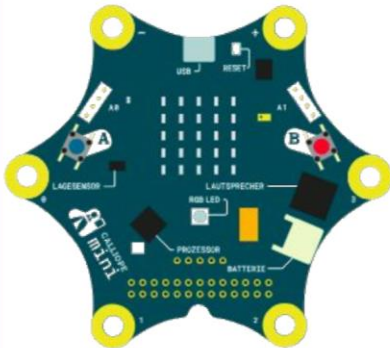




## Startvideo

## 1.1 Bewegung : Stoppuhr

APP CAMPS



## Hinweis

Der eingebaute **Zeitgeber** des Calliope mini wird nicht automatisch angezeigt, wenn du ihn startest. Du kannst dir aber eine sichtbare oder hörbare Kontrolle programmieren, damit du immer genau weißt, ob die Zeit gerade läuft.

## Info

Mit Hilfe von **Variablen** kann sich der Calliope mini Dinge merken. In unserem Fall merkt sich der Calliope mini die Zeit. Die Variable wird benötigt, um die Zeit vor der grafischen Anzeige zu speichern.

## Aufgabe

Wir entwickeln eine **Stoppuhr**.

- Gib zunächst [lab.open-roberta.org](http://lab.open-roberta.org) in deinen Browser ein und wähle dann Calliope mini als System. → Dort kannst du programmieren.

**Schritt 1:** Variable für die gestoppte Zeit anlegen.

Die Variable legst du an, indem du auf das kleine Plus (+) neben **Start** klickst.

- Nenne deine Variable "Zeit"
- Da wir die Zeit noch nicht kennen, tragen wir vorläufig den Wert 0 ein.

**Schritt 2:** Stoppuhr starten

- Suche den passenden Block, um die **Stoppuhr zu starten**. Damit du weißt, dass die Stoppuhr läuft, kannst du die **LED zum Leuchten bringen** und einen kurzen **Signalton** abspielen.

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

**Schritt 3:** Teste deine Anwendung.

Um zu testen, musst du das Programm auf dein Gerät übertragen.



## Nächste Aufgabe

Jetzt können wir die Stoppuhr zumindest schon mal starten. Aber wie beenden wir die Zeitmessung und wie geben wir die Zeit aus?

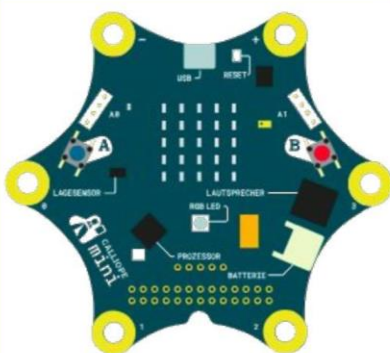
**Probiere es selbst aus, bevor du dir die nächste Lernkarte holst.**

&lt; Calliope Aufbaukurs Bewegung /&gt;

www.appcamps.de

## 1.2 Bewegung: Stoppuhr

APP CAMPS



## Hinweis

Die eingebaute **Stoppuhr** des Calliope mini misst in **Millisekunden**. Deshalb bietet es sich an, den Zeitgeber-Wert vor der Ausgabe **durch Tausend zu teilen**. Einige Blöcke findest du im erweiterten Modus. Klicke dazu auf: ☆2

## Info

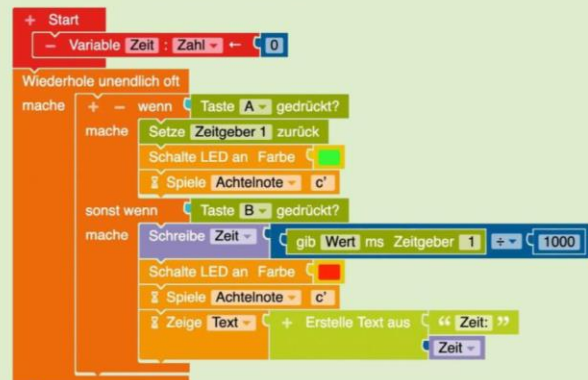
Der Zeitgeber des Calliope mini startet bei jedem Einschalten. Mit dem "Zurücksetzen" Block kannst du einen eigenen Startpunkt festlegen. Mittels einer **Variable** lässt sich ein **spezifischer Wert** zum Zeitpunkt eines Ereignisses (z.B. **Tastendruck**) speichern.

## Aufgabe

Gleich bist du am Ziel: Nun geht es darum, die Stoppuhr **anzuhalten** und den gemessenen Wert auf der LED-Matrix des Calliope mini auszugeben.

- Füge über das + einen weiteren **sonst wenn** Block hinzu.
- Nun hältst du die Stoppuhr mit Taste B an und verwendest deine **Variable**, um **die gestoppte Zeit** zu speichern.
- Das Ergebnis kannst du dir als Text anzeigen lassen.

Hier siehst du eine mögliche Lösung:



## Nächste Aufgabe

Eine Stoppuhr ist schon mal super. Aber wie wäre es mit einem Timer, bei dem die Zeit rückwärts läuft? Hast du eine Idee, wie man das programmieren kann?

