

Mathematik, Informatik, Naturwissenschaften, Technik (MINT)

Mit dem Calliope mini Informatik entdecken – Einführung in das algorithmische Problemlösen

Nahezu alle Lebensbereiche werden in der heutigen Welt von automatisierten Prozessen, insbesondere zur Datenverarbeitung, durchdrungen. In der Regel sollen konkrete Anwendungsprobleme gelöst werden. Beim autonomen Fahren beispielsweise werden Daten wie Abstände zu Hindernissen über Sensoren erfasst und von einer bestimmten Software ausgewertet. Die Ausgaben der Software beeinflussen die Aktionen bestimmter Aktoren, z. B. die der LEDs oder Motoren. Somit sind Informatikkompetenzen für ein selbstbestimmtes und verantwortungsvolles Leben von großer Bedeutung.

In der vorliegenden Handreichung werden zwei Doppelstunden vorgestellt, die Schüler:innen der Jahrgangsstufen drei bis sieben über blockbasiertes Programmieren kreativ Zugänge zur Informatik eröffnen, aber auch Herausforderungen der Unterrichtsintegration solcher Ansätze identifizieren. Der Einstieg in das algorithmische Problemlösen erfolgt spielerisch über Zeichenaufgaben sowie die Bewältigung eigener Projekte in Partner- oder Einzelarbeit. Eingeführt wird in das für die Informatik bedeutsame EVA-Prinzip (Eingabe, Verarbeitung, Ausgabe). In der zweiten Doppelstunde vertiefen die Lernenden ihr Verständnis durch den Einsatz eines Ultraschallsensors unter Herstellung von Bezügen zur realen Welt. Differenzierende Aufgaben ermöglichen frühzeitig Erfolgserlebnisse. Das Programmieren mit dem Calliope mini weckte Begeisterung bei allen Kindern und zeigte insbesondere Vorteile für sprachunabhängiges Lernen.

Autor:innen

Prof. Dr. Claudia Hildebrandt, Heidelberg School of Education, PH Heidelberg | Matthias Matzner, Heidelberg School of Education, Pädagogische Hochschule Heidelberg | Philipp Jung, Heidelberg School of Education, Pädagogische Hochschule Heidelberg | Jonas Braun, Heidelberg School of Education | Pädagogische Hochschule Heidelberg | Larissa Nebelung, Universität Heidelberg | Andreas Eller, Universität Heidelberg

Produkttyp

Handreichung für Lehrkräfte mit Unterrichtsmaterial

Schulstufe

Primarstufe (Jahrgangsstufe 3/4), Sekundarstufe I (Jahrgangsstufe 5/6/7)



Dieses Produkt ist unter der Lizenz CC BY-NC-SA 4.0 veröffentlicht. Ausgenommene Inhalte sind an den einzelnen Inhalten angegeben. Die Urheber:innen sollen bei der Weiterverwendung wie folgt angegeben werden: Hildebrandt, C., Matzner, M., Jung, P., Braun, J., Nebelung, L., Eller, A., Kompetenzverbund lernendigital, entstanden im Projektverbund MINT-ProNeD.



Mit dem Calliope mini Informatik entdecken – Was erwartet die Lernenden und die Lehrkräfte?

„Am allerliebsten hätte ich auch noch den Feuchtigkeitssensor ausprobiert!“
(Rückmeldung einer Viertklässlerin nach der 2. Doppelstunde, 2024)

In der Aussage der Viertklässlerin wird die in ihr geweckte Begeisterung für das Programmieren und die Arbeit mit Sensoren deutlich. Blockbasiertes Programmieren ermöglicht bereits Grundschulkindern einen niedrigschwelligen Zugang zur Informatik. Es lassen sich rasch Erfolgserlebnisse gepaart mit viel Freude und Entdeckungsdrang beobachten.

In dieser Handreichung wird eine Unterrichtseinheit, bestehend aus zwei Doppelstunden, zum Thema „Mit dem Calliope mini Informatik entdecken“ vorgestellt.

Erste Doppelstunde: Einführung – Mit dem Calliope mini das Programmieren entdecken

Während der ersten Doppelstunde werden die Schüler:innen spielerisch an das Programmieren herangeführt und entdecken explorativ das blockbasierte Programmieren. Zur Systematisierung lernen die Grundschulkinder die Unterschiede und Zusammenhänge zwischen **Eingabe, Verarbeitung und Ausgabe** kennen. Dieses sogenannte EVA-Prinzip ist grundlegend für die Informatik.

Zweite Doppelstunde: Mit dem Ultraschallsensor „Abstände hören“

Ziel der zweiten Doppelstunde ist eine **Hinführung zum algorithmischen Problemlösen** und zum Verstehen der Funktionsweise von Sensoren. Über einen Ultraschallsensor erhält der Calliope mini Abstandsdaten zu Gegenständen, welche die LEDs des Calliope mini als Zahlenwerte anzeigen sollen. Später werden beim Unterschreiten von bestimmten Abständen zu „Hindernissen“ zum Ultraschallsensor mit dem eingebauten Piezolautsprecher Tonsignale ausgegeben. Eine Verknüpfung zur Echoortung bei Fledermäusen oder Delfinen dient dabei als reales Beispiel aus der Natur.

Wir möchten Sie herzlich einladen und ermutigen, zusammen mit Ihren Schüler:innen die Informatik mithilfe des Calliope mini zu entdecken!

Die vorgestellte Unterrichtseinheit wurde mit mehreren Klassen der Stufen 3, 4 und 7 an unterschiedlichen Schulen erprobt und auf Basis gesammelter Erfahrungen weiterentwickelt.

Inhaltsverzeichnis

Mit dem Calliope mini Informatik entdecken – Einführung in das algorithmische Problemlösen	1
1. Hintergrund.....	5
1.1 Programmieren in der Grundschule – Warum das denn?.....	5
1.2 Algorithmisches Problemlösen.....	6
1.3 EVA-Prinzip.....	8
1.4 Die Aktoren und Sensoren des Calliope mini.....	8
1.5 Bausteine zum Programmieren.....	9
1.6 Hintergrund - Literaturverzeichnis	15
2. Erste Doppelstunde	16
2.1 Thema.....	16
2.2 Benötigtes Material	16
2.3 Kompetenzen.....	17
2.4 Verlaufsplan der ersten Doppelstunde	18
2.5 Arbeitsblätter und EVA-Schilder.....	23
2.6 Programmierbausteine und EVA-Karten	28
2.7 Mögliche Tafelbilder zum Einstieg	30
2.8 Weitere Beispielbilder.....	32
2.9 Mögliches Tafelbild – Sicherungsphase.....	33
2.10 Weitere Erläuterungen zum spielerischen Einstieg: Algorithmen ohne Computer erleben	34
2.11 Mögliche Stolperstellen in der explorativen Phase	34
3. Zweite Doppelstunde	36
3.1 Thema.....	36
3.2 Benötigtes Material	36
3.3 Kompetenzen.....	37
3.4 Verlaufsplan zur zweiten Doppelstunde.....	38
3.5 Mögliche Tafelbilder	43
3.6 Aufgaben zur zweiten Doppelstunde.....	45
3.7 Lösungen zu den Aufgaben der zweiten Doppelstunde.....	46
3.8 Die Funktionsweise eines Ultraschallsensors – Von der Natur zur Technik.....	47
3.9 Programmieren mit dem Ultraschallsensor – Vom Messen zur akustischen Ausgabe	48

