

Kleine Programmieraufträge

Aufgabe 1 – Wie heißt du?

Auf dem Calliope mini befindet sich eine LED-Matrix. Das sind die kleinen roten Lämpchen. Diese kann man ganz unterschiedlich programmieren. Wie wäre es mit einem Namensschild? Wählt dazu folgende Bausteine aus und verbinde sie miteinander:



Aufgabe 2 – Hast du Töne?

Der Calliope hat einen Lautsprecher, daher kann er auch Töne abspielen. Versucht einmal das folgende Programm:

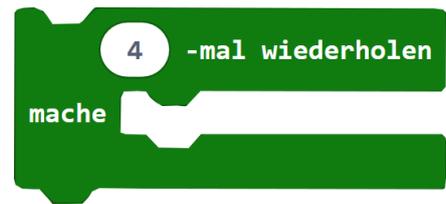


Wenn man nun Knopf A drückt, müsste euer Calliope drei verschiedene Töne abspielen.

Wie ihr seht, kann man auch mehrere Bausteine hintereinander setzen. Versucht nun eine eigene Komposition, wenn man *Knopf B* drückt.

Aufgabe 3 – Schleifen

Immer wenn man möchte, dass der Computer etwas mehrmals hintereinander macht, braucht man Programmschleifen. Versucht es einmal mit eurem Programm aus Aufgabe 2. Die drei verschiedenen Noten sollen viermal hintereinander abgespielt werden.



Wie könnte das Programm aussehen?

Aufgabe 4 – Schüttel mich!

Statt der Knöpfe A und B kann man als Eingabe auch den Befehl „wenn geschüttelt“ verwenden. Nun wollen wir programmieren, dass der Calliope mini die Temperatur anzeigt, wenn er geschüttelt wird. Wählt dazu folgende Bausteine aus und setzt sie zusammen:



Temperatur (°C)



Wie sieht die fertige Programm aus? Testet sie mithilfe des Simulators.

Aufgabe 5 – Würfel

Nun versucht doch einmal, mit Hilfe des Calliope einen Würfel zu programmieren. Die wesentlichen Bausteine dafür kennt ihr schon. Ihr braucht aber auch noch diesen Baustein:

wähle eine zufällige Zahl von 0 bis 10

Funktioniert der Würfel?

Aufgabe 6 – eigene Idee umsetzen

Nun setzt eure eigenen Ideen um. Probiert dabei auch Bausteine aus, die ihr bisher noch nicht verwendet habt. Wenn ihr interessante Dinge dabei herausfindet, teilt eure Erkenntnisse mit euren Mitschülerinnen und Mitschülern.

Programmierauftrag: Tresor mit Alarmanlage

(Fortgeschritten)

Nutzt den Calliope mini, um einen Prototypen für einen Tresor mit Alarmanlage zu erstellen. Programmiert mit Hilfe der Entwicklungsumgebung MakeCode: <https://makecode.calliope.cc/>

Aufgaben:

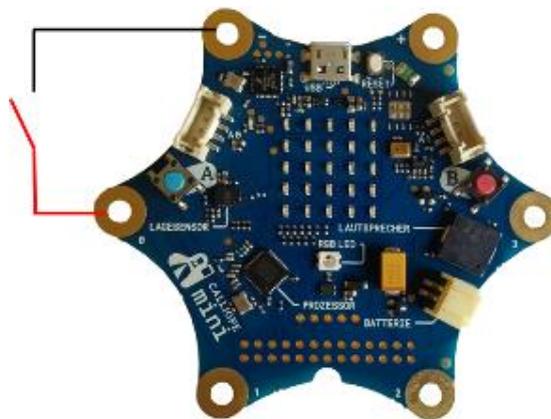
1. Wie funktioniert ein Tresor mit Alarmanlage?
 - a. Wie soll sich euer Tresor verhalten, wenn
 - i. dieser unauthorisiert geöffnet wird?
 - ii. dieser von euch geöffnet wird?
 - b. Wie könnte eurer Tresor-Programm aufgebaut sein? (z.B. wenn x passiert, dann soll y)
2. Ihr habt nun schon einige Erfahrung mit dem Calliope. Welche Programmierbausteine könnten dafür in Frage kommen?
3. Programmiert und testet eure Idee.
4. Überlegt euch nun ein interessantes und praktisches Design für euren Tresor mit Alarmanlage. Ihr könnt auch weitere Hilfsmittel verwenden, z.B. Karton, Krokodilklemmen, Alufolie, usw.
5. Stellt euren Tresor mit Alarmanlage am Ende der Klasse vor.

Tipps:



Für dieses Projekt kann der Pin P0 auf der linken Seite der Platine verwendet werden. P0 wird als gedrückt registriert, wenn der Pin mit Masse (-) verbunden ist. Nutze dies für deine Alarmanlage.

Der Calliope mini kann offene und geschlossene Stromkreise erkennen. Um einen Stromkreis zu schließen, muss beispielsweise Pin 0 mit Masse (-) verbunden werden. Den Kontakt stellt man am besten mit Krokodilklemmen her.



Programmierauftrag: Alarmanlage

Nutzt den Calliope mini, um einen Prototypen für eine Alarmanlage zu erstellen. Programmiert mit Hilfe der Entwicklungsumgebung MakeCode <https://makecode.calliope.cc/>

Aufgaben:

1. Wie funktioniert eine Alarmanlage?
 - a. Wie soll eure Alarmanlage funktionieren? Wie aktiviert man den Alarm? Wie deaktiviert man den Alarm?
 - b. Was soll beispielsweise passieren, wenn das Zelt geöffnet wird oder eine Tasche genommen wird?
 - c. Wie könnte eurer Alarmanlagen-Programm aufgebaut sein? (z.B. wenn x passiert, dann soll y)
2. Ihr habt nun schon einige Erfahrung mit dem Calliope. Welche Programmierbausteine könnten dafür in Frage kommen?
3. Programmiert und testet eure Idee.
4. Überlegt euch nun ein interessantes und praktisches Design für eure Alarmanlage.
5. Stellt eure Alarmanlage am Ende der Klasse vor.

Tipp:

