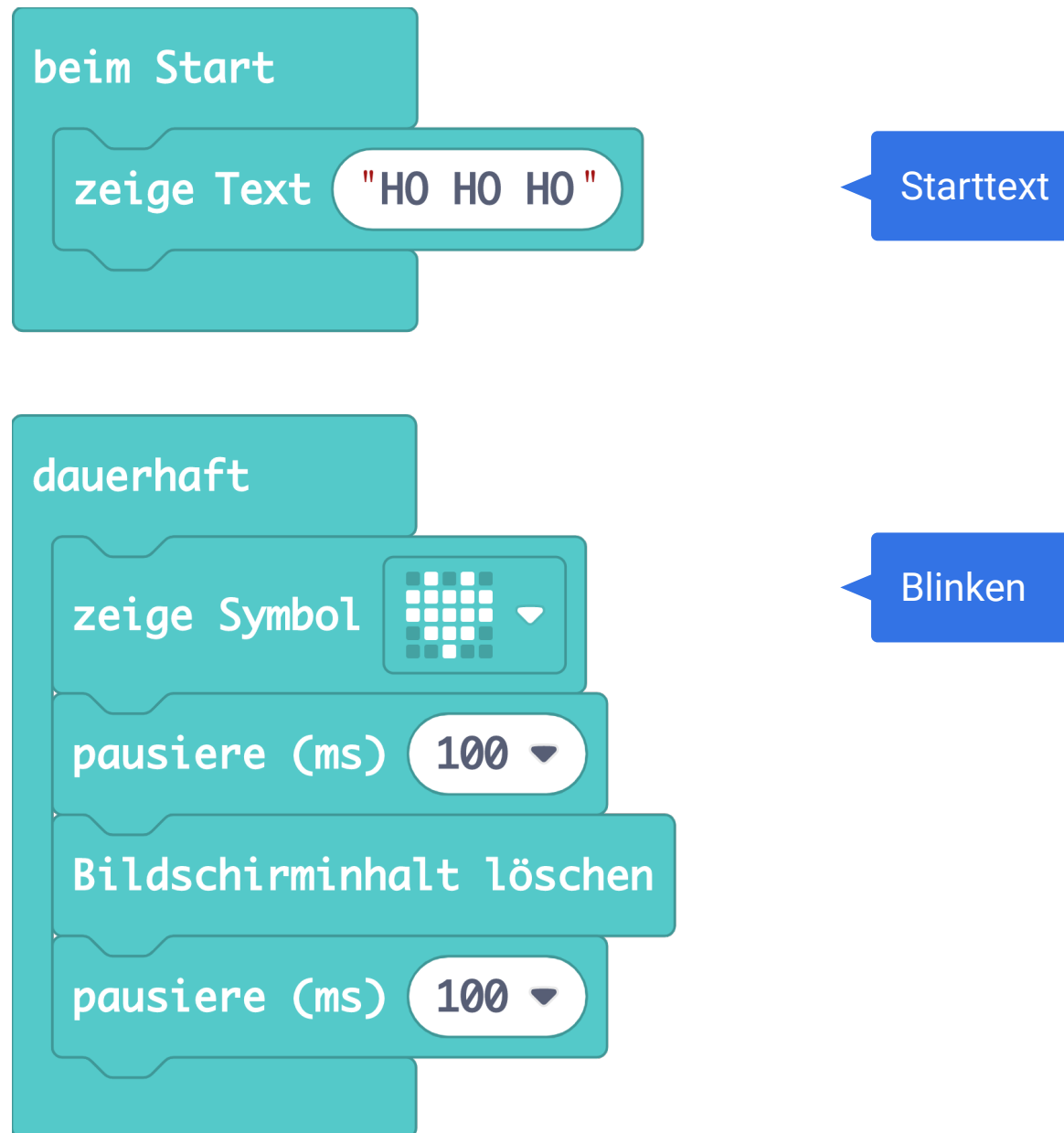


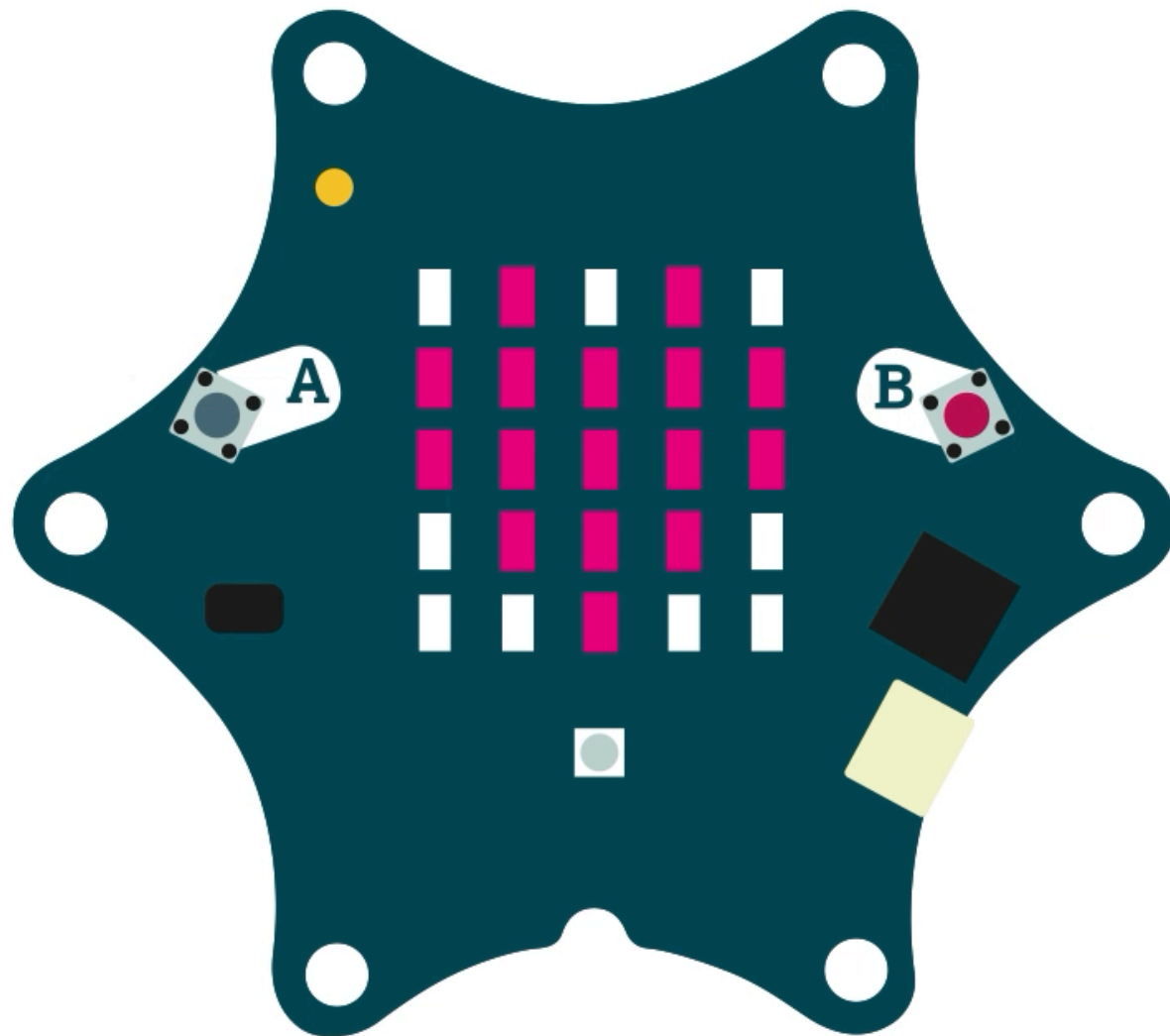
TIPP:

Der mini.sleep-Befehl sorgt dafür, dass der Text bis zum Ende angezeigt wird. Wenn dein Text länger sein sollte, dann musst du den mini.sleep verlängern. Dann wird dein ganzer Text angezeigt und läuft über das LED-Feld.

PROGRAMM:

[mini-Rudolph.hex](#)





Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

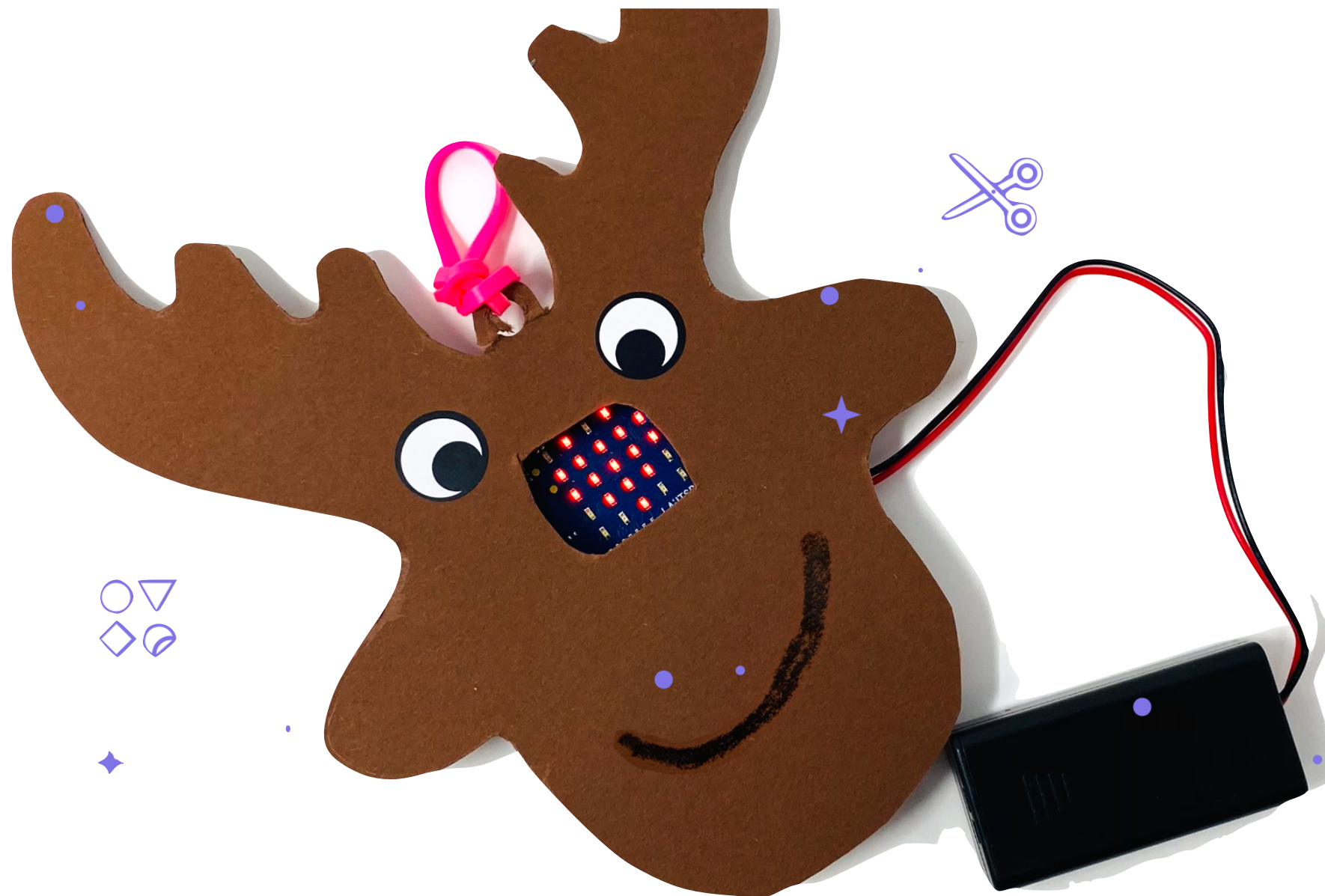
Folgende Fragestellungen helfen beim Testen:

- Läuft der Starttext bis zum Ende durch?
- Blinkt die LED-Matrix im gewünschten Rhythmus?
- Leuchten die LEDs lang genug?
- Ist eure Form zu erkennen? Habt ihr noch eine Idee für eine Nasenform?
- Habt ihr noch weitere Ideen, die RGB-LED mit einzubeziehen?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



So könnte euer Rudolph aussehen.