3.10 Mögliche Stolperstellen: „Die angezeigten Zahlen ergeben doch gar keinen Sinn!“	49
3.11 Zusammenführung der Erkenntnisse und Ausblick auf weitere Anwendungen	50

1. Hintergrund

1.1 Programmieren in der Grundschule – Warum das denn?

Nahezu alle Lebensbereiche werden in der heutigen Welt von Informatiksystemen, insbesondere zur Datenverarbeitung, durchdrungen. Somit sind Informatikkompetenzen für ein selbstbestimmtes und verantwortungsvolles Leben von großer Bedeutung. Automatisierte Prozesse dienen in der Regel dazu, konkrete Anwendungsprobleme zu lösen und Fehler zu reduzieren. Beim autonomen Fahren beispielsweise werden Daten wie Abstände zu Hindernissen über Sensoren erfasst und von einer bestimmten Software ausgewertet. Die Ausgaben der Software beeinflussen die Aktionen bestimmter Akteure, z. B. die der Motoren. Diese „spezifische Zusammenstellung von Hardware-, Software- und Netzwerkkomponenten zur Lösung eines Anwendungsproblems“ (Best et al., 2019, S. 21) wird **Informatiksystem** genannt. Beispiele für Informatiksysteme sind sich autonom bewegende Systeme, Bankautomaten, aber auch Laptops und Smartphones.

Die Studie von Baum et al. (2019) zeigt, dass Viertklässlern, obwohl sie oft nur über begrenzte Vorkenntnisse zu Informatiksystemen verfügen, in der Lage sind, **altersangemessene Systeme** wie den Calliope mini **zielgerichtet zu programmieren** (Baum et al., 2019).

Ein weiterer Vorteil des Calliope mini liegt in der möglichen **Verwendung visueller Programmiersprachen**. Syntaxfehler werden vermieden und damit die Chance gegeben, einen niedrigschwelligen Einstieg in die Welt des Programmierens zu ermöglichen. Modrow, Mönig und Strecker (2011) betonen, dass visuelle Programmiersysteme wie Scratch einen spielerischen Zugang zur Algorithmik ermöglichen und das selbstständige Arbeiten fördern. Kinder und Jugendliche, die mit Scratch arbeiten, entwickeln oft eigene Problemstellungen, debuggen ihre Programme kontinuierlich und erweitern ihre Projekte eigenständig – ein Verhalten, das die kreative und iterative Natur der Informatik hervorhebt. Diese Erfahrungen zeigen, dass visuelle Programmierumgebungen ein wichtiger Baustein für das Ziel „Informatik für alle“ sind (Modrow et al., 2011).

Der Calliope mini bietet zudem eine ideale Grundlage für den **Übergang in die Sekundarstufe I**. Mit dem Calliope mini lassen sich zentrale inhalts- und prozessbezogene Kompetenzen aus dem Wahlfach Informatik der Sekundarstufe I vorbereiten, so dass im Sinne des didaktischen Prinzips des Spiralcurriculums ein langfristiger Lernprozess ermöglicht wird. Entsprechend den Vorgaben des Bildungsplans Aufbaukurs Informatik Klasse 7 (unter „Modellieren und Implementieren“, soll ausgehend von spielerisch-probierenden Ansätzen eine schrittweise Entwicklung hin zu planvollem und strukturiertem Arbeiten erfolgen (Ministerium für Kultus, Jugend und Sport Baden-Württemberg, 2016b, 2.2).

Darüber hinaus unterstützt das Arbeiten mit dem Calliope mini die Entwicklung von Kompetenzen aus dem Bereich „Kommunizieren und sich verständigen“ (Ministerium für Kultus, Jugend und Sport Baden-Württemberg, 2016a, 2.3) sowie „Kommunizieren und Kooperieren“ (Ministerium für Kultus, Jugend und Sport Baden-Württemberg, 2016b, 2.3). Schüler:innen können ihre Ideen, Beobachtungen und Lösungswege dokumentieren und diese unter Verwendung fachlicher Begriffe diskutieren. Dabei greifen sie auf vorhandene Medien und Kommunikations- und Kooperationsinfrastrukturen zurück und bringen in Diskussionen Toleranz, respektvollen Umgang sowie Offenheit gegenüber verschiedenen

Lösungswegen, Ansichten und Meinungen ihrer Mitschüler:innen zum Ausdruck (Ministerium für Kultus, Jugend und Sport Baden-Württemberg, 2016, 2.3).

Auch die prozessbezogene Kompetenz „Analysieren und Bewerten“ wird gefördert: Schüler:innen untersuchen anhand ihrer Projekte Programme und analysieren deren Funktionalität, indem sie Kontrollstrukturen identifizieren und schrittweise den Programmablauf nachvollziehen. Dies dient dem tieferen Verständnis von Informatiksystemen und befähigt sie, Risiken und Chancen einzuschätzen und zu bewerten (Ministerium für Kultus, Jugend und Sport Baden-Württemberg, 2016b, 2.4).

Ein großer Vorteil des Calliope mini ist seine langfristige Nutzbarkeit. Mit der visuellen und Scratch-ähnlichen Programmieroberfläche MakeCode oder Open Roberta bleibt der Mikrocontroller auch in der Sekundarstufe I ein wertvolles Werkzeug. Das zeigt sich auch in der Praxis. Wie Modrow, Mönig und Strecker (2011) hervorheben, wird die Programmieroberfläche Scratch, die ursprünglich für Grund- und Vorschulen konzipiert wurde, erfolgreich bis in die Sekundarstufe I und darüber hinaus genutzt. Der Calliope mini kann damit Schüler:innen über einen längeren Zeitraum begleiten und so den Grundstein für eine fundierte Informatikbildung legen.

Durch seine kindgerechte Gestaltung, seine Vielseitigkeit und die Anpassungsmöglichkeiten an die Anforderungen der Sekundarstufe I bietet der Calliope mini somit eine ideale Möglichkeit, Informatik frühzeitig und praxisnah in den Schulalltag zu integrieren. Lehrkräfte haben damit ein wirkungsvolles Werkzeug zur Hand, um die Neugier der Schüler:innen zu wecken, grundlegende Kompetenzen zu vermitteln und sie auf eine digitalisierte Zukunft vorzubereiten.

Weiterführende Links zu den Bildungsplänen der Grundschule:

2.3 Kommunizieren und sich verständigen

<https://www.bildungsplaene-bw.de/,Lde/LS/BP2016BW/ALLG/GS/SU/PK/03>

Weiterführende Links zu den Bildungsplänen der Sekundarschulen:

<https://www.bildungsplaene-bw.de/,Lde/LS/BP2016BW/ALLG/SEK1/INF7>

2.2. Modellieren und Implementieren

<https://www.bildungsplaene-bw.de/,Lde/LS/BP2016BW/ALLG/SEK1/INF7/PK/02>

2.3 Kommunizieren und Kooperieren

<https://www.bildungsplaene-bw.de/,Lde/LS/BP2016BW/ALLG/SEK1/INF7/PK/03>

2.4 Analysieren und Bewerten

<https://www.bildungsplaene-bw.de/,Lde/LS/BP2016BW/ALLG/SEK1/INF7/PK/04>

1.2 Algorithmisches Problemlösen

Gerade im Bereich der Informatik werden Anwendungsprobleme in der Regel unter Zuhilfenahme von Algorithmen gelöst. Die Unterrichtseinheit mit dem Calliope mini zeichnet sich besonders durch die Förderung des algorithmischen Problemlösens aus.

