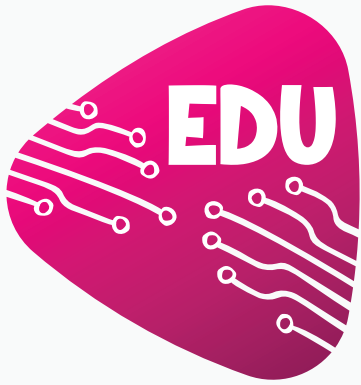


Mit EDU schreibst du deine ersten Computerbefehle ganz spielerisch!



Ist es im Klassenzimmer angenehm oder eher zu warm, zu kalt?
Mit EDU kannst du das überprüfen und die Temperatur ganz genau messen und anzeigen lassen.

Möchtest du eine lustige kleine Melodie zusammenstellen?
Mit EDU lassen sich ganz individuelle Tonfolgen programmieren.

Wie das alles geht?
Das siehst du gleich.

Mit dem EDU-Board und dem dazugehörigen EDU-Editor und seinen vielen, bunten EDU-Programmierbausteinen lassen sich endlos viele Programme schreiben!

Nach diesen ersten paar Übungen wird dir das Programmieren gar nicht mehr geheimnisvoll vorkommen!

ACHTUNG, FERTIG, LOS!

Wir bauen zusammen ein erstes kleines Programm. Sicherlich kennst du das „Achtung, fertig, los“ aus dem Sport. Das wollen wir auf dem Computer programmieren.

Suche im Block **Sensoren** den Programmbaustein „wenn Taste OK geklickt mache“ und füge ihn unter Loop ein.



Wenn du nun die OK-Taste auf deinem EDU-Board drückst, wird der Befehl ausgeführt, den du als nächstes einfügst.

Suche im Block **Bildschirm** „Text auf Bildschirm ausgeben“ und platziere ihn wie folgt.



Füge als Text in Zeile 1 das Wort „ACHTUNG“ ein.

Suche im Block **Zeit** den Programmbaustein „warte 1 Sekunde“.



Und schon hast du dein erstes Programm geschrieben! Lade es nun mit einem Klick „Hochladen“ auf dein EDU-Board und beobachte, was passiert.

Ganz am Ziel sind wir natürlich noch nicht. Nach der Wartezeit von einer Sekunde wollen wir auf dem kleinen Bildschirm des EDU-Boards das Wort „FERTIG“ darstellen lassen. Wo du diesen Programmbaustein findest, weißt du ja bereits.

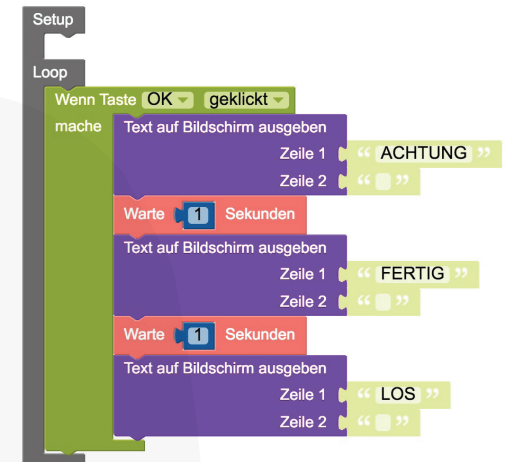


Was folgt als nächstes? Richtig, wir warten wieder eine Sekunde.