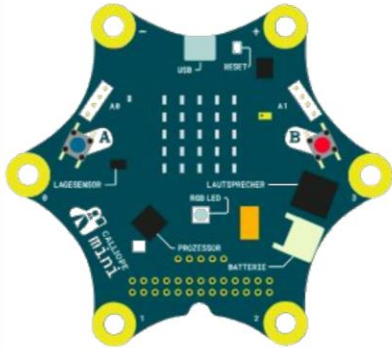
**Probiere es selbst aus, bevor du dir die nächste Lernkarte holst.**

&lt; Calliope Aufbaukurs Bewegung /&gt;

www.appcamps.de



### 1.3 Bewegung: Countdown Timer

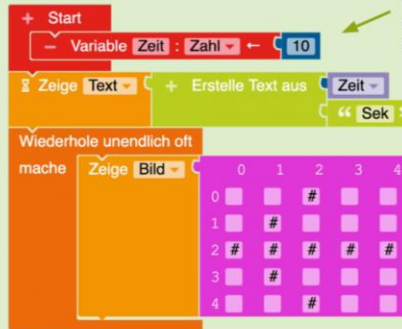


#### Aufgabe

Im Gegensatz zur Stoppuhr zählt die Zeit beim Timer rückwärts. Wähle selbst aus, wie lange der Timer laufen soll.

- Für den Timer benötigst du wieder eine Variable. Du kannst damit die Anfangszeit definieren und deren Veränderung speichern bzw. auslesen.
- Zu Beginn gibst du deiner Variable nun einen Wert.
- Um dem Benutzer die Bedienung zu erleichtern, kannst du **Texte** und **Bilder** nutzen.

Mit diesem Wert kannst du die Laufzeit (Sek.) deines Timers definieren.



Hier kannst du anzeigen, wie lange der Timer läuft.

Der Pfeil nach links soll den Benutzer dazu bringen, Knopf A zu drücken.

#### Info

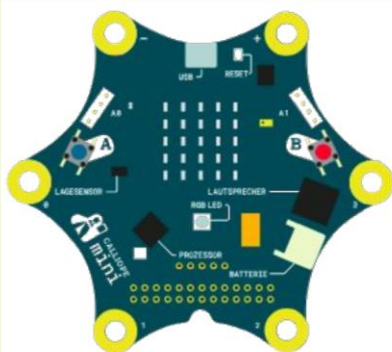
Der eingebaute Zeitgeber vom Calliope mini kann leider nicht rückwärts zählen. Mittels einer Variable kannst du dir aber einfach selbst einen Timer programmieren.

#### Nächste Aufgabe

Als nächstes musst du deine Variable Schritt für Schritt kleiner werden lassen. Hast du eine Idee, wie du die Zahl genau einmal in der Sekunde verringern kannst?

Probieren es selbst aus, bevor du dir die nächste Lernkarte holst.

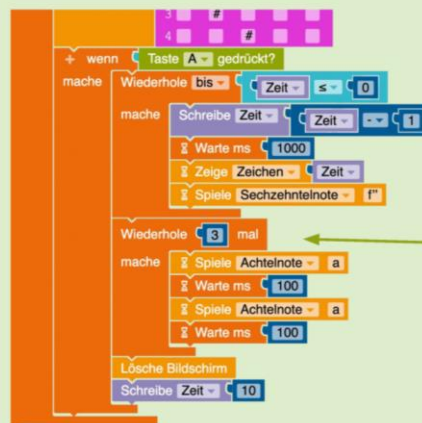
### 1.4 Bewegung: Countdown Timer



#### Aufgabe

Jetzt erweitern wir unser Programm und starten den Timer.

- Wenn die Taste A gedrückt wird, dann soll der Timer loslaufen.
- Du benötigst nun einen **mathematischen Operator**, der deine Variable verringert. Damit das Herunterzählen genau einmal in der Sekunde passiert, verwendest du einen **Warte Block**. Zusätzlich kannst du das Herunterzählen mit einem **Piepstön** verdeutlichen.
- Wenn der Timer abgelaufen ist, soll ein **Tonsignal** erfolgen.



Hier programmierst du die Tonfolge, die bei abgelaufenem Timer ertönt.

#### Info

Falls du mal einen Block nicht findest, dann schau im erweiterten Bereich nach. ☆2

#### Hinweis

Mit dem **Wiederhole bis Block** kannst du Befehle an Bedingungen knüpfen. Achte darauf, dass du die Blöcke für den Timer **innerhalb** der **Wiederhole unendlich oft** Schleife platzierst.

#### Nächste Aufgabe

Bist du fertig? Dann überlege dir, wie du deinen Timer noch erweitern kannst. Falls du keine eigenen Ideen hast, schau auf der Bonuskarte nach.



## Bonuskarte

**Schon alles erledigt?**

Du hast heute eine Stoppuhr und einen Timer selbst programmiert. Jetzt kannst du dir überlegen, wie du deine Programme noch erweitern kannst. Hier findest du ein paar Ideen von uns:

**Erweitere deine Stoppuhr.**

- Du könntest zum Beispiel jeweils bei Klick auf B die Zeit in einer Variable speichern und am Ende über die Pins 0, 1, 2 und 3 die Rundenzeiten ausgeben. Dafür musst du natürlich mehrere Variablen anlegen.
- Du könntest deine gemessene Zeit an einen anderen Calliope funken.

**Erweitere deinen Timer**

- Du könntest über die Taste B die Zeit des Timers eingeben. Dann musst du die Laufzeit nicht immer im Code ändern, sondern kannst das direkt auf dem Spielfeld - oder wo du den Timer eben nutzt - machen.

**Hast du noch andere Ideen? Super, dann kannst du das ausprobieren. Viel Spaß dabei!**