Nach Hildebrandt und Matzner (2023) umfasst das algorithmische Problemlösen:

„**AP1** das Erkennen, dass eine Problemsituation existiert.

AP2 das Verstehen der gegebenen Situation und des Ziels.

AP3 die Spezifizierung der verschiedenen (Teil-)Probleme durch Reduzierung bzw. Abstraktion auf die für die Lösung des Problems relevanten Aspekte.

AP4 das Entwerfen einer ausführbaren Handlungsvorschrift (die auch aus mehreren einzelnen verknüpften Handlungsvorschriften bestehen kann) zur Lösung des Problems, die bestimmte Merkmale erfüllen muss, und zwar

- ist die Handlungsvorschrift nicht unendlich lang,
- liefert die Handlungsvorschrift nach endlicher Zeit ein Ergebnis,
- ist die Handlungsvorschrift so präzise formuliert, dass die Schritte zur Ausführung eindeutig sind,
- gibt es für die Handlungsvorschrift keine formale Vorschrift, d.h., sie kann zum Beispiel in Textform, in sogenanntem Pseudocode, in Form einer grafischen Darstellung oder als Programm in einer Programmiersprache angegeben werden.

AP5 das Überprüfen und Bewerten sowie ggf. Überarbeiten der Lösung (bzw. des Ergebnisses) unter Einbeziehung der Ausgangssituation und des Ziels.“ (Hildebrandt & Matzner, 2023, S. 138)

Auch in der Grundschule und den frühen Klassenstufen der weiterführenden Schulen ist es bedeutsam, eigene Lösungen bzw. Ergebnisse altersangemessen zu überprüfen und zu bewerten (siehe AP5). Allerdings kann der Weg zur Problemlösung durchaus weniger strukturiert erfolgen als in den späteren Klassenstufen der Sekundarstufe I und II. Ein wesentliches methodisches Prinzip der Einheit ist das entdeckende Lernen nach Hartmann, Näf und Reichert (2007). Sie betonen, dass entdeckendes Lernen explizit Gewicht auf Aspekte wie „selbständiges Arbeiten, Kreativität und kritische Reflexion“ (Hartmann et al., 2007, S. 65) legen soll. Themenwahl und Vorbereitung sind entscheidende Voraussetzungen für das praktische Erleben und Erforschen in der Grundschule. Durch Trial-and-Error können die Schüler:innen eigene Lösungswege entdecken, Hypothesen aufstellen und durch ihre Programme optimieren. Aber auch in der Grundschule ist bereits systematisches Vorgehen denkbar.

Um dieses entdeckende Lernen bestmöglich zu begleiten, empfiehlt sich der Einsatz von Teamteaching. Besonders in einer Lerngruppe, in der viele Kinder keine Vorerfahrung mit Programmieren oder dem Umgang mit Mikrocontrollern haben, profitieren sie von einer anfänglich intensiven individuellen Unterstützung bei Problemen. Zwei Lehrkräfte können flexibler auf Fragen und Schwierigkeiten eingehen, schneller Hilfestellung geben und die Schüler:innen dazu ermutigen, eigene Lösungsansätze auszuprobieren. Alternativ können auch externe Expert:innen aus der Informatik oder technikaffine Schüler:innen, die beispielsweise in einer Programmier-AG aktiv sind, unterstützend in den Unterricht eingebunden werden. Dadurch entsteht eine offene Lernatmosphäre, in der der Austausch und das gemeinsame Problemlösen im Mittelpunkt stehen.

Durch die Kombination aus entdeckendem Lernen, individueller Unterstützung und der gezielten Förderung algorithmischer Problemlösekompetenzen bietet diese Unterrichtseinheit eine wertvolle Grundlage, um Schüler:innen frühzeitig an die Welt der Informatik heranzuführen.

Trotz der spielerischen Herangehensweise sind einige zentrale Grundlagen notwendig, um ein fundiertes Verständnis für das Programmieren zu schaffen, die im Folgenden dargestellt werden. Eine wesentliche Basis bildet das EVA-Prinzip, um die Funktionsweise von Informatiksystemen auf einer elementaren Ebene darzustellen.

1.3 EVA-Prinzip

In Gablers Wirtschaftslexikon kann man nach Mathes und Seufert (2022, S. 3) folgende Definition von Informatik finden:

„Informatik ist die Wissenschaft von der systematischen Erhebung von Informationen, deren Verarbeitung zu Daten und deren zweckmäßige Ausgabe, um damit einen zusätzlichen Nutzen zu generieren.“

In diesem Zusammenhang sprechen die Autoren vom sogenannten **EVA-Prinzip**: **E**ingabe, **V**erarbeitung, **A**usgabe. Dieses Prinzip ist auch beim Calliope mini erkennbar: Sensordaten werden als Eingaben erfasst, im Mikrocontroller verarbeitet und beispielsweise durch LEDs oder Töne als Ausgabe dargestellt. Komplexere Funktionsweisen können mit der Erweiterung um interne Weitergabe beschrieben werden, welche dann auch abstraktere Daten als die Eingabedaten über Tastatur o. ä. darstellen würden und komplexere Softwarearchitektur ermöglicht.

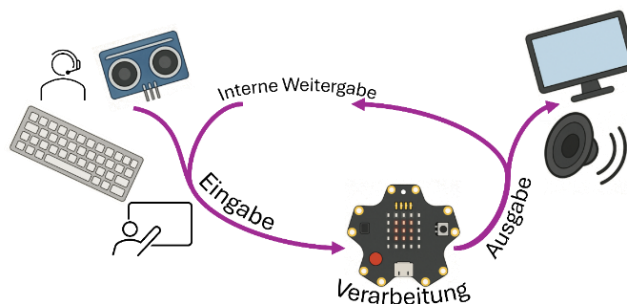


Abbildung 1 Mit dem EVA-Prinzip kann die Funktionsweise eines Systems als Eingabe, Verarbeitung und Ausgabe beschrieben werden (Bild von den Autor:innen).