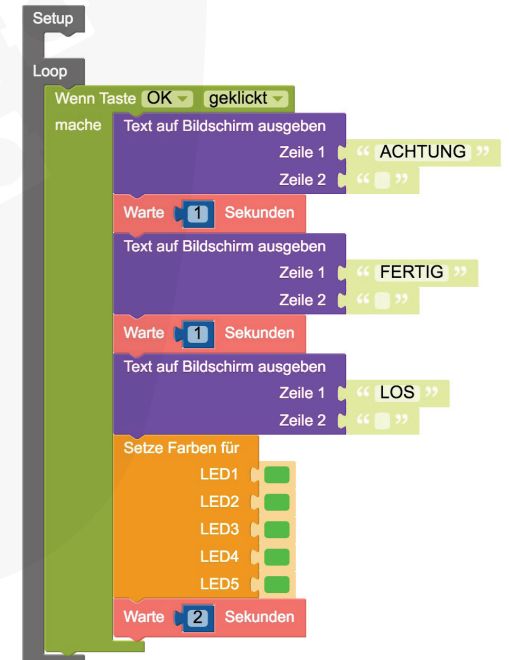
Nun soll auf dem Bildschirm das Wort „Los“ dargestellt werden. Hierzu fügst du wieder den Programmbaustein „Text auf Bildschirm ausgeben“ im Block **Bildschirm** ein.

Nun sollte dein Programm so aussehen:



Damit etwas Farbe ins Spiel kommt, lassen wir bei „LOS“ die LEDs grün leuchten. Sicher findest du diesen Programmbaustein ganz allein.

Dann warten wir erneut 2 Sekunden.



ACHTUNG, FERTIG, LOS!

Nachdem 2 Sekunden lang das Wort "LOS" auf dem Bildschirm angezeigt wurde und die LEDs grün leuchteten, soll der Bildschirm gelöscht und die LEDs wieder ausgeschaltet werden. Dafür wird die Farbe schwarz verwendet.



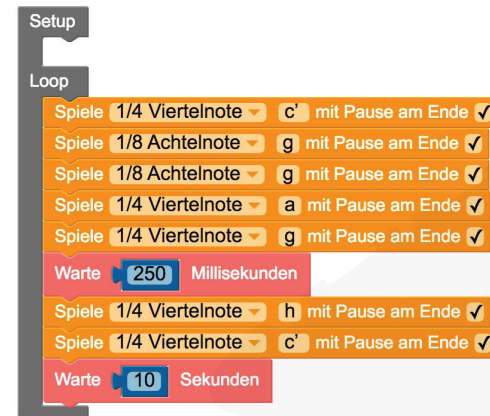
Dein Programm ist nun fertig und du hast das EDU-Board schon voll im Griff!

Wenn du möchtest, kannst du natürlich deine eigenen Anpassungen daran vornehmen oder gleich zum nächsten Übungsprogramm springen.

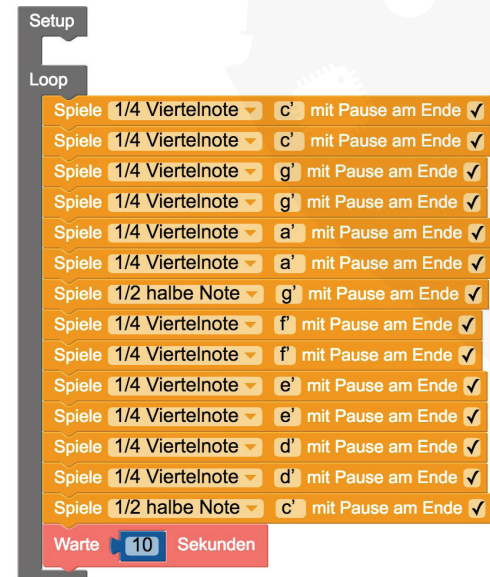
TON / MELODIE ABSPIELN

Mit dem EDU-Board kannst du ganz einfach Töne und Melodien abspielen lassen. In den Blöcken **♫ Ton** und **🕒 Zeit** findest du die passenden Bausteine dafür.

Beispiel 1



Beispiel 2

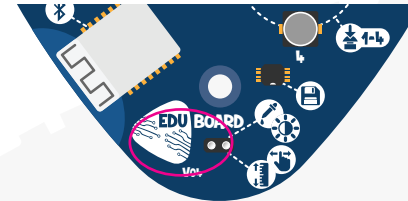


Versuche nun, eigene Melodien zu bauen.

