

## Projekt: Sirene und Blaulicht



Gelegentlich hört und seht ihr sie, wenn ihr durch die Straßen lauft: die Sirenen und Blaulichter von Polizei, Feuerwehr und Rettungsdienst. Wenn ihr dieses Projekt bearbeitet habt, könnt ihr euer eigenes Blaulicht und eine Sirene mit dem Calliope mini simulieren. Dazu werdet ihr auch die bedingte Verzweigung kennenlernen.

Bevor ihr loslegt: Denkt daran, dass ihr ein **neues Projekt** anlegt

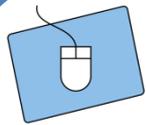
Blaulicht und Sirene



und es unter einem passenden Namen **speichert**.

### Die Sirene

Ihr startet mit der Sirene. Sie soll ertönen, wenn **Knopf A** gedrückt gehalten wird.



Programmiert eine Sirene, indem ihr Schritt für Schritt der Anleitung folgt.

1. Damit die Sirene nur ertönt, wenn ihr **Knopf A** gedrückt haltet, braucht ihr aus Logik



diesen Block:  
Baut ihn in den **dauerhaft**-Block ein.

2. Wenn **Knopf A** gedrückt wird, dann soll die Sirene erklingen. Was muss also in die Lücke eingesetzt werden, in der jetzt noch „**wahr**“ steht?



Wieso kann man denn nicht einfach

wenn Knopf A gedrückt  
nehmen?



Würdet ihr den von Calli vorgeschlagenen Block nehmen, dann würde eure Sirene starten, wenn ihr **Knopf A**

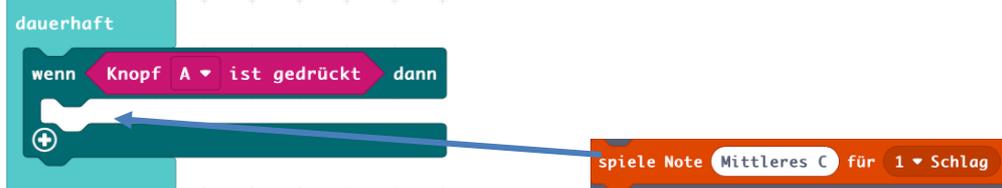


drückt, dann aber auch angeschaltet bleiben. Mit dem **dauerhaft**-Block erreicht ihr, dass eure Sirene **nur** erklingt, solange **Knopf A gedrückt bleibt**. Wenn ihr **Knopf A** also wieder loslasst, ist der Calliope mini wieder stumm.

## Projekt: Sirene und Blaulicht



3. Jetzt braucht ihr noch Töne für eure Sirene. Ihr findet sie in der Kategorie **Musik**.



4. Was passiert? Beschreibt das bisherige Programm, indem ihr in dem folgenden Satz die Lücken ergänzt.

Wenn man \_\_\_\_\_ ,  
dann \_\_\_\_\_ .



### Bedingte Verzweigung

Der **Wenn-dann**-Block wird euch in der Programmierung häufiger begegnen.

Wenn **Knopf A** gedrückt wird,  
dann wird ein **Ton** abgespielt.

Oder allgemeiner:

Wenn eine **Bedingung wahr** ist,  
dann wird ein **Befehl ausgeführt**.

Deshalb spricht man in der Informatik auch von einer **bedingten Verzweigung**. Mit dem **Wenn** wird erst einmal abgefragt, ob eine bestimmte Bedingung erfüllt ist (hier: das Drücken von **Knopf A**). Wenn dies der Fall ist, die Bedingung also wahr ist, dann wird der Befehl hinter **dann** ausgeführt. Falls nicht, geht das Programm zum nächsten Befehl über.



Wenn ihr **Knopf A** gedrückt haltet, ertönt ein Ton. Aber das hört sich noch nicht wirklich nach einer Sirene an, oder?

Baut daher noch mehrere **Töne** ein, damit eine richtige Sirene entsteht. Testet euer Programm.

Polizei und Feuerwehr benutzen ein sogenanntes Folgetonhorn. Ihr könnt es in etwa mit den **Tönen A und D** nachbilden. Ihr könnt aber natürlich auch andere Töne für eure Sirene verwenden.

Andere Töne könnt ihr auswählen, indem ihr in dem Block für die Töne **Mittleres C** anklickt und dann einen anderen Ton auf der Klaviertastatur antippt.



Abbildung 1: Klaviertastatur

## Projekt: Sirene und Blaulicht



Im Moment wechseln die Töne noch sehr schnell. Das liegt daran, dass die einzelnen Töne nur je einen Schlag erklingen.