1.4 Die Aktoren und Sensoren des Calliope mini

Der Calliope mini ist mit unterschiedlichsten Sensoren und Aktoren ausgestattet. Hier sind die vielfältigen Eingabe- und Ausgabemöglichkeiten des Calliope mini zu finden:

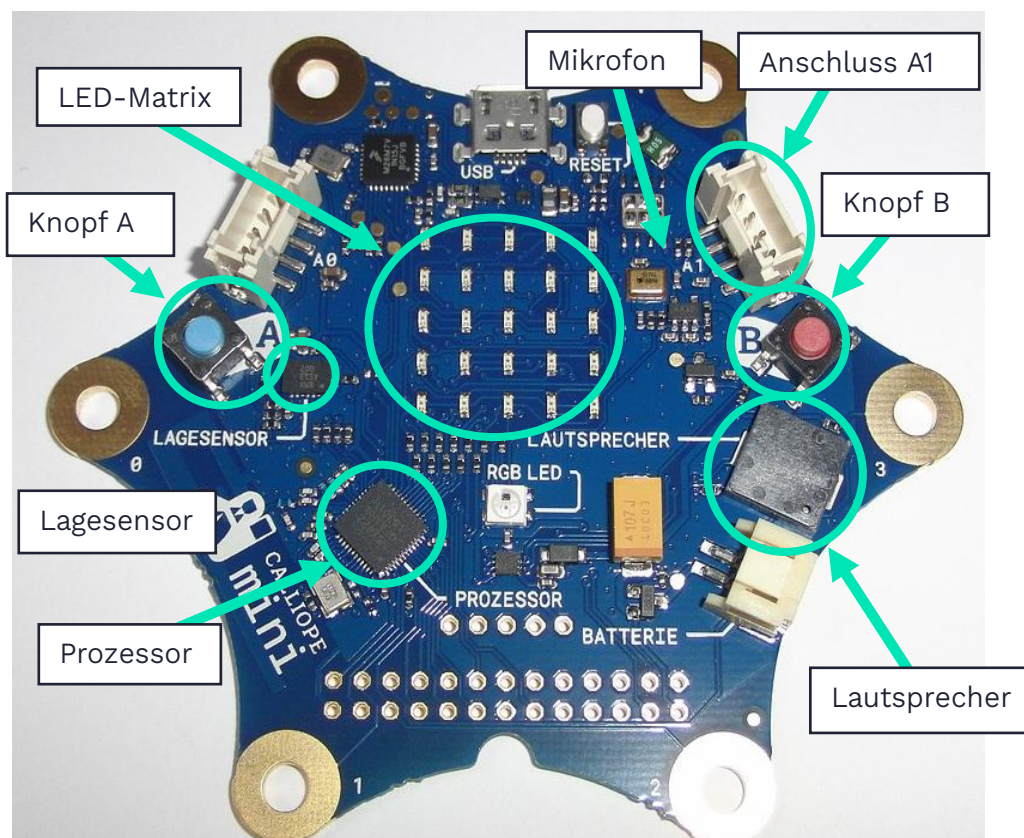


Abbildung 2 Der Calliope mini von der Calliope gGmbH

Weiterführende Informationen sind hier zu finden:

<https://calliope.cc/calliope-mini/uebersicht>

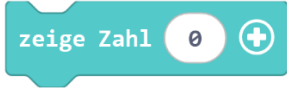
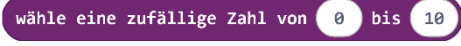
https://inf-schule.de/informatiksysteme/calliope/aktoren_und_sensoren

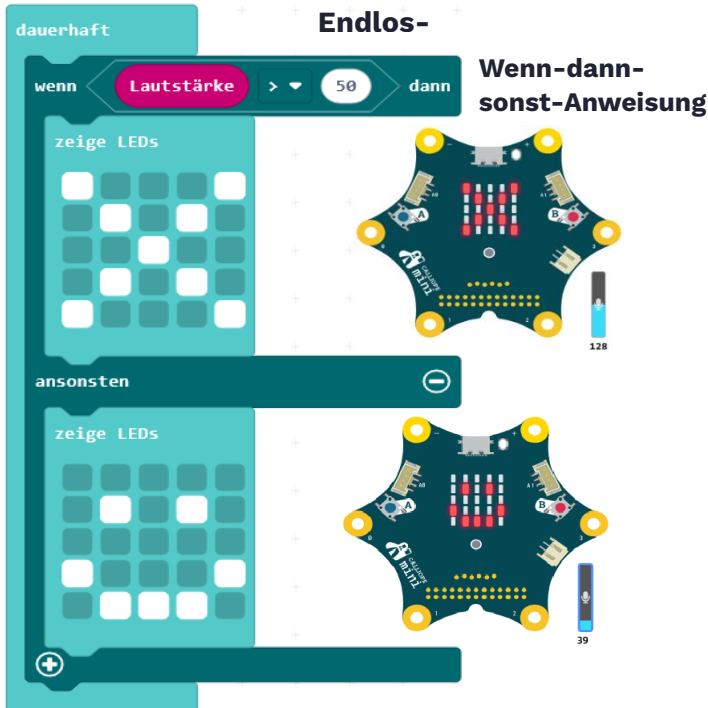
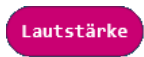
Die folgenden Inhalte beziehen sich auf den Einsatz des Calliope mini (Version 2.1) in Kombination mit iPads.

1.5 Bausteine zum Programmieren

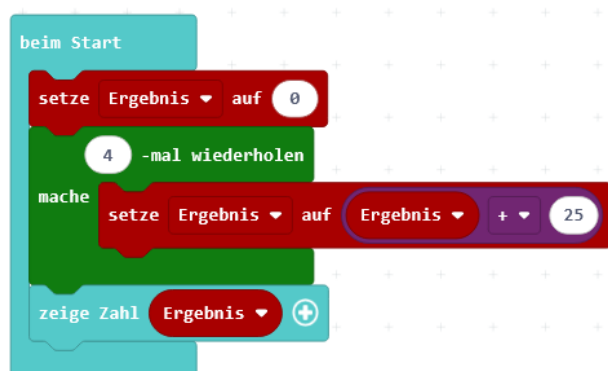
Im MakeCode-Editor (<https://makecode.calliope.cc/>) gibt es verschiedenen Programmierbausteine. In den folgenden drei Tabellen werden in diesen Materialien verwendete Programmierblöcke erklärt, Programmierbeispiele für Einschübe und Erweiterungen vorgestellt und relevante informatische Fachbegriffe dargestellt.

Verwendete Programmierblöcke in MakeCode / relevante Fachbegriffe (Grafiken: selbsterstellte Screenshots der Programmierumgebung Microsoft MakeCode - © Microsoft/MIT-Lizenz)	Funktionsweise
Anweisung	Führt die Programmbausteine, welche innerhalb dieses Blocks stehen, beim Start des Calliope mini einmalig aus.
Anweisung, Endlosschleife	Führt die Programmbausteine, welche innerhalb dieses Blocks stehen, beim Start des Calliope mini dauerhaft aus. Dieser Programmteil läuft also automatisch immer wieder von vorne durch.
Bedingte Anweisung: if (wenn)-Anweisung: eine einfache Kontrollstruktur Variable	Sobald Knopf A gedrückt wird, werden die Programmbausteine innerhalb dieses Blocks ausgeführt. Die Anweisungen in diesem Block werden nur dann ausgeführt, wenn die Bedingung erfüllt, in diesem Fall der Wert der Variablen „geklickt“ „wahr“ bzw. „1“ ist. Das ist der Fall, wenn Knopf A (Eingabe) des Calliope mini gedrückt wird. Variable „geklickt“ wird in diesem Fall durch ein pinkes Oval dargestellt. Dieser Variable können in diesem Fall nur die Werte „wahr“ bzw. „1“ oder „falsch“ bzw. „0“ zugewiesen werden.
Bedingte Anweisung innerhalb des Programms	Die Anweisungen innerhalb dieses Blockes werden nur dann ausgeführt, wenn die Bedingung , die bei „wahr“ eingefügt wird, erfüllt ist.
Logischer Ausdruck, der einen Wahrheitswert (wahr oder falsch) liefert	Es werden zwei Werte miteinander verglichen und der logische Ausdruck liefert dann „wahr“ oder „unwahr/falsch“.