GESTENERKENNUNG

Das EDU-Board kann auch Gesten erkennen. Dazu „wischst“ du mit deiner Hand über den Sensor ganz unten auf der Platine. Hoch, runter, links und rechts.

Bevor das aber funktioniert, musst du einige



Programmschritte vorbereiten.

Füge die Bausteine aus den Blöcken **🔌 Sensoren** und **🖥 Bildschirm** wie unten abgebildet selbständig zusammen.



Beobachte nun, was passiert, wenn du mit deiner Hand über den Sensor wischst.

TEMPERATUR

Und nun zur Raumtemperatur!

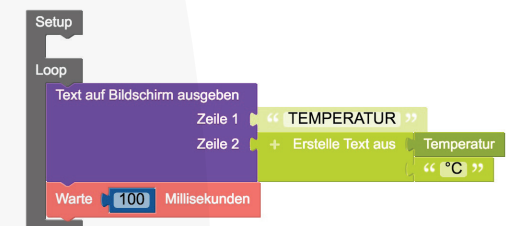
Du bist ja nun ein fortgeschrittener Programmierer und weißt sicher, wie du das entsprechende Programm für dieses Beispiel schreibst. Du findest alle benötigten Bausteine in den Blöcken:

🖥 Bildschirm

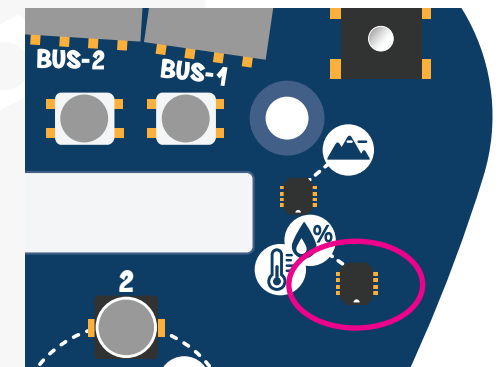
🔌 Sensoren

📄 Text

🕒 Zeit



Damit wir auch genügend Zeit haben, den aktuell gemessenen Temperaturwert vom kleinen Bildschirm abzulesen, definieren wir eine kurze Pause von 100 Millisekunden (1/10 Sekunde), bevor der neue Wert auf dem Bildschirm angezeigt wird.



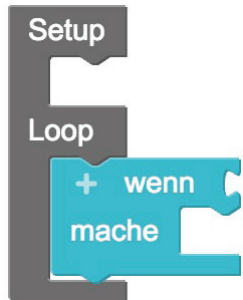
Tipp: Um die Temperatur künstlich zu erhöhen, legst du einfach einen Finger auf den Sensor. Die gemessene Temperatur steigt dann ganz langsam an.

ALARMANLAGE

Hast du dein EDU-Board inzwischen so lieb gewonnen, dass du es mit niemandem teilen magst? Dann bist du sicher froh, wenn es einen Alarm auslöst, sobald man es dir wegnehmen will.

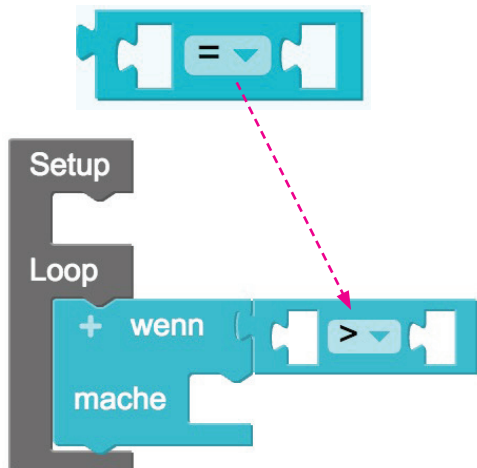
Das lässt sich auch gut programmieren. Allerdings gibt es dazu einen neuen Befehl, den du zuerst eingeben musst:

Ziehe aus dem Block **Logik** den folgenden Baustein in die Klammer Loop.