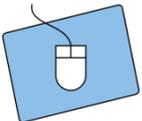
spiele Note Mittleres C für 1 ▾ Schlag

Ändern könnt ihr die Tonlänge, indem ihr die Anzahl der Schläge ändert. Dazu klickt ihr einfach auf den **kleinen Pfeil** neben der 1.

Experimentiert ein wenig mit den Tönen und Längen, bis ihr eine Sirene komponiert habt, die euch gefällt.

## Das Blaulicht

Jetzt fehlt noch das Blaulicht.



Welches Bauteil des Calliope mini könnt ihr nutzen, um ein Blaulicht zu programmieren?

Ihr erinnert euch nicht mehr? Calli weiß Rat.



Auf dem Calliope mini ist eine RGB-LED verbaut. RGB steht für Rot, Grün, Blau. Mit der RGB-LED können alle Farben gemischt werden. Im Simulator ist die RGB-LED unter dem LED-Bildschirm und sieht so aus:



Eure Aufgabe ist nun, die **RGB-LED** und **Knopf B** so zu programmieren, dass sie zusammen als Blaulicht funktionieren. Dies bedeutet konkret:

Wenn ihr **Knopf B** gedrückt haltet, dann soll die **RGB-LED blau** leuchten.

Hierfür benötigt ihr die folgenden Blöcke. Sucht sie heraus, und baut sie richtig in euer schon bestehendes Sirenen-Programm ein. Denkt auch daran, die Farbe anzupassen:

setze RGB-LED-Farbe auf

Knopf B ▾ ist gedrückt

wenn wahr ▾ dann

Testet das gesamte Programm. Da ihr die **Knöpfe A** und **B** leider nicht gleichzeitig drücken könnt, probiert ihr sie am besten einfach nacheinander aus.

## Projekt: Sirene und Blaulicht



Euer Blaulicht – also die RGB-LED – soll jetzt nicht nur **blau** leuchten, sondern **an- und ausgehen**.

- Um das zu erreichen, müsst ihr unter dem Block, der die RGB-LED **blau** leuchten lässt, also noch einen Block ergänzen, der die RGB-LED **ausschaltet**:

eingebaute RGB-LED ausschalten

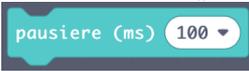
Was fällt euch auf, wenn ihr nun **Knopf B** gedrückt haltet?

Wahrscheinlich ist euch aufgefallen, dass die RGB-LED jetzt gar nicht mehr leuchtet. Das liegt daran, dass die beiden Aktionen

setze RGB-LED-Farbe auf **blau**  
und **eingebaute RGB-LED ausschalten**

ganz schnell hintereinander ausgeführt werden – so schnell, dass das menschliche Auge nicht folgen kann.

Damit eure Augen den Unterschied zwischen „LED blau“ und „LED aus“ wahrnehmen können, müsst ihr den Wechsel zwischen „blau“ und „aus“ etwas verlangsamen. Das macht ihr, indem ihr

Pausen einbaut: 

- Ergänzt in eurem Programm zwei -Blöcke, um den Wechsel zwischen „blau“ und „aus“ sehen zu können. Experimentiert auch ein bisschen mit der Länge der Pausen.

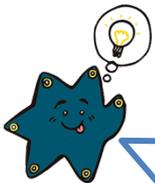
Keine Ahnung, wo die Pausen-Blöcke hingehören? Fragt Calli, den ihr auf der nächsten Seite findet.



### Pausen

Das **ms** im **Pausen**-Block steht für **Millisekunden**. 1000 Millisekunden sind eine Sekunde.

## Projekt: Sirene und Blaulicht



```

wenn Knopf B ist gedrückt dann
  setze RGB-LED-Farbe auf
  pause 100 (ms)
  eingebaute RGB-LED ausschalten
  pause 100 (ms)
  
```

So könnte euer Programmteil für das Blaulicht mit Pausen aussehen:



Testet jetzt noch einmal nacheinander Sirene und Blaulicht.

*Klappt alles? Sehr gut, denn dann habt ihr euer Blaulicht-und-Sirene-Projekt erfolgreich bewältigt und könnt mit einem neuen Projekt beginnen. Sollte es noch Fragen oder Probleme geben, kontaktiert jederzeit einen Betreuer oder eine Betreuerin.*



### Quellenverzeichnis:

**Abbildung 1** – Screenshot des MakeCode-Editors (<https://makecode.calliope.cc/>) ergänzt durch Noten

**Programmierblöcke** – Quelle: Screenshot des MakeCode-Editors (<https://makecode.calliope.cc/>)

**Abbildung der RGB-LED** – Quelle: Screenshot des Simulators des MakeCode-Editors (<https://makecode.calliope.cc/>)

angefertigt vom InfoSphere-Team