 Verzweigung	If-then-else-Anweisung (Wenn-dann-sonst-Anweisung) Führt das Programm aus, welches oben innerhalb dieses Blocks stehen, falls die Bedingung „wahr“ ist. Liefert der logische Ausdruck „unwahr“, dann wird das Programm ausgeführt, welches unten innerhalb dieses Blocks, nach „ansonsten“ stehen. Anstelle des Wahrheitswerts „wahr“ kann ein <i>logischer Ausdruck</i> im Kopf der Anweisung stehen (z. B. Variable a > Variable b) und dann wahr oder unwahr liefert.
 Anweisung	Bei diesem türkisen Element handelt es sich um eine einzelne Anweisung „Zahl 0 über die LED-Matrix des Calliope mini anzeigen“. Die Anzeigedauer kann über das „+“ geändert werden.
 Anweisung	Tonwiedergabe Bei diesem orangenen Element handelt es sich um eine einzelne Anweisung „Mittleres C über den integrierten Lautsprecher für einen Schlag ausgeben“. Der Calliope mini spielt als Ausgabe einen gewählten Ton für eine gewählte Dauer ab. Ton und Dauer können geändert werden.
 Variable	Der ovalen länglichen Variable „wähle eine zufällige Zahl von 0 bis 10“ wird eine Zufallszahl (eine ganze Zahl) zwischen 0 und 10 zugewiesen. Das Minimum und Maximum können innerhalb geändert werden oder durch weitere Variablen gesetzt werden.
 Variable Der Ultraschallsensor, in diesem Fall an Pin C16 (A1 RX) angeschlossen, misst den Abstand vom Sensor zu davor befindlichen Objekten in cm.	Messwertverarbeitung Ermittelte Entfernung (in cm) vom Ultraschallsensor, der an Pin C16 (A1 RX) angeschlossen ist, beispielsweise zu einem Hindernis wird der ovalen und violetten Variable zugewiesen (Eingabe). Dieser Block ist nur verfügbar, wenn die Erweiterung „Grove“ installiert wurde (dafür in „Make-Code“ unter „Erweiterungen“ „Grove“ anklicken).

Programmierbeispiele	Beschreibung
„Ein Würfel“  Bedingte Anweisung, Anweisung, Variable Immer dann wenn der Calliope mini geschüttelt wird, wird eine zufällige Zahl ermittelt und über die LED-Matrix am Calliope mini angezeigt.	Die Anweisung in diesem Block wird immer genau dann ausgeführt, wenn die Bedingung, in diesem Fall, dass der Calliope mini geschüttelt wird, „wahr“ ist. Der ovalen länglichen Variable „wähle eine zufällige Zahl von 1 bis 6“ wird eine Zufallszahl (eine ganze Zahl) zwischen 1 und 6 zugewiesen (Verarbeitung). Die Zufallszahl wird über die LED-Matrix am Calliope mini angezeigt (Ausgabe).
„Lärmalarm“ Nach dem Programmstart wird der Lautstärkesensor (Eingabe) immer wieder überprüft, ob ein Wert über 50 erreicht wird (Verarbeitung). Wenn der Lautstärkesensor einen Wert größer als 50 hat, dann wird ein „X“ auf der LED-Matrix angezeigt, ansonsten wird ein Smiley auf der LED-Matrix angezeigt (Ausgabe).  Die Anweisungen in diesem Block werden „dauerhaft“, also fortlaufend, wiederholt. Es handelt sich um eine Endlosschleife. Lautstärkesensor  Das Mikrofon des Calliope mini (siehe Abbildung 2) kann als Lautstärkesensor genutzt werden und die Umgebungslautstärke messen. Die Lautstärke wird im Wertebereich 0 (leise) – 250	If-then-else-Anweisung (Wenn-dann-sonst-Anweisung) Die Anweisungen in diesem Block werden „dauerhaft“, also fortlaufend, wiederholt. Wenn der Variablen Lautstärke ein Wert größer als 50 zugewiesen wird, dann wird ein X auf der LED-Matrix angezeigt, ansonsten (also, wenn der Wert, der der Variablen Lautstärke zugewiesen wird, 50 oder kleiner ist), wird ein Smiley auf der LED-Matrix angezeigt (Ausgabe).

(laut) ausgegeben und der Variablen „Lautstärke“ permanent zugewiesen (**Eingabe**).



Vorab muss eine neue Variable erzeugt werden, in diesem Fall „Ergebnis“.

Der grüne Block stellt eine sog. **For-Schleife** dar, mit der man die Anweisungen innerhalb dieses Blockes mit einer bestimmten Anzahl von Wiederholungen, in diesem Fall vier Wiederholungen, ausführt (**Verarbeitung**).

Zählschleife zur Wiederholung von Anweisungen

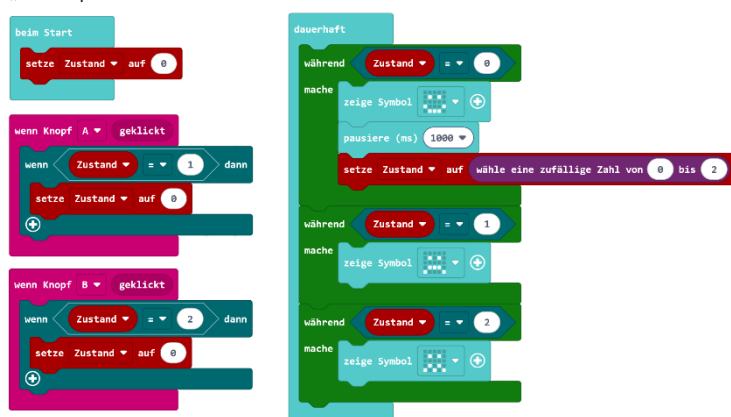
Beim Start des Programms wird der Variablen „Ergebnis“ der Wert 0 zugewiesen.

Anschließend werden die Anweisungen in der For-Schleife 4-mal ausgeführt (**Verarbeitung**):

1. Durchlauf: Ergebnis=Ergebnis + 25
[Ergebnis = 0 + 25 = 25]
2. Durchlauf: Ergebnis=Ergebnis + 25
[Ergebnis=25 + 25 = 50]
3. Durchlauf: Ergebnis=Ergebnis + 25
[Ergebnis=50 + 25 = 75]
4. Durchlauf: Ergebnis=Ergebnis + 25
[Ergebnis=75+25=100]

Die der Variablen „Ergebnis“ zugewiesene Zahl (100) wird über die LED-Matrix am Calliope mini angezeigt (**Ausgabe**).

„Calli-pet“



Endlosschleife, While-Schleife, logischer Ausdruck, Variable

Sequenz: Mehrere Anweisungen bilden eine Sequenz.