0 TANNENBAUM

02



MATERIAL

Tonpapier
Schere
Kleber
Buntstifte
Deko zum Verzieren

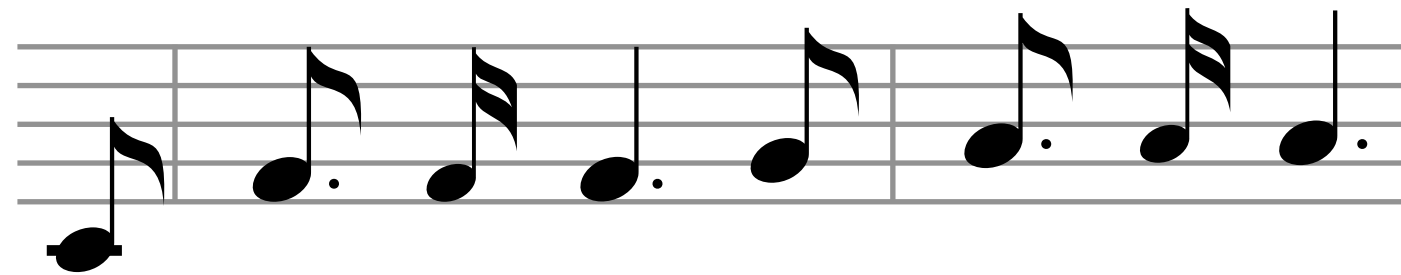
SCHWIERIGKEIT



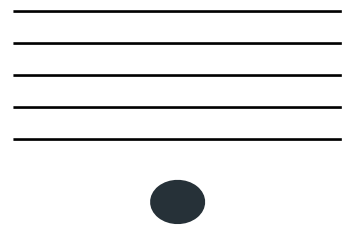
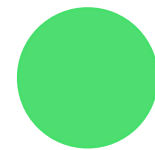
FÜR DICH ZUM
HERUNTERLADEN:
Bastelvorlage



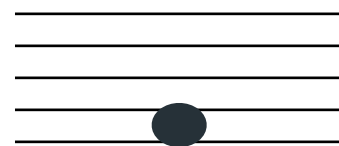
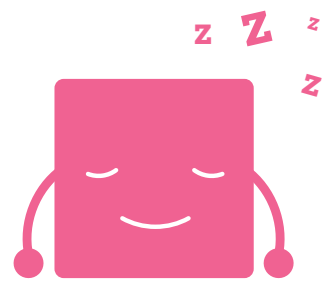
Lasst den Calliope mini euer Lieblingslied singen.



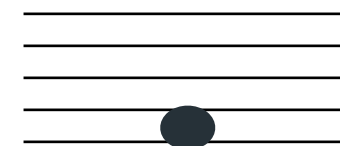
Start



C



F



F



TIPP:

Achtel-, Viertel- und ganze Noten lassen sich über die mini.sleep-Funktion abspielen.
1000 mini.sleep=ganze Note
500 mini.sleep=halbe Note
250 mini.sleep=Viertel-note

PLAYGROUND

Ausgabe:
RGB
Lautsprecher

Funktionen:
Start Funktion
Pause



PLAYGROUND

```
func start (){  
  rgb.on(color: .green)  
  sound.on(note:.C)  
  mini.sleep (1000)  
  sound.on(note:.F)  
  mini.sleep (250)  
  sound.on(note:.F)  
  mini.sleep (250)  
  sound.on(note:.F)  
  mini.sleep (1000)  
  sound.on(note:.G)  
  mini.sleep (500)  
  sound.on(note:.A)  
  mini.sleep (250)  
  sound.on(note:.A)  
  mini.sleep (250)  
  sound.on(note:.A)  
  mini.sleep (500)  
  sound.off()  
  rgb.off()  
}
```

Pause

TIPP:

Die Notenlänge kann für den Ton definiert werden.

Die Pausen helfen dabei, die Melodie besser wiederzugeben.

PROGRAMM:

[mini-Song.hex](#)



```
beim Start
  setze RGB-LED-Farbe auf 
  spiele Note Mittleres C für 1/2 Schlag
  pausiere (ms) 250
  spiele Note Mittleres F für 1/4 Schlag
  pausiere (ms) 250
  spiele Note Mittleres F für 1/4 Schlag
  pausiere (ms) 250
  spiele Note Mittleres F für 1/4 Schlag
  pausiere (ms) 250
  spiele Note Mittleres G für 1/2 Schlag
  pausiere (ms) 250
  spiele Note Mittleres A für 1/2 Schlag
  pausiere (ms) 250
  spiele Note Mittleres A für 1/4 Schlag
  pausiere (ms) 250
  spiele Note Mittleres A für 1/2 Schlag
```

Notenlänge

Pause

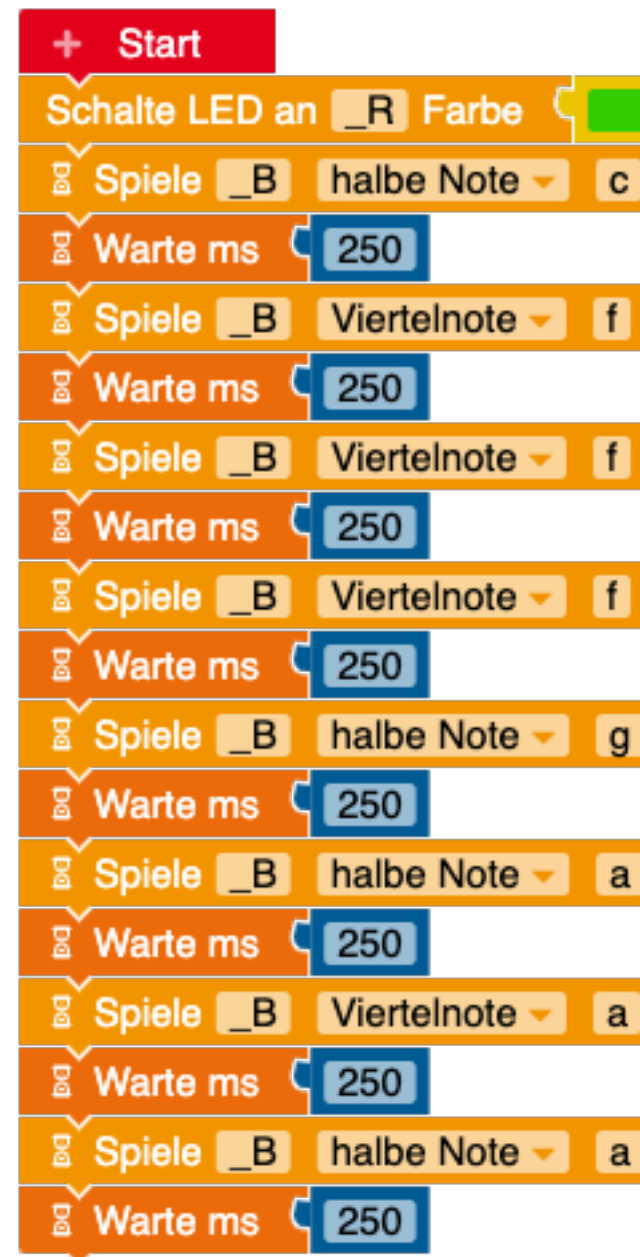


TIPP:

Die Notenlänge kann für den Ton definiert werden. Die Pausen helfen dabei, die Melodie besser wiederzugeben.

PROGRAMM:

mini-Song.xml

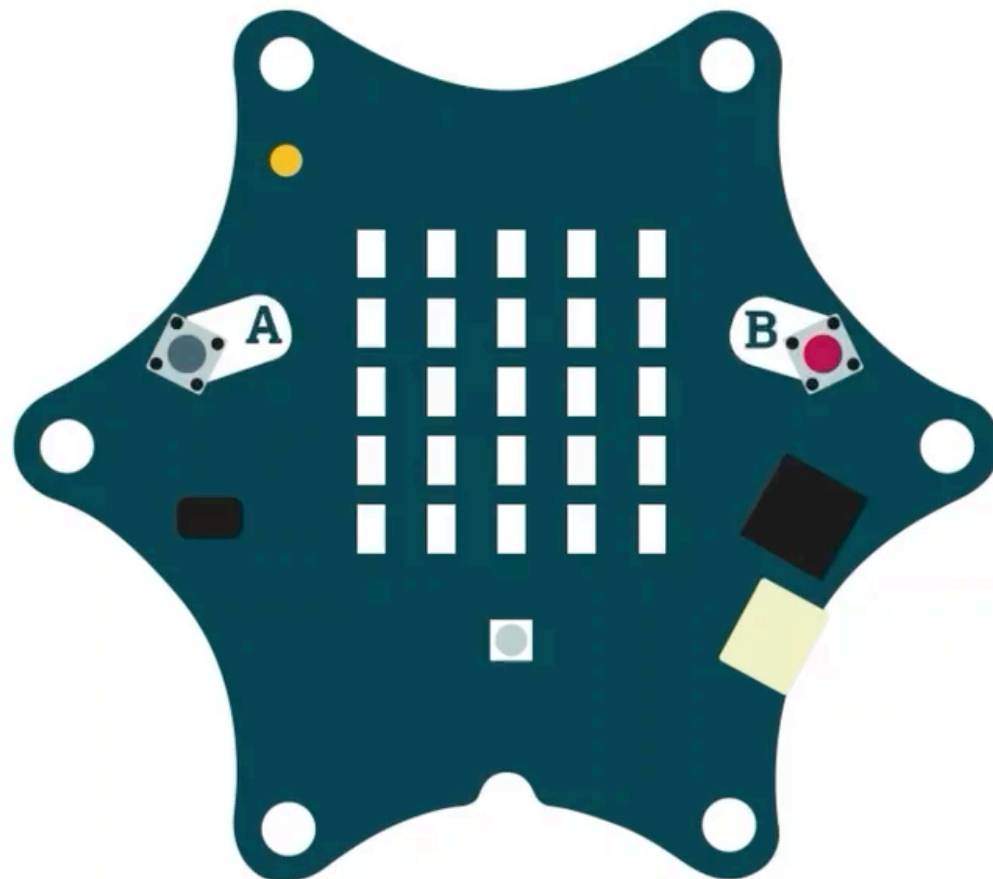


```
+ Start
Schalte LED an _R Farbe
  Spiele _B halbe Note c
  Warte ms 250
  Spiele _B Viertelnote f
  Warte ms 250
  Spiele _B Viertelnote f
  Warte ms 250
  Spiele _B Viertelnote f
  Warte ms 250
  Spiele _B halbe Note g
  Warte ms 250
  Spiele _B halbe Note a
  Warte ms 250
  Spiele _B Viertelnote a
  Warte ms 250
  Spiele _B halbe Note a
  Warte ms 250
```

Notenlänge

Pause





Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Werden alle Töne nacheinander abgespielt?
- Sind die Pausen zwischen den einzelnen Tönen lang genug oder eventuell zu lang?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



So könnte euer klingender
• mini-Tannenbaum
aussehen.



CALLIOPE MINI WEIHNACHTSBAUM

Macht aus eurem Weihnachtsbaum einen Calliope mini-Weihnachtsbaum! In diesem Projekt werden wir die unterschiedlichen Lichter des Calliope mini nutzen, sodass er wie eine bunte Weihnachtskugel leuchtet. Sobald es dunkel genug ist, blinken die roten Lichter am Tannenbaum auf.

Welche anderen Möglichkeiten gibt es, um die Lichter am Baum erstrahlen zu lassen?

Habt ihr neben Weihnachtskugeln noch andere Ideen?