Füge folgenden Baustein (auch aus dem Block **Logik**) hinten an.

Vergiss nicht, das Symbol = in das Symbol > zu ändern.

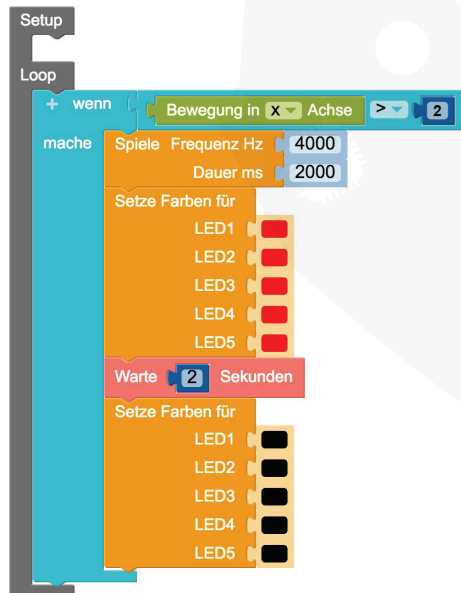


Befülle nun die leeren Felder darin mit "Bewegung in X Achse" aus dem Block **Sensoren** und „eine Zahl“ aus dem Block **Rechnen**.



Das bedeutet: Sobald der Beschleunigungssensor (Bewegung in X-Achse) eine Bewegung misst (Wert grösser als 2), wird der nächste von dir programmierte Befehl ausgeführt.

Du programmierst diesen nun wie folgt: Spiele einen Ton ab, der 2000 Millisekunden (2 Sekunden) andauert und setze die LED-Farbe auf Rot. Warte dann 2 Sekunden und schalte die LEDs wieder aus.



Bravo!
Deine Alarmanlage ist fertig programmiert.
Lade das Programm auf dein EDU-Board.

LICHTSCHALTER

Als letztes Programmierbeispiel bauen wir einen ganz einfachen Lichtschalter.

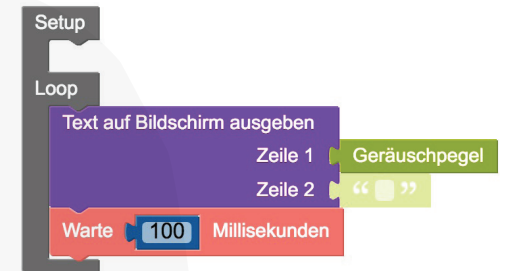
Wird der Knopf links gedrückt, geht das Licht an. Wird der Knopf rechts gedrückt, geht das Licht aus.

In den Blöcken **Sensoren** und **LED** findest du die passenden Bausteine dafür.



Um das Ganze etwas cooler zu gestalten, kannst du auch das Ein- und Ausschalten mit einem Händeklatschen steuern.

Dazu brauchen wir allerdings einen Zwischenschritt. Das EDU-Board muss wissen, wie laut es ohne Händeklatschen ist. Das können wir mit diesem Programm überprüfen.



Wir laden das Programm auf das EDU-Board und lesen den Wert vom Bildschirm ab.

Hinweis: Der Wert einer lärmfreien Umgebung liegt im Normalfall bei etwa 500. Überprüfe dein Messresultat.

Das EDU-Board soll nun alle Werte über 800 für ein Händeklatschen halten und dementsprechend das Licht ein- und ausschalten.

In den Blöcken **Logik**, **Sensoren**, **Rechnen**, **LED** und **Zeit** findest du die passenden Bausteine.

Aufgabe: Versuche herauszufinden, wozu die Wartezeit von 200 ms benötigt wird.



Alle Programmschritte erfolgreich erstellt?
Noch spannendere Aufgaben folgen bald!