Zu Beginn wird „Zustand“ auf „0“ gesetzt und in der Endlosschleife überprüft. Dabei wird die erste While-Schleife ausgeführt. Nun wird der Variablen „Zustand“ ein Wert zwischen 0 und 2 zugewiesen – damit könnte die Bedingung für diese Schleife nicht mehr gegeben sein. Wird dann aufgrund des Wertes der Variable „Zustand“ eine andere Schleife ausgeführt, wird so lange ein bestimmtes Frowny gezeigt, bis der entsprechende Knopf gedrückt wird.

Fachbegriff	Erklärung
Anweisung	Eine Anweisung ist ein einzelner Befehl in einem Programm.
Bedingte Anweisung	Eine Anweisung, die nur ausgeführt wird, wenn eine bestimmte Bedingung erfüllt ist. Beispiel: „Wenn die Lautstärke über 50 ist, zeige einen Frownie“.
Endlosschleife	Eine Schleife, die nicht automatisch endet, sondern dauerhaft läuft. Sie wird z. B. genutzt, um einen Spielstand immer wieder zu überprüfen.
Messwertverarbeitung	Die Messwertverarbeitung ist der Teil eines Programms, der gemessene Daten – z. B. von einem Sensor – aufnimmt, auswertet und weiterleitet .
Lautstärkesensor	Ein Sensor, der die Lautstärke der Umgebung misst . Der Lautstärkesensor des Calliope mini gibt Werte zwischen 0 (leise) und laut (250) zurück.
Schleife	Eine Programmstruktur, die Anweisungen wiederholt ausführt , welche sich in der Schleife befinden. Besondere Arten sind Zähl-, While- oder Endlosschleife.
Zählschleife	Eine Schleife, die eine bestimmte Anzahl von Wiederholungen hat. Beispiel: „Wiederhole 5-mal: Springe vorwärts“.
While-Schleife (Währenderschleife)	Auch While-Schleife; wiederholt die in der Schleife stehenden Anweisungen, solange wie der dazugehörige logische Ausdruck „wahr“ ist.
Wenn-dann-sonst-Verzweigung	Eine Entscheidung im Programm: Wenn eine Bedingung erfüllt ist, dann passiert etwas – sonst passiert etwas anderes.
Variable	Eine Art Speicherplatz, der einen bestimmten Wert „festhält“, z. B. „Punkte = 10“. Der Wert kann sich im Laufe des Programms ändern.

1.6 Hintergrund - Literaturverzeichnis

1. Baum, Kevin; Kirsch, Nadine; Reese, Kerstin; Schmidt, Pascal; Wachter, Lukas; Wolf, Verena (2019). Informatikunterricht in der Grundschule? – Erprobung und Auswertung eines Unterrichtsmoduls mit Calliope mini. 18. GI-Fachtagung Informatik und Schule. Informatik für alle. Bonn: Gesellschaft für Informatik. pp. 49-58. DOI: 10.18420/infos2019-b1.
2. Best, Alexander; Borowski, Christian; Büttner, Katrin; Freudenberg, Rita; Fricke, Martin; Haselmeier, Kathrin; Herper, Henry; Hinz, Volkmar; Humbert, Ludger (federführend); Müller, Dorothee; Schwill, Andreas; Thomas, Marco (2019): Kompetenzen für informatische Bildung im Primarbereich. Bonn: Gesellschaft für Informatik e.V. DOI: 10.18420/rec2019_061.
3. Hartmann, W., Näf, M., Reichert, R. (2007). Informatikunterricht planen und durchführen. Berlin, Heidelberg: Springer-Verlag.
4. Hildebrandt, Claudia, Matzner, Matthias (2023). Charakterisierung von Leistungsgruppen beim algorithmischen Problemlösen. In: Informatikunterricht zwischen Aktualität und Zeitlosigkeit: 20. GI-Fachtagung Informatik und Schule, INFOS 2023, Würzburg, Germany, September 20-22, 2023. Ed. by Lutz Hellmig and Martin Hennecke. Vol. P-336. LNI. Gesellschaft für Informatik, Bonn, 2023, pp. 137–146. DOI: 10.18420/INFOS2023-012.
5. Ministerium für Kultus, Jugend und Sport Baden-Württemberg (2016a). Bildungsplan der Grundschule. Sachunterricht. <https://www.bildungsplaene-bw.de/,Lde/LS/BP2016BW/ALLG/GS/SU>
6. Ministerium für Kultus, Jugend und Sport Baden-Württemberg (2016b). Gemeinsamer Bildungsplan der Sekundarstufe I. Aufbaukurs Informatik. <https://www.bildungsplaene-bw.de/,Lde/LS/BP2016BW/ALLG/SEK1/INF7>
7. Mathes, M.A., Seufert, J. (2022). Was ist Informatik? In: Programmieren in C++ für Elektrotechniker und Mechatroniker. Wiesbaden: Springer Vieweg. DOI: 10.1007/978-3-658-38501-9_1.
8. Modrow, E., Mönig, J., Strecker, K. (2011). Wozu Java? Plädoyer für grafisches Programmieren. In: Log In, Heft Nr. 168, S. 35–41.
9. Springer Fachmedien Wiesbaden GmbH: Gabler Wirtschaftslexikon. <https://wirtschaftslexikon.gabler.de>

2. Erste Doppelstunde

2.1 Thema

Einführung in das handlungsorientierte algorithmische Problemlösen anhand der Programmierung des Calliope mini. In der ersten Doppelstunde werden die Schüler:innen spielerisch an das Konzept von Algorithmen und das Programmieren herangeführt. Sie entdecken explorativ das blockbasierte Programmieren mit Calliope mini und wenden das EVA-Prinzip an.

2.2 Benötigtes Material

- ein Calliope mini (hier angepasst an die Version 2.1) + Erweiterungsset mit Ultraschallsensor (für die zweite Doppelstunde) wie zum Beispiel „Sensorbox - 20 Grove Sensoren (10 Ultraschall- und 10 Feuchtigkeitssensoren)“ pro Person
- digitale Endgeräte (Laptops, Tablets), bluetoothfähig oder mit USB-Anschluss
- installierte Calliope mini App (siehe unten unter Software) oder Online-Aufruf des Editors von MakeCode (<https://makecode.calliope.cc/>)
- Tafel/digitale Alternative
 - (ausgedrucktes) Material, für alle je 1-mal: 01-AB-EVA-Prinzip, Hilfestellung: 01-AB-EVA-Prinzip-Loesungsideen
 - für die Tafel: Programmierbausteine (siehe 2.5)
- Checkliste zur Vorbereitung auf die Stunde
 - ✓ **Startprogramm** wurde auf die Calliope mini übertragen. Dies mindert die Fehleranfälligkeit insbesondere für die Verbindung via Bluetooth. Eine Anleitung zur Ausführung des Startprogramms ist hier zu finden: <https://calliope.cc/programmieren/mobil/hilfe#startprogramm>
 - ✓ **Calliope mini App** wurde auf den genutzten digitalen Endgeräten oder eine Alternative dazu installiert (MakeCode kann auch über den Browser aufgerufen werden). Hier sind die Apps zu finden: <https://calliope.cc/programmieren/mobil>
<https://calliope.cc/programmieren/tools>
 - ✓ Eine stabile Internetverbindung ist für alle Endgeräte vor Ort vorhanden.
 - ✓ Die genutzten digitalen Endgeräte (iPads, Laptops) sind aufgeladen und für den Notfall ist vor Ort ein Stromanschluss vorhanden.
 - ✓ Alle benötigten Arbeitsblätter und Druckvorlagen wurden gedruckt bzw. sind digital vorhanden.