MATERIAL

Tonpapier
Schere
Kleber
Buntstifte
Deko zum Verzieren

SCHWIERIGKEIT



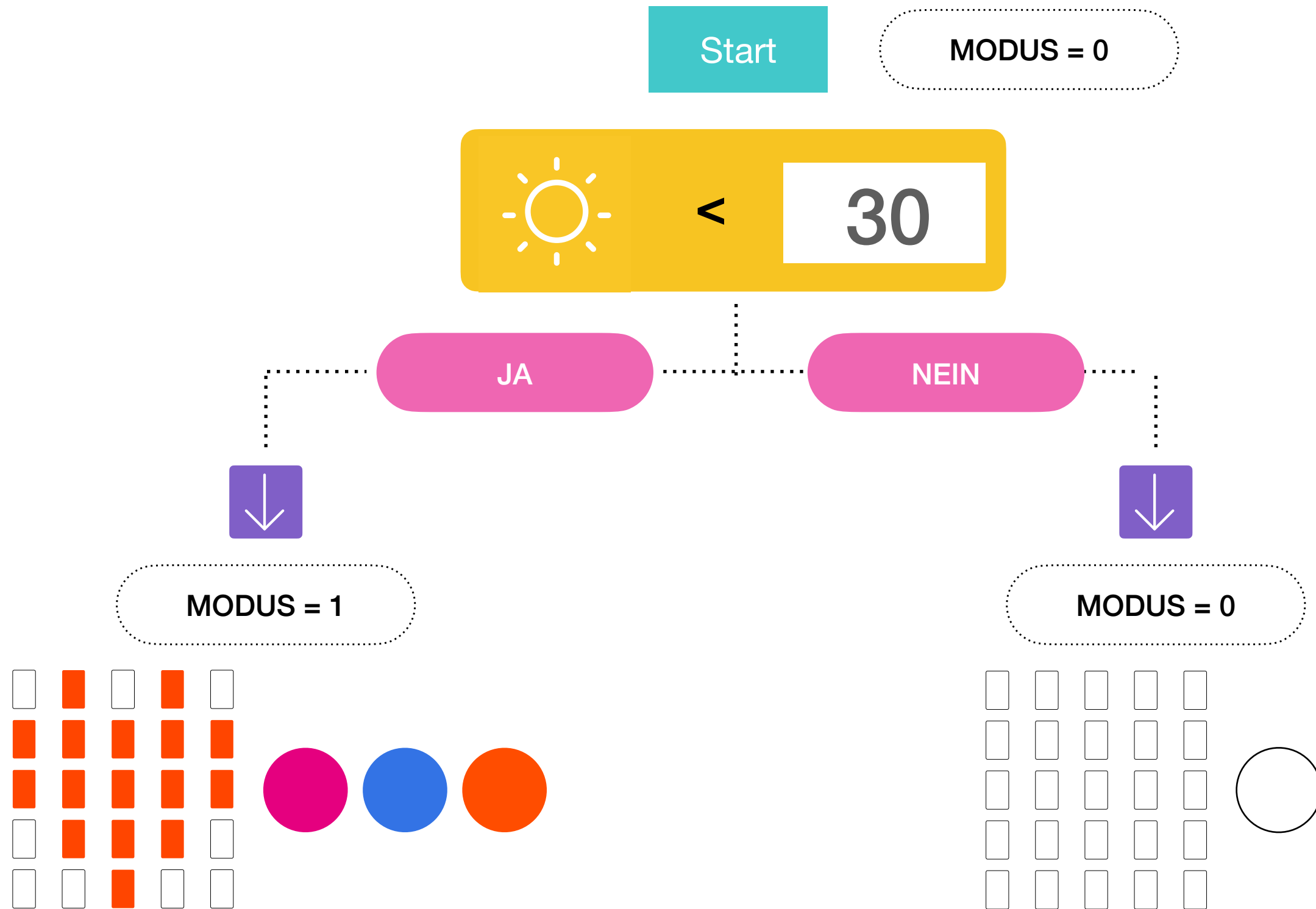
FÜR DICH ZUM HERUNTERLADEN:

Bastelvorlage



PROGRAMMABLAUFPLAN

03





TIPP:

Die Variable `modus` ist hier eine Art Schalter. Je nach Helligkeit wird der Wert entweder auf 1 oder 0 gestellt um so die Weihnachtskugel ein- bzw. auszuschalten.

PLAYGROUND

Ausgabe:
Kombination

Funktionen:
Start Funktion
Pause
Löschen
Variablen
Wenn... dann



PLAYGROUND

```
public var modus = 0
func forever() {
  if io.brightness < 30{
    modus = 1
  }else{
    modus = 0
  }
  if modus == 1{
    display.show(image: .heart)
    rgb.on(color:.blue)
    mini.sleep(200)
    rgb.on(color:.magenta)
    display.clear()
    mini.sleep(200)
    display.show(image: .heart)
    rgb.on(color:.orange)
    mini.sleep(200)
    display.clear()
  }else{
    display.clear()
    rgb.off()
    mini.sleep(200)
  }
}
```

Helligkeit wird abgefragt

hell

dunkel

TIPP:

Die Variable modus ist hier eine Art Schalter. Je nach Helligkeit wird der Wert entweder auf 1 oder 0 gestellt um so die Weihnachtskugel ein- bzw. auszuschalten.

PROGRAMM:

PROGRAMM:
mini-Kugel.xml



Helligkeit wird abgefragt

hell

dunkel



TIPP:

Die Variable `modus` ist hier eine Art Schalter. Je nach Helligkeit wird der Wert entweder auf 1 oder 0 gestellt um so die Weihnachtskugel ein- bzw. auszuschalten.

PROGRAMM:
[mini-Kugel.hex](#)



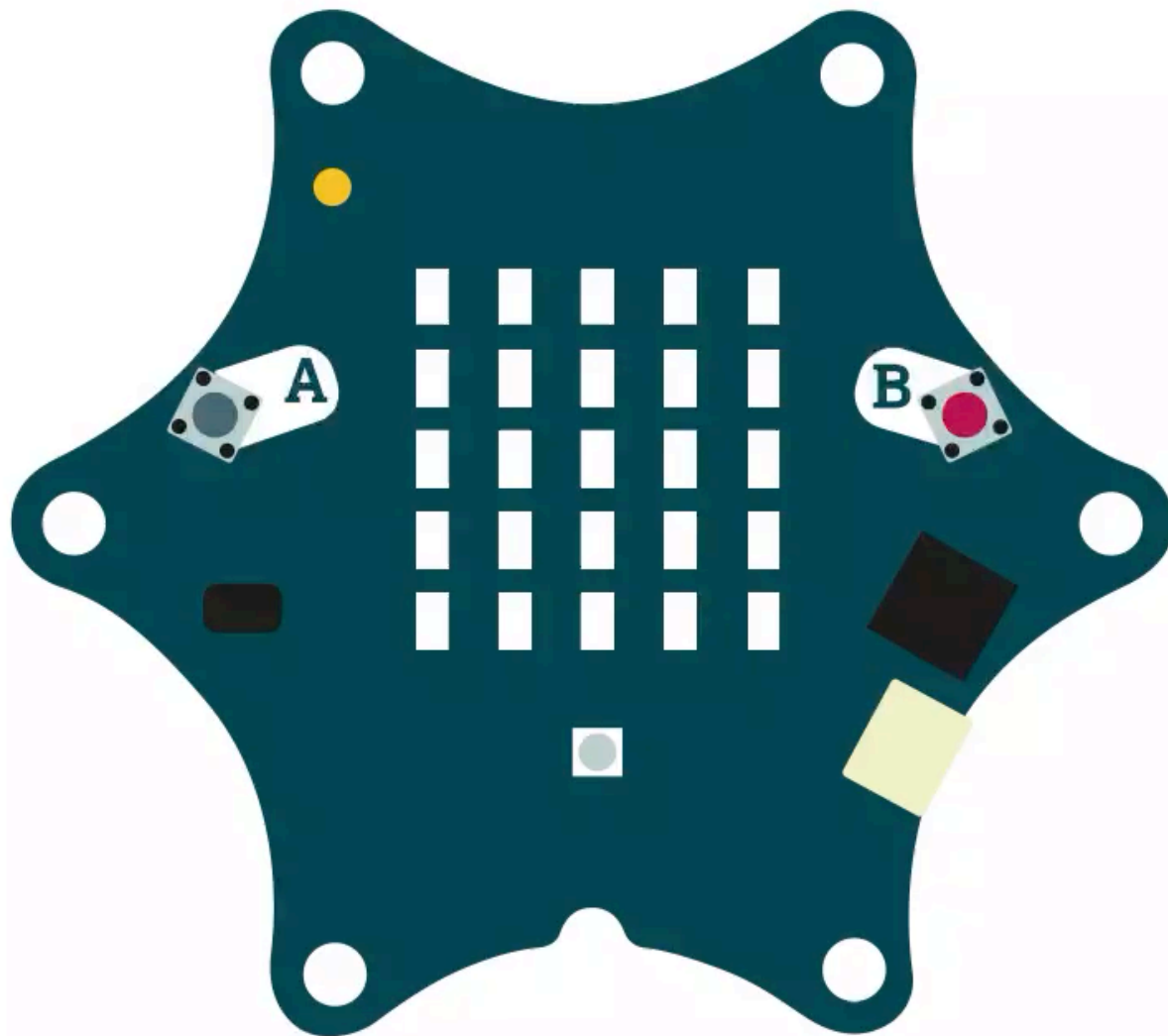
```
beim Start
  zeige Zahl Lichtstärke
  setze modus auf 0
```

```
dauerhaft
  wenn Lichtstärke < 30 dann
    setze modus auf 1
  ansonsten
    setze modus auf 0
  wenn modus = 1 dann
    zeige Symbol [Weihnachtskugel]
    setze RGB-LED-Farbe auf [blau]
    pausiere (ms) 100
    Bildschirminhalt löschen
    pausiere (ms) 100
    zeige Symbol [Weihnachtskugel]
    setze RGB-LED-Farbe auf [orange]
    pausiere (ms) 100
    Bildschirminhalt löschen
  ansonsten
    Bildschirminhalt löschen
    eingebaute RGB-LED ausschalten
```