Software



Calliope mini App

<https://calliope.cc/programmieren/mobil>



Microsoft MakeCode

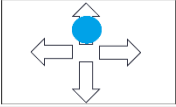
<https://makecode.calliope.cc>

2.3 Kompetenzen

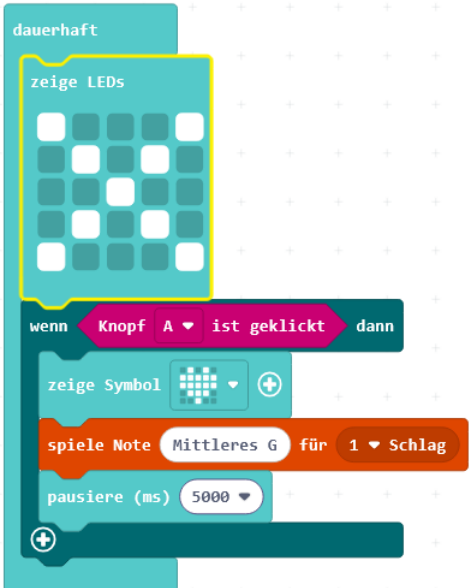
Die Schüler:innen

- führen gegebene Programme nach Anleitung aus, indem vorhandene Calliope-mini-Programme mit dem entsprechenden Werkzeug geöffnet, auf den Calliope mini übertragen und ausgeführt werden (AFB I).
- stellen Algorithmen in verschiedenen (formalen) Darstellungsformen dar, indem sie die vorgefertigten Programmierbausteine sowie die visuelle Programmierumgebung (hier MakeCode) zur Entwicklung von Algorithmen nutzen (AFB II).
- entwerfen, realisieren und testen Algorithmen mit den algorithmischen Grundbausteinen Anweisung, Sequenz, Wiederholung, bedingte Anweisung und Verzweigung, indem sie Programme zur Lösung einfacher Aufgaben/Probleme für den Calliope mini entwerfen, implementieren und testen (AFB II/III).
- programmieren ein Informatiksystem, indem sie Programme für den Calliope mini schreiben und testen (AFB II/III).
- stellen Algorithmen dar und vergleichen diese unter Verwendung einer altersgerechten, informatischen Fachsprache (AFB II/III).
- geben grundlegende, allgemeingültige Beschreibungen der Funktion und Arbeitsweise von Informatiksystemen an (EVA-Prinzip), indem sie die Programme für den Calliope mini hinsichtlich der Komponenten des EVA-Prinzips untersuchen und den entsprechenden Teilbereichen begründet zuordnen (AFB II/III).

2.4 Verlaufsplan der ersten Doppelstunde

U-Phase	S-/L-Aktivität (Grafiken: selbsterstellte Bilder bzw. selbsterstellte Screenshots der Programmierungsumgebung Microsoft MakeCode – © Microsoft/MIT-Lizenz)	Sozialform	Medien, Material
	Vor der Unterrichtsstunde: Verteilen der Tablets/Notebooks und Calliope minis → 1 bis 2 pro 2er-Team, Geräte dürfen nicht ohne Erlaubnis der Lehrkraft ausgepackt und benutzt werden.		
Einstieg mit Team- Teaching 15 Min.	- Begrüßung; Überblick über die Inhalte der Stunde: Einführung in das handlungsorientierte algorithmische Problemlösen - L2 erstellt abgewandt von L1 und nur für S sichtbar eine erste Zeichnung. - L1 steht mit dem Rücken und mit Gehörschutz (oder Ohrstöpseln) zur Tafel mit Kreide zum Zeichnen in der Hand. L1 sieht zu zeichnende Abbildung nicht. Während der Phase, wo die S die Anweisungen zum Zeichnen an die Tafel bringen, kann L1 auch den Raum verlassen. - Anweisungen zum Zeichnen (Programmierbausteine) werden an einer Tafelseite durch S in eine Reihenfolge gebracht, Bsp.: Zeichnung  Anweisungen an der Tafel Zeichne (Kreis, Dreieck, Quadrat)  Gehe nach links, rechts, oben, unten  Zeichne (Kreis, Dreieck, Quadrat) 	Plenum, GA/PA S-S-Interaktion; L1/S-Tätigkeit PA, Plenum	Programmierbausteine (siehe 2.5), Magnete, Tafel, Kreide (oder digitale Alternative), Edding, Kreppklebeband ggf. Gehörschutz Weitere Beispielzeichnungen (siehe 2.6)

	Interpretation), auch wenn Figuren dann z. B. übereinander gezeichnet werden. Wenn bei dem Beispiel-  programm die Anweisung „Gehe nach oben“ fehlt, könnte folgende Zeichnung entstehen: • Bei Bedarf: S denken sich weitere zu zeichnende Abbildungen aus und geben im Plenum L1 entsprechende Anweisungen. • Bei Bedarf: S zeichnen und „programmieren“ sich in PA gegenseitig anhand der vorgegebenen Programmierbausteine (am Platz oder an der Tafel) • L2 moderiert, leitet zu informatischem Kontext über: Wenn die sog. Handlungsanweisungen nicht eindeutig formuliert sind, dann können unterschiedliche Zeichnungen herauskommen, da die Menschen über die Handlungsanweisungen hinaus ihre Erfahrungen mit hineinbringen. Informatiksysteme sind in der Regel nicht in der Lage, nicht eindeutige Anweisungen auszuführen.		
Einstieg ohne L2 15 Min.	• In Dreiergruppen: S1 zeichnet eine Abbildung mit Quadraten, Dreiecken und Kreisen in das auf ein Blatt Papier. • S2 erstellt mit vorgegebenen Programmierbausteinen, die vorher für die Dreiergruppen mehrfach ausgedruckt werden müssen, die Handlungen, die nötig sind, damit die Zeichnung von S1 „herauskommt“. • Zeichnung von S1 wird verdeckt, nun nur noch die Handlungsanweisungen von S2 sind zu sehen • S3 zeichnet nun nach der Handlungsvorschrift von S2 das Bild. • Nach Vollendung können S die beiden Bilder vergleichen und besprechen mögliche Gründe bei Abweichungen. • Fehler oder besser „Fragen“, gibt es hier an vielen Stellen – nicht zuletzt in der Interpretation, welche Programmierbausteine was bedeuten sollen – aber eben auch ganz informatischer Art: Wird die Position des „Stiftes“ gespeichert oder wird immer wieder von der Mitte angefangen? Bedeutet „Drehe“ eine Drehung um 90 oder um 180 Grad? Oft aber auch, ob man weitere geometrische Formen in „zeichne ____“ einbauen kann, für „gehe nach ____“ spezifischere Beschreibungen hineinschreiben darf und ähnliches.	Plenum Gruppenarbeit S-S-Interaktion	Weitere Beispielzeichnungen (siehe 2.6) DIN A4 Blätter: 2 je S