Helligkeit wird abgefragt

hell

dunkel

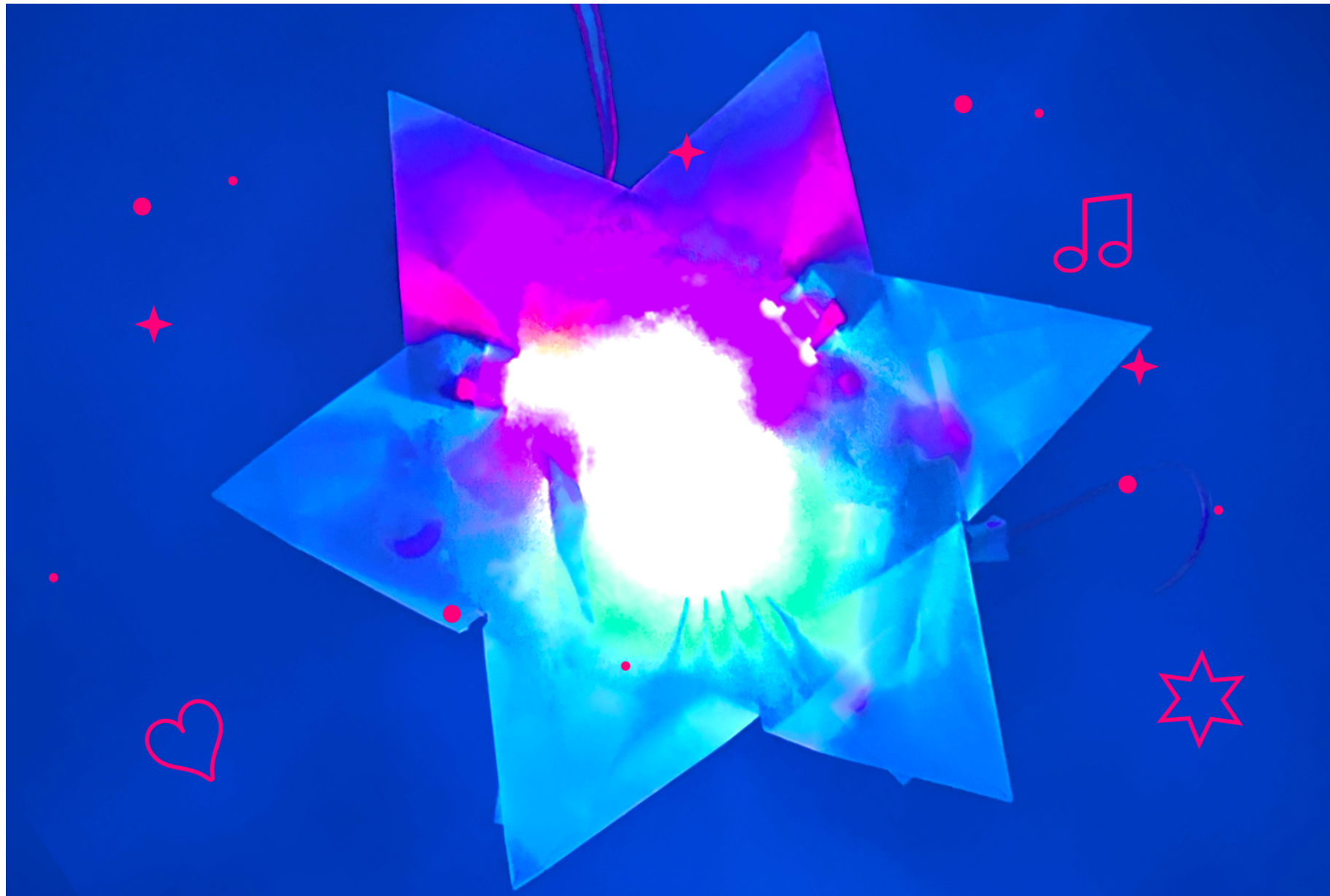


Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Zum Testen bietet sich an, die Hand über den Calliope mini zu halten. Reagiert er auf die unterschiedlichen Helligkeitswerte?
- Blinkt die RGB-LED, sobald es dunkel wird?
- Ist alles wieder aus, sobald es hell wird?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



So könnte euer blinkender mini-Baumschmuck aussehen.

SCHWIERIGKEIT



ALARMANLAGE FÜR DIE KEKSDOSE

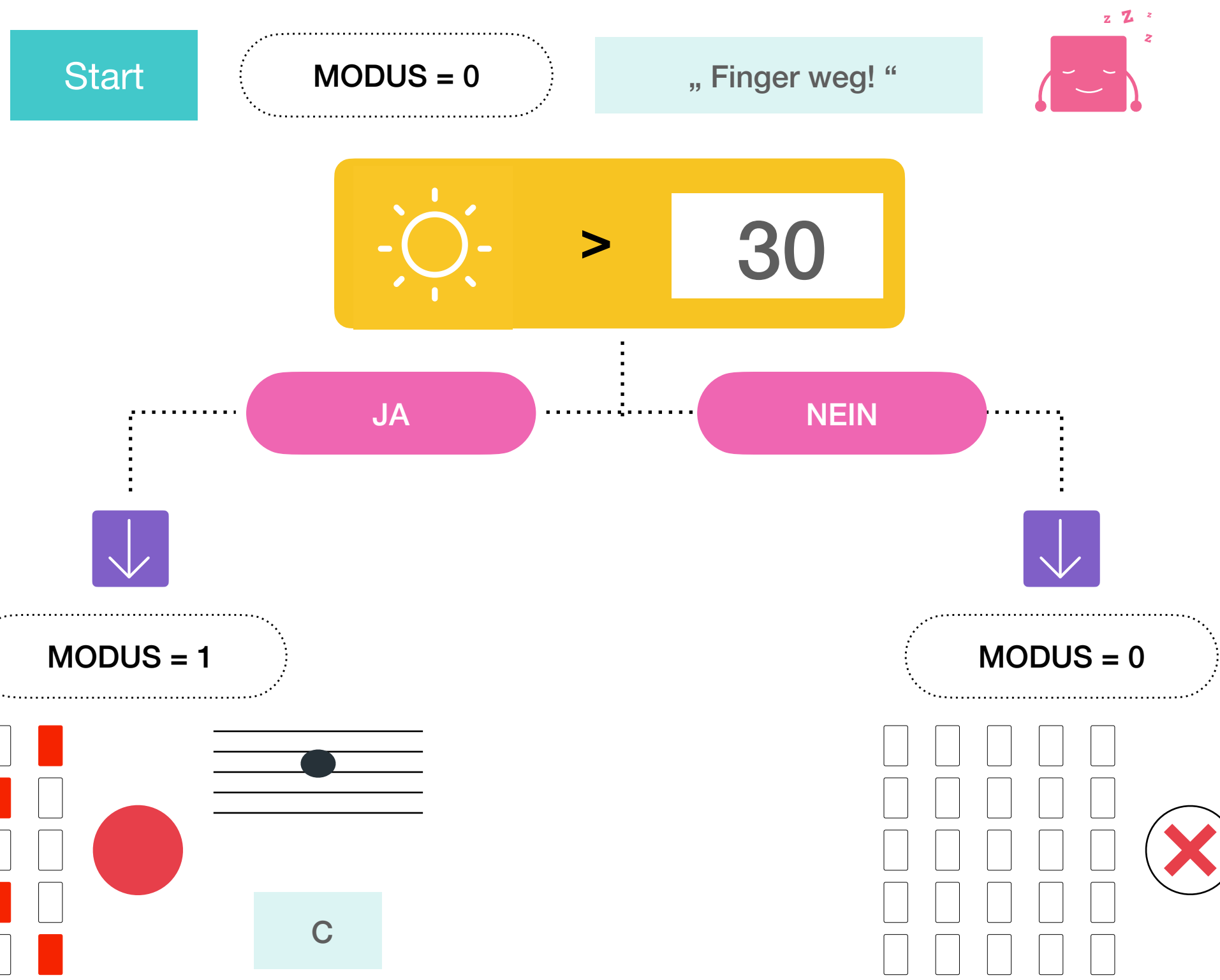
04



Da hat sich doch schon wieder jemand heimlich an den Weihnachtsplätzchen bedient! Programmiert eine Keksdose mit Alarmanlage!

PROGRAMMABLAUFPLAN

04





TIPP:

Die Variable `modus` ist hier eine Art Schalter. Das Ändern des Wertes zwischen 1 und 0 ermöglicht es, den Alarm an- und auszuschalten.

PLAYGROUND

Ausgabe:
Kombination

Funktionen:
Start Funktion
Pause
Löschen
Variablen
Wenn... dann



PLAYGROUND

```
public var modus = 0
func start(){
  display.show(text:"5,4,3,2,1")
  mini.sleep(5000)
  rgb.off()
  sound.off()
  display.clear()
}
func forever() {
  if io.brightness > 30{
    modus = 1
  }
  if modus == 1{
    rgb.on(color:.red)
    sound.on(note: .A)
    display.show(image: .scissors)
    mini.sleep(200)
    display.clear()
    display.show(image: .scissors)
    mini.sleep(200)
  }else{
    rgb.off()
    sound.off()
    display.clear()
    mini.sleep(2000)
  }
}
func onButtonAB() {
  modus = 0
}
```

Startanimation

Helligkeit wird abgefragt

hell

dunkel

Alarm ausschalten

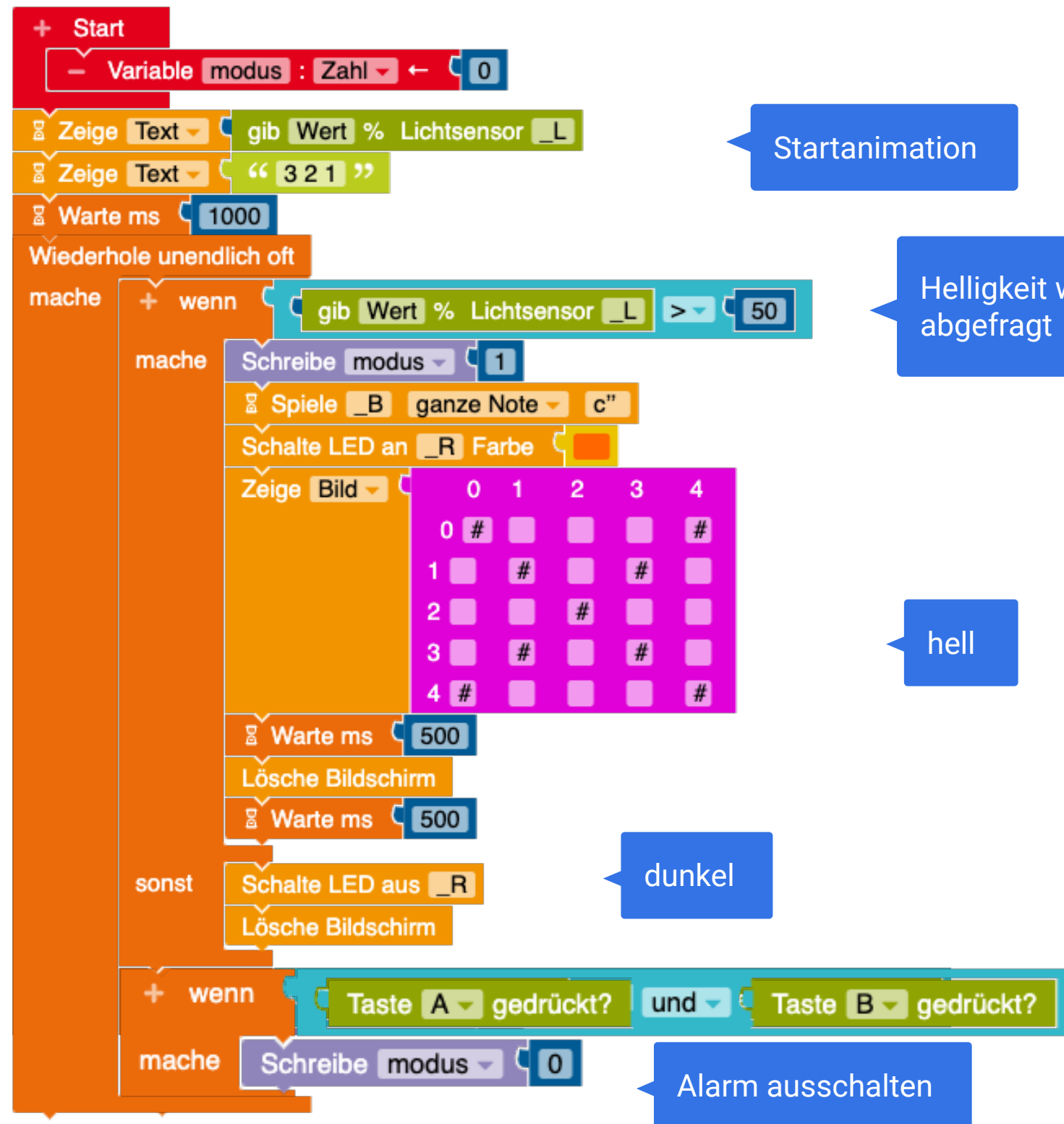


TIPP:

Die Variable modus ist hier eine Art Schalter. Das Ändern des Wertes zwischen 1 und 0 ermöglicht es, den Alarm an- und auszuschalten.

PROGRAMM:

mini-Keksalarm.xml



Startanimation

Helligkeit wird abgefragt

hell

dunkel

Alarm ausschalten



TIPP:

Die Variable `modus` ist hier eine Art Schalter. Das Ändern des Wertes zwischen 1 und 0 ermöglicht es, den Alarm an- und auszuschalten.

PROGRAMM:

[mini-Keksalarm.hex](#)



```
beim Start
  zeige Zahl Lichtstärke
  setze modus auf 0
  zeige Text "3 2 1"
  pausiere (ms) 1000
```

Startanimation

```
wenn Knopf A+B gedrückt
  setze modus auf 0
```

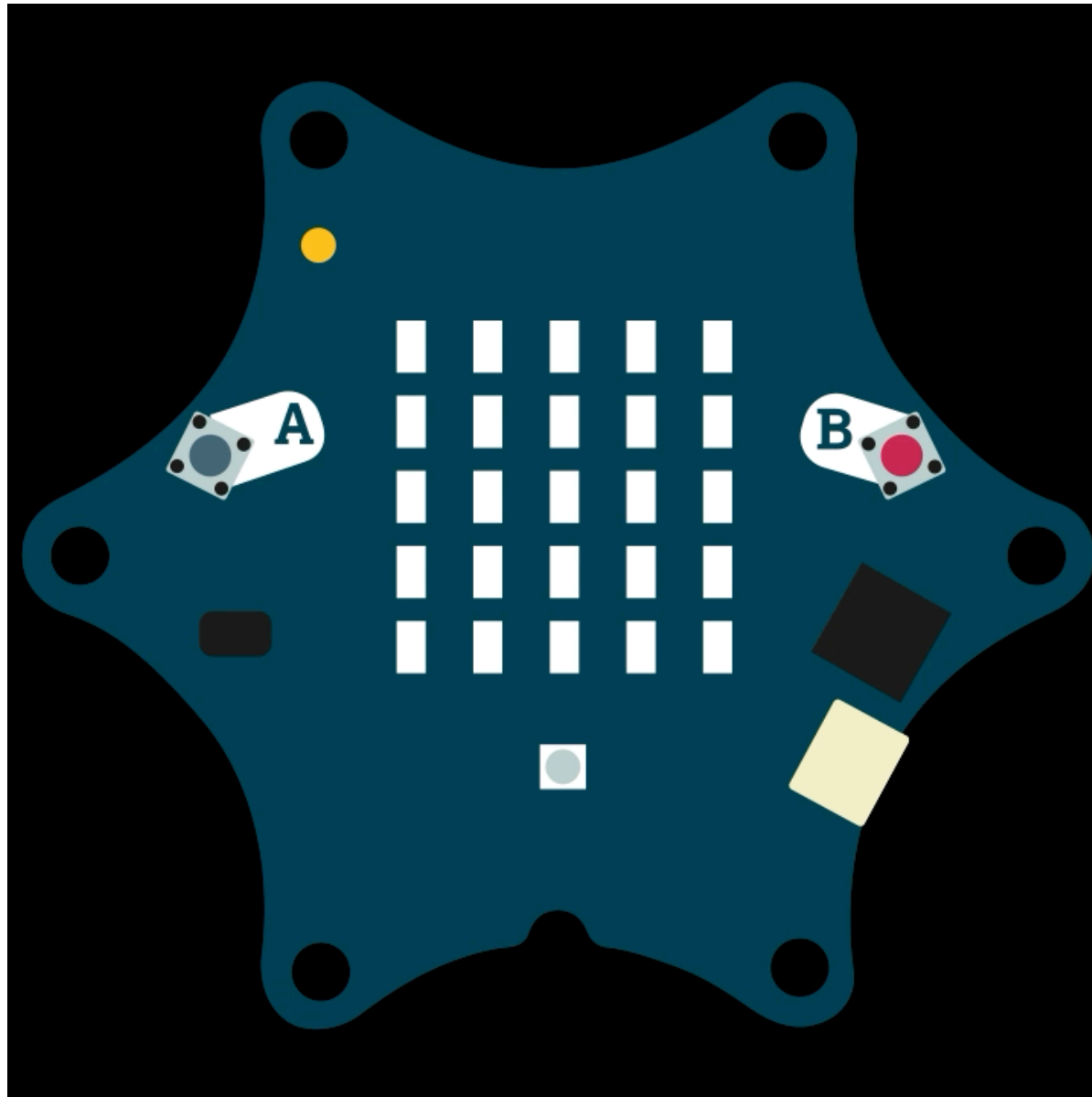
Alarm ausschalten

```
dauerhaft
  wenn Lichtstärke > 30 dann
    setze modus auf 1
  +
  wenn modus = 1 dann
    setze RGB-LED-Farbe auf [rot]
    Klingelton (Hz) Mittleres C
    zeige LEDs
    pausiere (ms) 100
    Bildschirminhalt löschen
    pausiere (ms) 100
  -
  ansonsten
    Klingelton (Hz) 0
    Bildschirminhalt löschen
    eingebaute RGB-LED ausschalten
  +
```

Helligkeit wird abgefragt

hell

dunkel



Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Wird die Startanimation angezeigt?
- Reicht die Zeit für die Positionierung des Calliope mini im Versteck?
- Zum Testen bietet sich an, die Hand über den Calliope mini zu halten. Reagiert er auf die unterschiedlichen Helligkeitswerte?
- Ertönt ein Ton, sobald es hell wird?
- Blinkt die RGB-Matrix, sobald es hell wird?
- Ist alles wieder aus, sobald die Tasten A und B gemeinsam gedrückt wurden?
- Welche Töne eignen sich besonders für eine Alarmanlage?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



Wie sieht eure Keksdose aus?

Vorsicht mini-gesichert!





NEUJAHR-S-ORAKEL

MATERIAL

Acrylkugel
Schere
Kleber
Buntstifte
Deko zum Verzieren

SCHWIERIGKEIT



FÜR DICH ZUM HERUNTERLADEN:

Bastelanleitung
Rezept

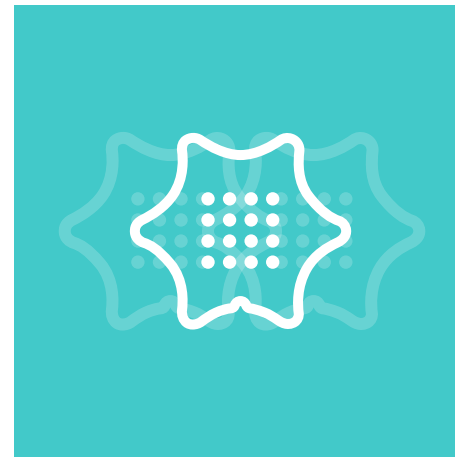


Das Calliope mini-Orakel wird gern und häufig gebaut.
Nun kommt das Orakel für das neue Jahr **2021!**

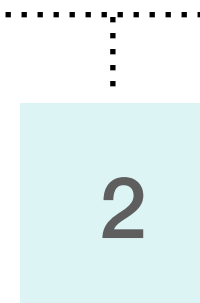
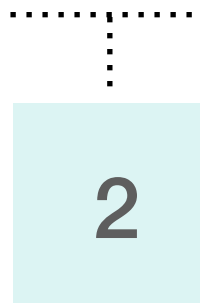
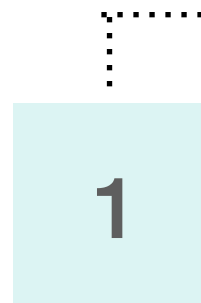
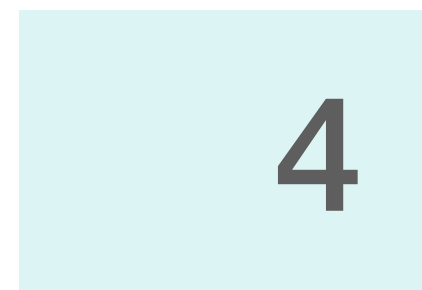
Hier wird der Kreativität keine Grenze gesetzt. Die Anzahl der verschiedenen Antwortmöglichkeiten ist unbegrenzt. Überlegt euch noch weitere Symbole, Farben oder Bilder über das LED-Feld.

Hier ein paar Tipps für positive und lustige Sprüche zum neuen Jahr:

- Bleib gesund!
- Keep cool und lüfte stoß!
- Dich erwartet eine Überraschung.
- Genieß den Augenblick.
- 2021 wird ein tolles Jahr!



...



„Glücklich“

„Aufregend“

„Großartig“

„Fantastisch“



TIPP

Für jede mögliche zufällige Zahl benötigst du eine Antwort.

Zufällige Zahl

```
func onShake() {  
  let zufall = random(1...5)  
  if zufall == 1{  
    display.show(text: "Sei positiv und bleib gesund!")  
  }else if zufall == 2{  
    display.show(text: "Keep cool und lüfte stoß!")  
  }else if zufall == 3{  
    display.show(text: "Dich erwartet eine Überraschung.")  
  }else if zufall == 4{  
    display.show(text: "Genieß den Augenblick.")  
  }else if zufall == 5{  
    display.show(text: "2021 wird ein tolles Jahr!")  
  }  
}
```



PLAYGROUND

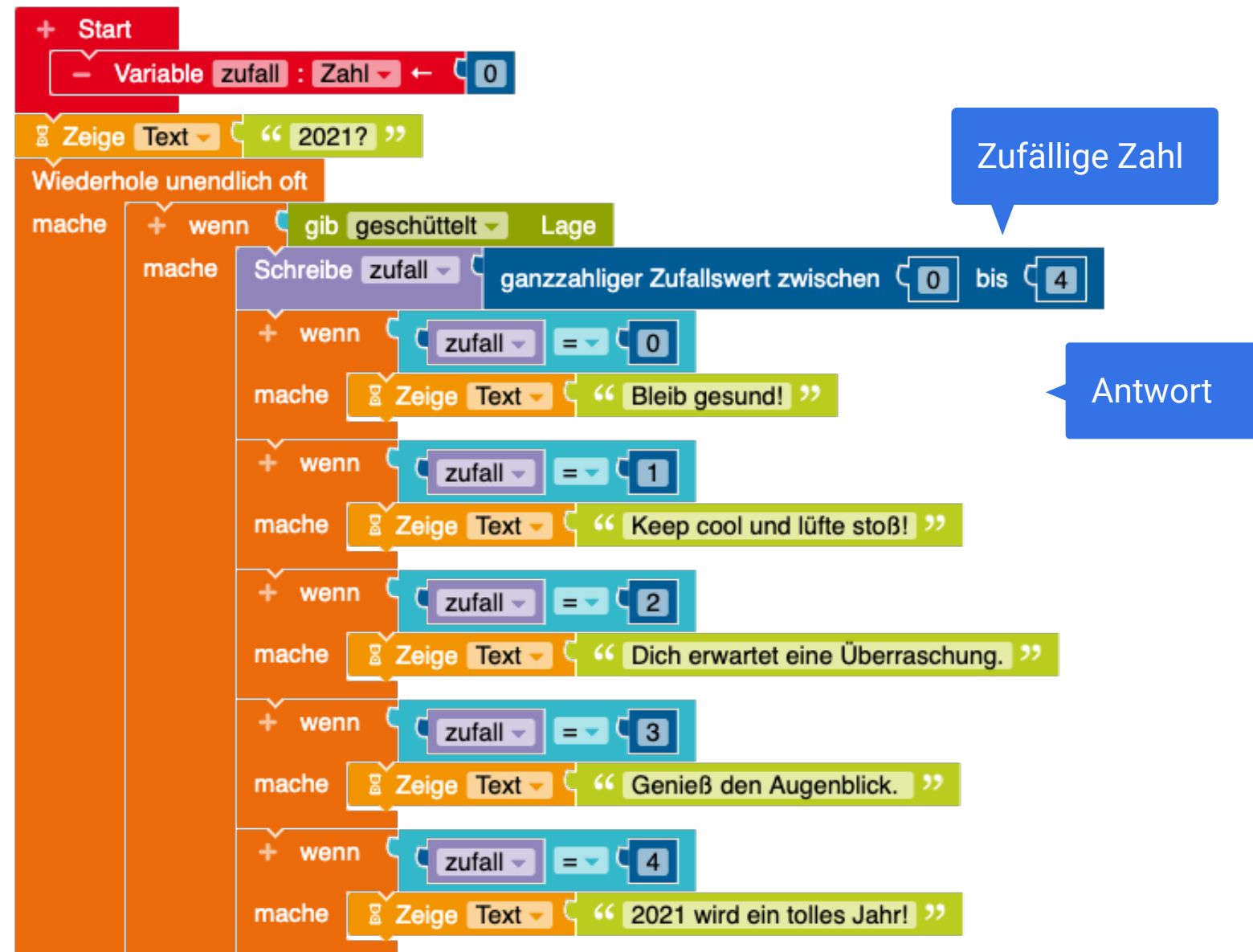


TIPP:

Für jede mögliche zufällige Zahl benötigst du eine Antwort.

PROGRAMM:

PROGRAMM:
NEPO-Neujahrs-
Orakel.xml



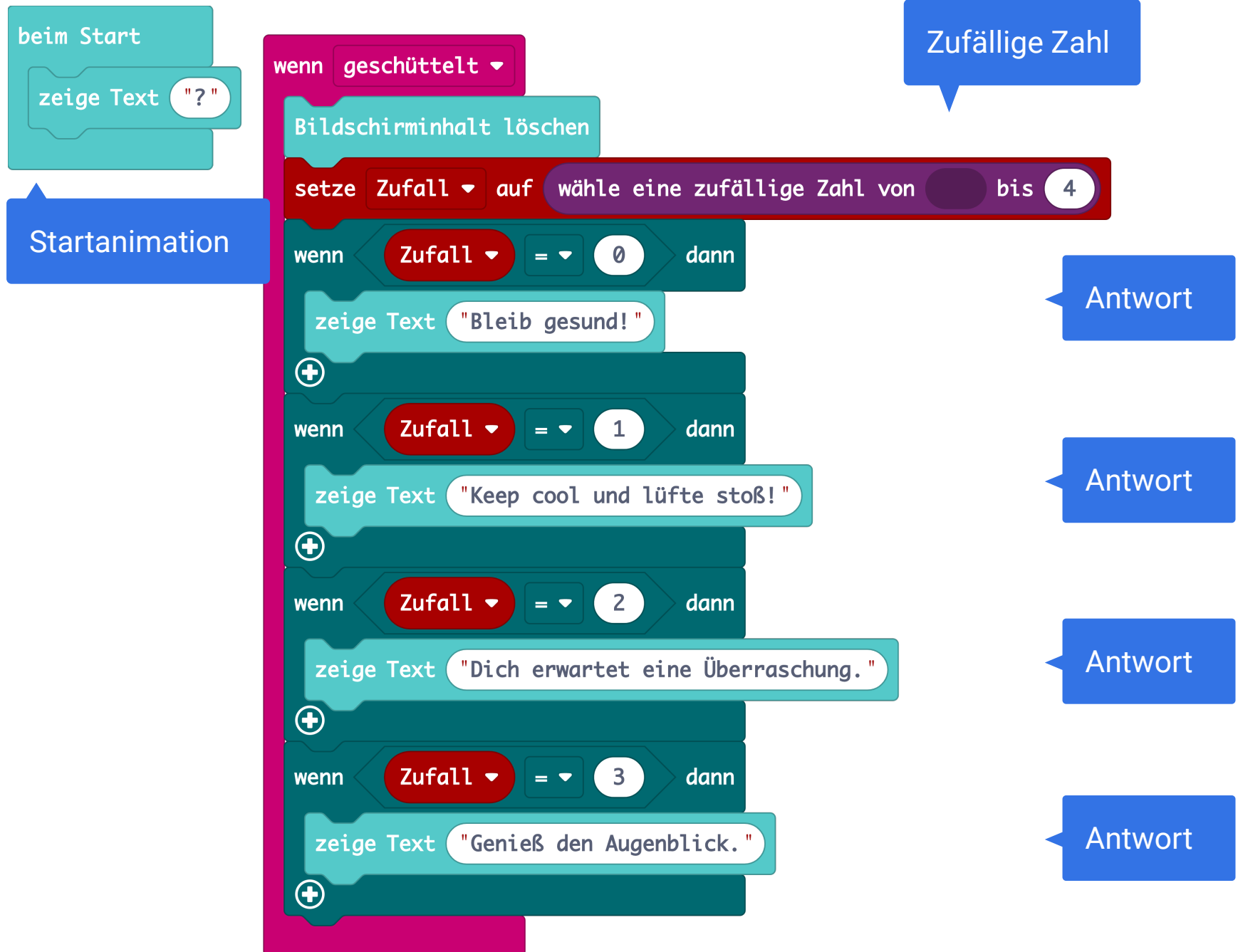


TIPP:

Für jede mögliche zufällige Zahl benötigst du eine Antwort.

PROGRAMM:

mini-Neujahrs-Orakel.hex



TIPP:

Für jede mögliche zufällige Zahl benötigst du ein Symbol und eine Farbe.

Fallen dir noch andere Symbole ein?
Lass die LED-Matrix leuchten!



PLAYGROUND

```
func onShake() {  
    let zufall = random(1...4)  
    if zufall == 1{  
        rgb.on(color: .orange)  
        display.show(image:.flash )  
    }else if zufall == 2{  
        rgb.on(color: .magenta)  
        display.show(image:.heart )  
    }else if zufall == 3{  
        rgb.on(color: .green)  
        display.show(grid:  
[0,0,1,1,1,0,0,0,1,1,0,0,1,0,1,0,1,0,0,0,1,0,0,0,0] )  
    }else if zufall == 4{  
        rgb.on(color: .cyan)  
        display.show(image:.scissors )  
    }  
}
```

Zufällige Zahl

Antwort - Farbe und Symbol

Antwort - eigenes Symbol

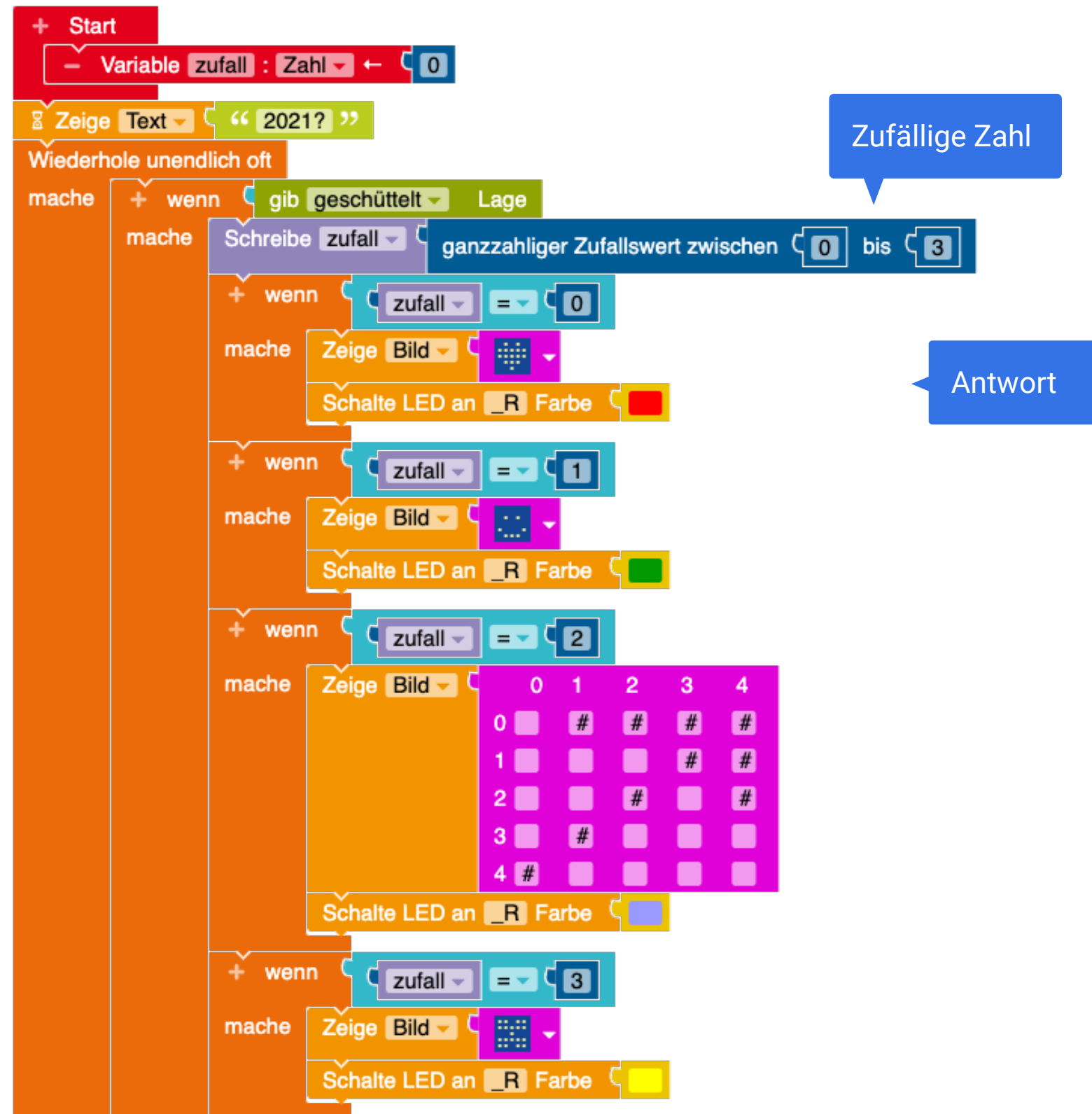


TIPP:

Für jede mögliche zufällige Zahl benötigst du eine Antwort.

PROGRAMM:

PROGRAMM:
NEPO-Neujahrs-
Orakel.xml





TIPP:

Für jede mögliche zufällige Zahl benötigst du ein Symbol und eine Farbe.

PROGRAMM:
mini-Neujahrs-Orakel-Symbole.hex



```
beim Start
  zeige Text "2021?"
```

Startanimation

```
wenn geschüttelt
  Bildschirminhalt löschen
  setze Zufall auf wähle eine zufällige Zahl von bis 4
  wenn Zufall = 0 dann
    zeige Symbol [Symbol]
    setze RGB-LED-Farbe auf [Rot]
  +
  wenn Zufall = 1 dann
    zeige Symbol [Symbol]
    setze RGB-LED-Farbe auf [Blau]
  +
  wenn Zufall = 2 dann
    zeige Symbol [Symbol]
    setze RGB-LED-Farbe auf [Magenta]
  +
  wenn Zufall = 3 dann
    zeige Symbol [Symbol]
    setze RGB-LED-Farbe auf [Grün]
  +
```

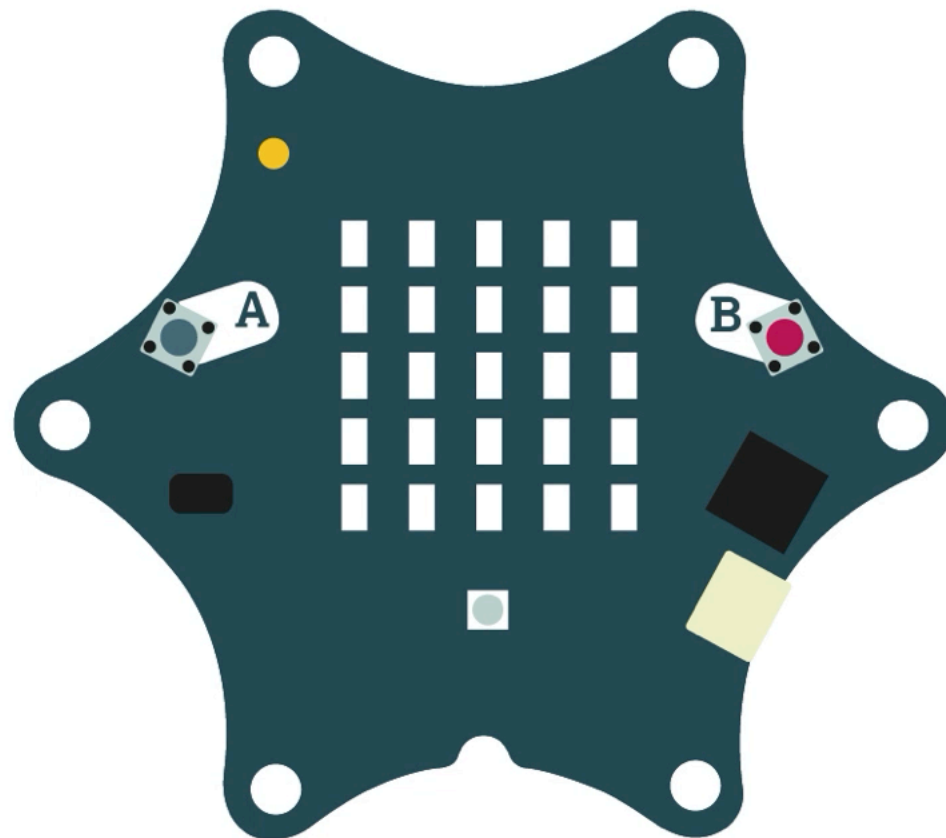
Zufällige Zahl

Antwort

Antwort

Antwort

Antwort

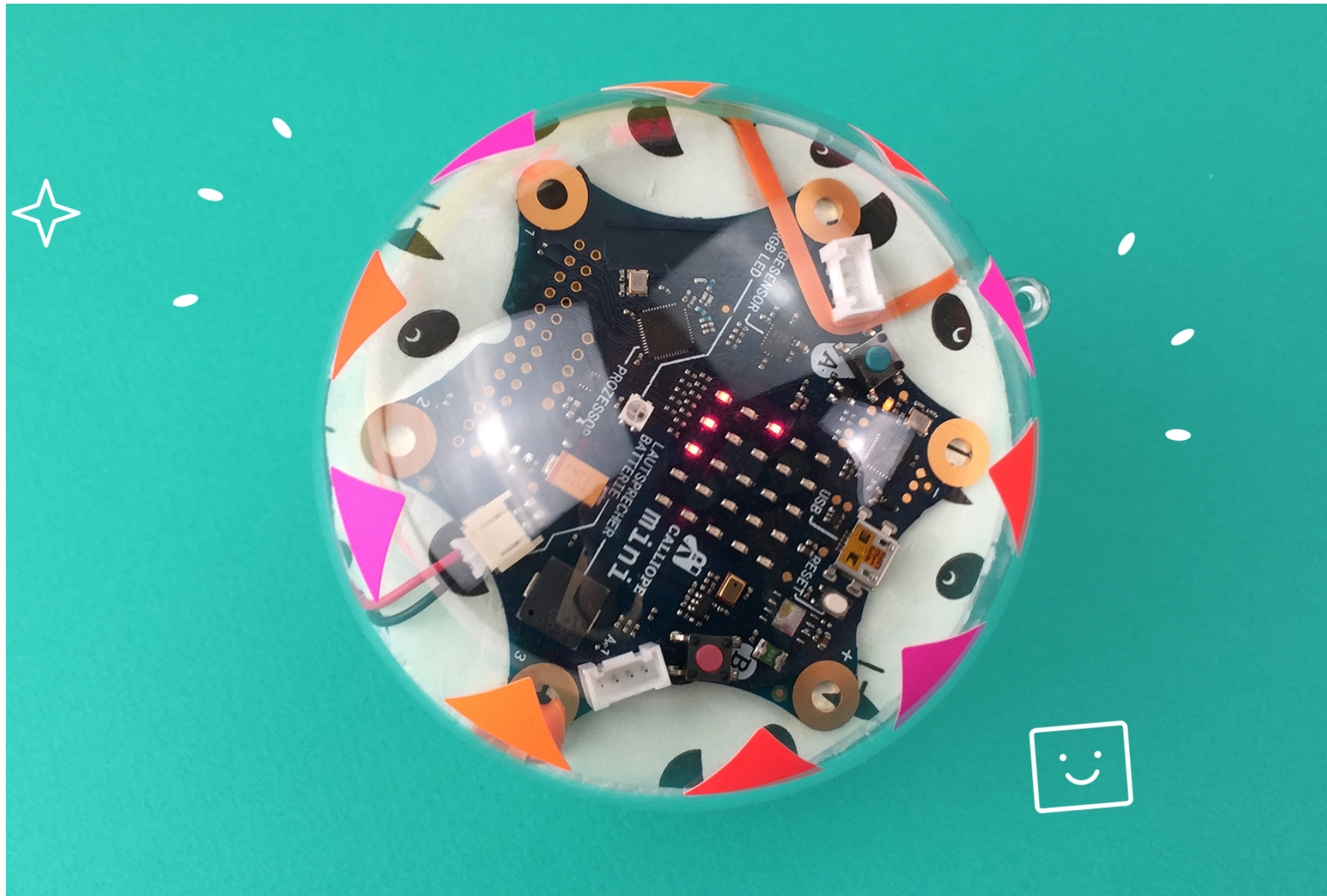


Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Werden alle Optionen angezeigt?
- Sind die Nachrichten eventuell zu lang? Könnten sie sinnvoll gekürzt werden?
- Tipp! Die Textnachrichten sollten zu Ende durchgelaufen sein, bevor der Calliope mini erneut geschüttelt wird.
- Welche Ideen habt ihr noch für passende Farben und Symbole?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



So könnte euer Neujahrs-Orakel aussehen.

SCHWIERIGKEIT



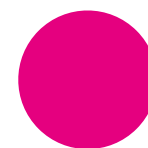
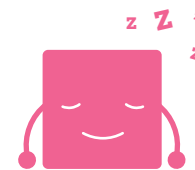
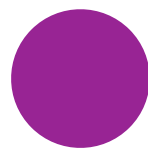
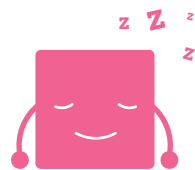
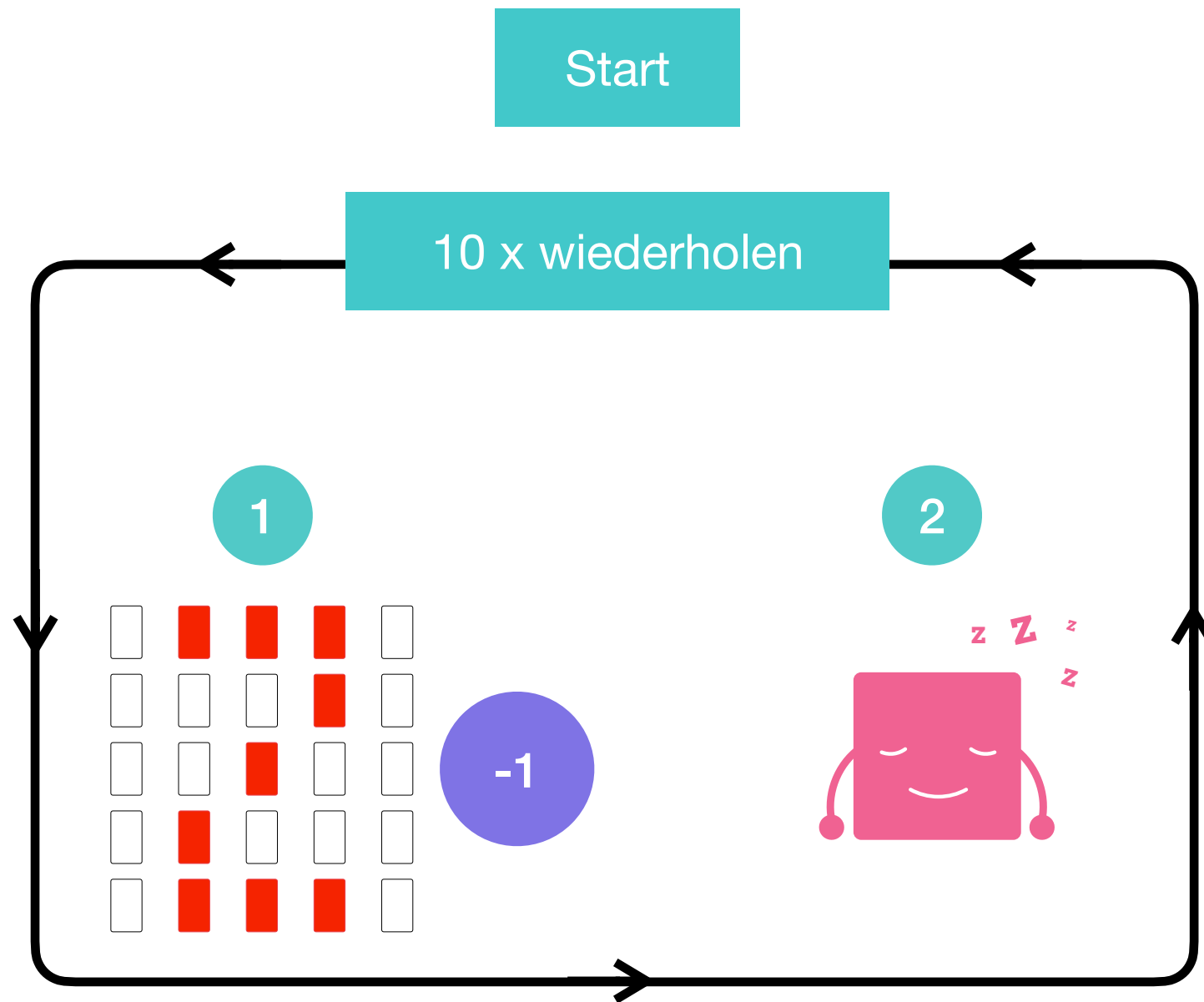
10... 9... 8... DER SILVESTER-TIMER

05



Programmiere deinen eigenen Silvester Countdown mit dem Calliope mini! Entscheide selbst, wie viele Sekunden er rückwärts zählen soll und lasse nach der 0 und einer Pause einen Neujahrsgruß und ein Bild oder bunte Lichter erscheinen.

PROGRAMMABLAUFPLAN



PLAYGROUND

Ausgabe:
RGB
Text

Funktionen:
Start Funktion
Pause
Löschen
Schleife

TIPP:

Damit der Countdown am Ende bei 0 landet, wird in jeder Wiederholungsschleife von 10 immer einer mehr abgezogen und auf der LED-Matrix angezeigt.



PLAYGROUND

```
func start (){  
  rgb.off()  
  for i in 1 ... 10 {  
    display.show(number: 10-UInt16(i))  
    mini.sleep(1000)  
  }  
  rgb.on(color: .green)  
  mini.sleep(100)  
  rgb.on(color: .cyan)  
  mini.sleep(100)  
  rgb.on(color: .magenta)  
  mini.sleep(100)  
  rgb.on(color: .purple)  
  display.show(text: "2021")  
}
```

Schleife

PARTY :)

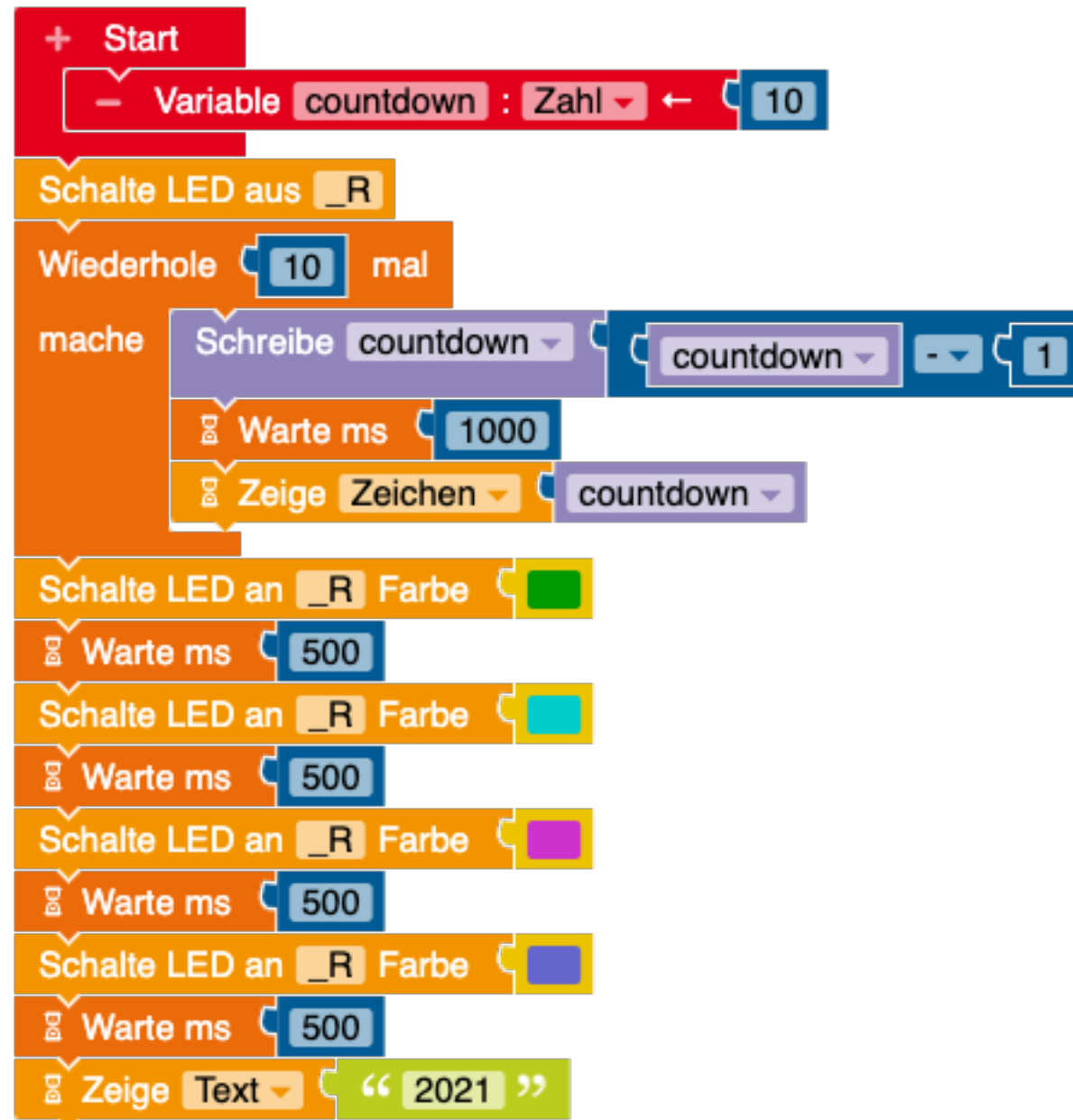
TIPP:

In der Wiederholungsschleife wird bei jedem Durchgang von der Variable „Countdown“ eins abgezogen.

PROGRAMM:

PROGRAMM:

mini-Countdown.xml



```
+ Start
- Variable countdown : Zahl ← 10
Schalte LED aus _R
Wiederhole 10 mal
  mache
    Schreibe countdown - 1
    Warte ms 1000
    Zeige Zeichen countdown
  mache
  Schalte LED an _R Farbe [Green]
  Warte ms 500
  Schalte LED an _R Farbe [Cyan]
  Warte ms 500
  Schalte LED an _R Farbe [Purple]
  Warte ms 500
  Schalte LED an _R Farbe [Blue]
  Warte ms 500
  Zeige Text " 2021 "
```

Schleife

PARTY :)

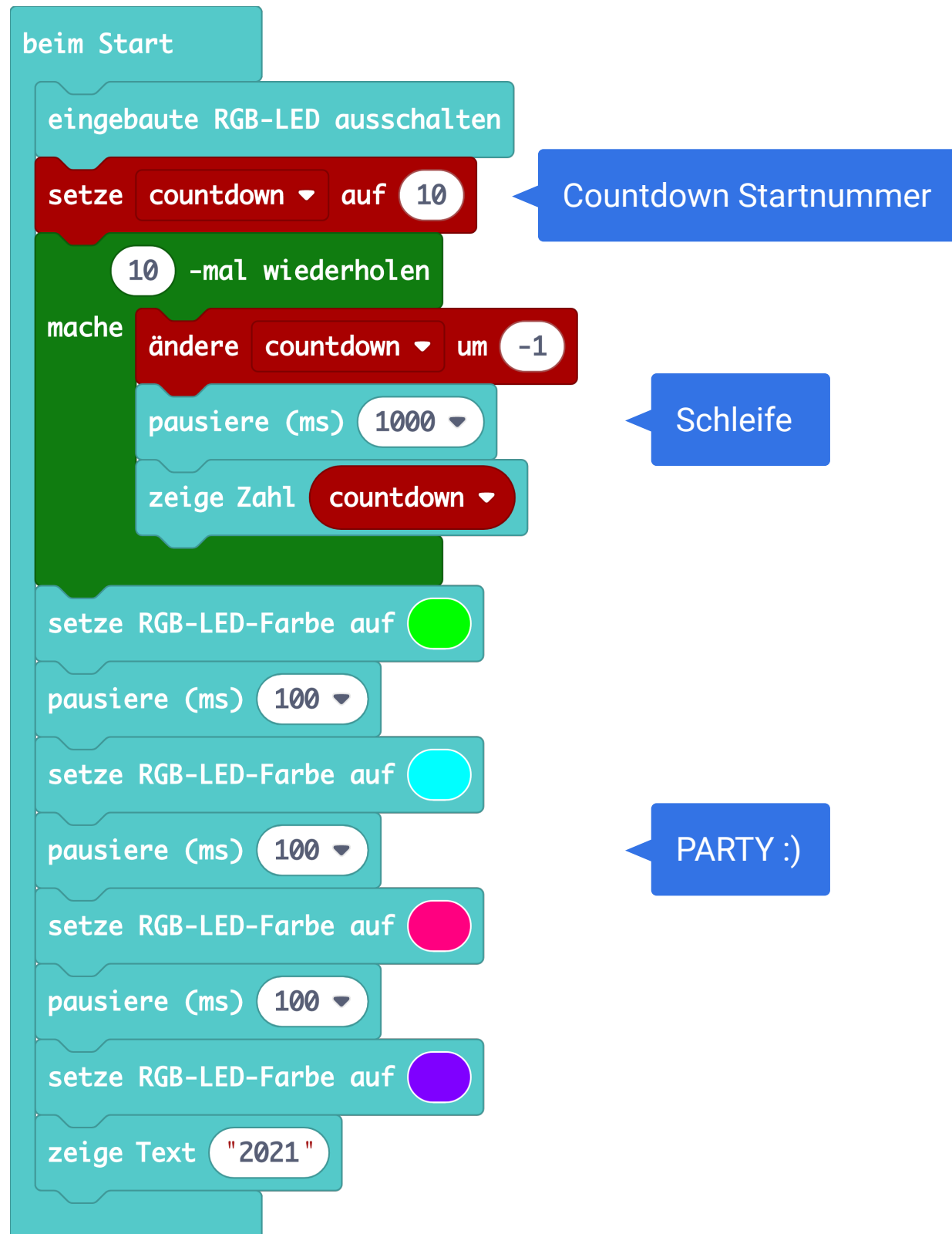


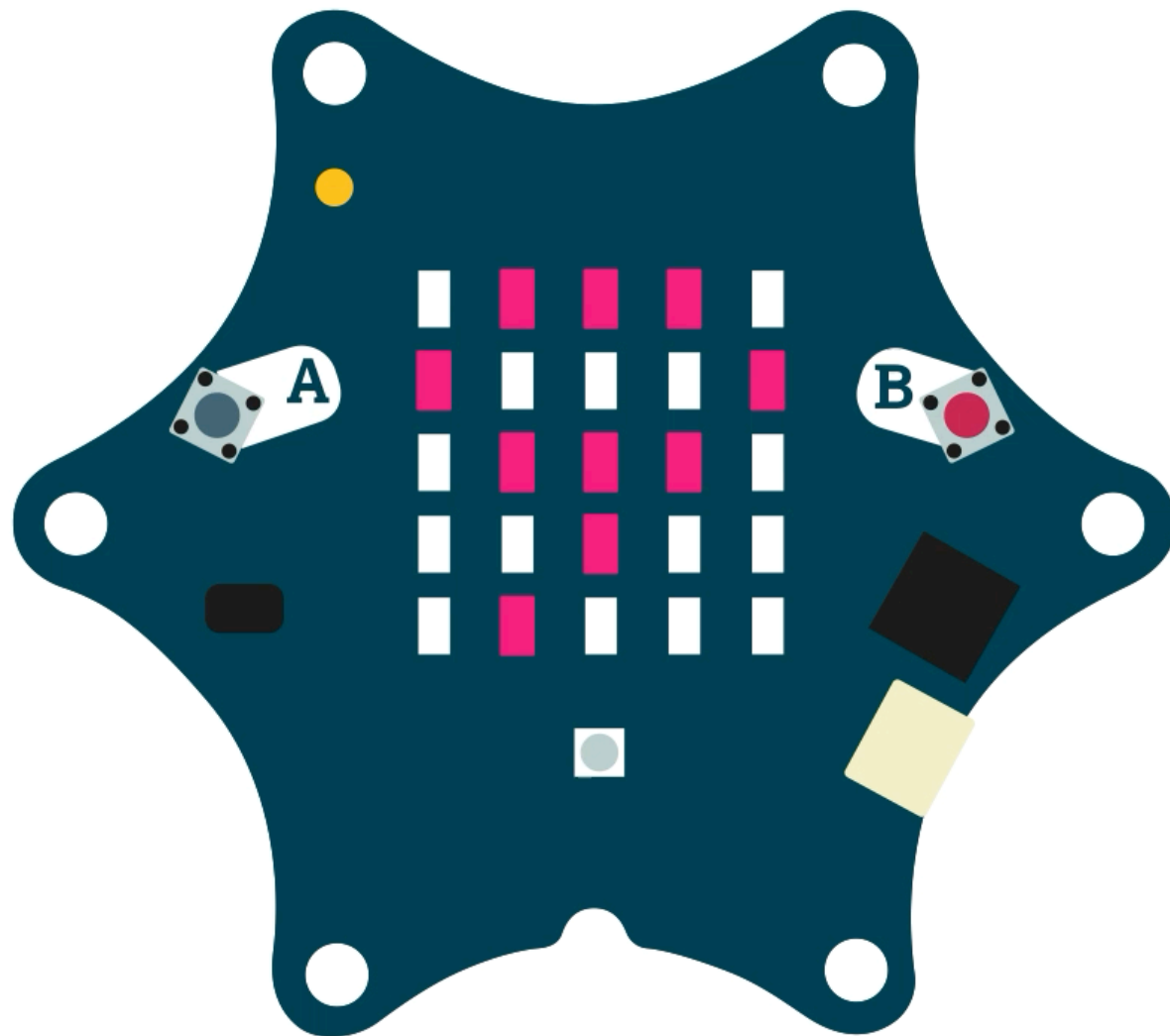
TIPP:

In der Wiederholungsschleife wird bei jedem Durchgang von der Variable „Countdown“ eins abgezogen.

PROGRAMM:

[mini-Countdown.hex](#)





Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Fängt der Countdown bei der gewünschten Zahl an?
- Zählt der Countdown rückwärts bis 0?
- Blinkt die RGB-LED in verschiedenen Farben?
- Wird der Text „2021“ angezeigt?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



CALLIOPE

Calliope gGmbH, Berlin

Verantwortlich:

Prof. Franka Futterlieb

Nadine Klein, Stadtschule Travemünde

Jørn Alraun, Amando Pascotto, Carolin Helmes

Anregungen und Anmerkungen können an hilfe@calliope.cc gesendet werden.