Überleitung 15 Min.	- Tablets/Notebooks und Calliope minis werden gemeinsam für den Einsatz vorbereitet; S folgen den Schritten, die L1 für alle durchführt (siehe bei Bedarf unter 4 Programmierungsumgebung und 5 Programme übertragen im Arbeitsheft unter https://calliope.cc/schulen/arbeitsheft, ggf. 01-Hinweisblatt.pdf). - Beispielprogramm (siehe rechts) wird mit den S beispielhaft nach und nach erstellt, besprochen und gemeinsam auf den Calliope mini übertragen (evtl. Musik verbieten, wenn Geräuschpegel bzw. Ablenkungspotenzial zu hoch). - Anhand des Beispielprojekts werden Eingabe, Ausgabe und Verarbeitung (EVA-Prinzip) erwähnt. Taste A ist ein Eingabemodul, das LED-Feld und der Lautsprecher sind Ausgabemodule. - Speicherung des ersten Projekts durch die S auf dem lokalen Gerät. 	Plenum	Projektion via Beamer; Tablets/Notebooks mit MakeCode, Calliope mini
Erarbeitung 28 Min.	- S sollen selbstständig das Einstiegsprojekt verändern, testen und erneut verändern, so dass sie schließlich ein eigenes „Produkt“ entwickeln. - Gegen Ende dieser Phase wird das erste Arbeitsblatt (Das EVA-Prinzip – Arbeitsblatt 1) verdeckt auf die S-Tische gelegt und die EVA-Schilder werden an die Tafel gehängt. EVA-Schilder: 	PA	Calliope mini, Tablet/Notebook, Tafelhinweis für SuS: zwischen- durch die Projekte immer wieder speichern
Sicherung 20 Min.	- Vorstellung eines S-Produktes mit EVA-Elementen - Begriffe Eingabe, Verarbeitung, Ausgabe werden anhand des Beispielprogramms erklärt. - In der Informatik geht es darum, systematisch Informationen zu erheben, z. B. die Temperatur, Lautstärke usw. Diese Informationen werden in für die Informatiksysteme verarbeitbare Daten umgewandelt, verarbeitet, um eine zweckmäßige Ausgabe zu generieren, zum Beispiel das Piepen des	S-Aktivität, Plenum, L	Projektion via Beamer; Tablets/Notebooks mit MakeCode, Calliope mini

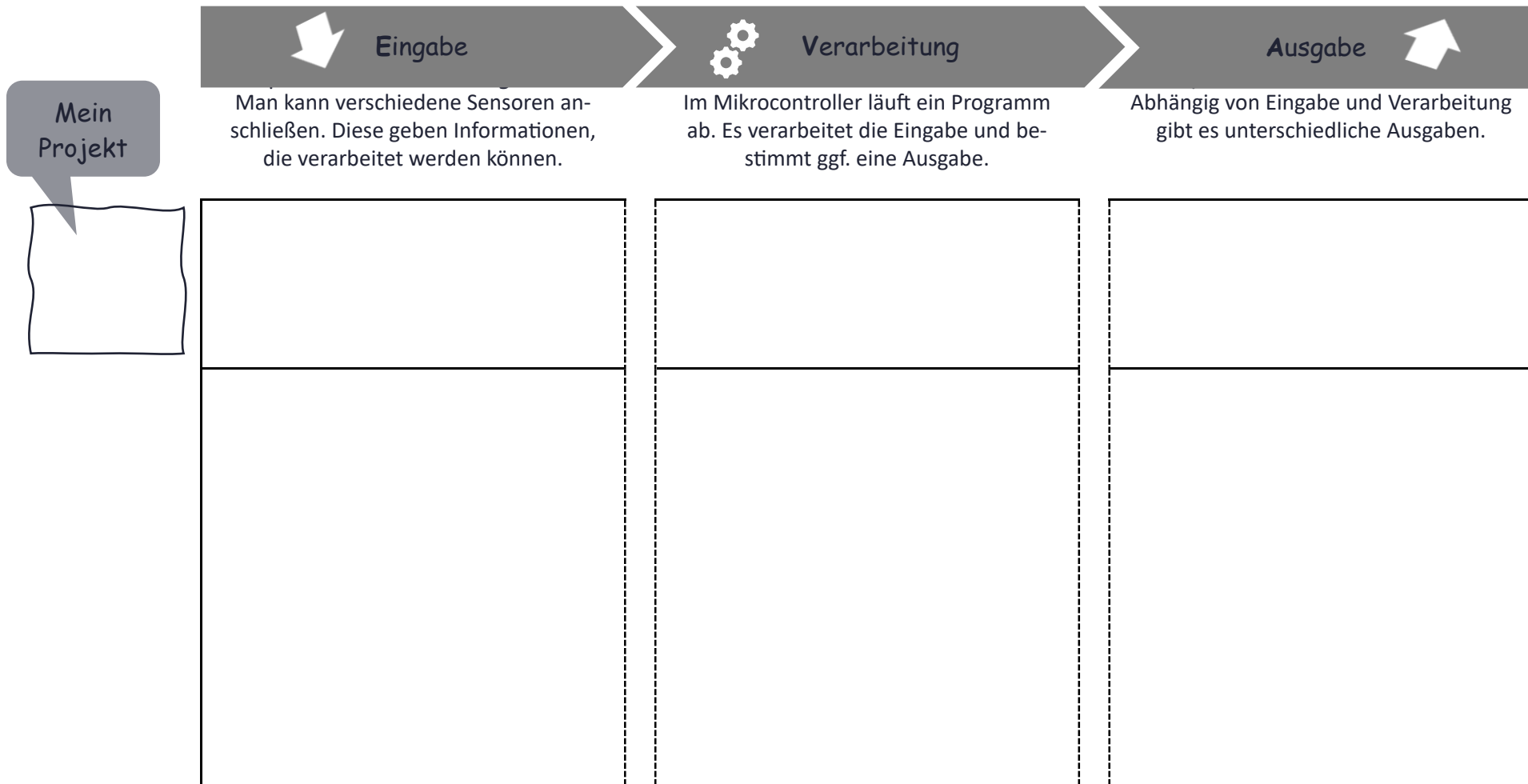
	Kühlschranks, wenn die gemessene Temperatur im Kühlschrank eine bestimmte, vorher festgesetzte Temperatur übersteigt. In diesem Zusammenhang spricht man vom sogenannten EVA-Prinzip: Eingabe, Verarbeitung, Ausgabe. Die Erhebung von Informationen und ihre Überführung in Daten ist in der Regel immer an bestimmte Zwecke gebunden. Dieses Prinzip ist auch im Calliope mini erkennbar: Sensordaten werden als Eingaben erfasst, im Mikrocontroller verarbeitet und durch LEDs oder Töne als Ausgabe dargestellt. - Fiktives Beispielprojekt: Wenn die Umgebungslautstärke größer gleich einen bestimmten Wert (hier 100) ist, dann wird dauerhaft ein Ton nach einer Pause von 1000ms = 1s abgespielt. - Sensordaten wie das Drücken der Taste A werden als Eingaben erfasst. Taste A ist ein Eingabemodul, das LED-Feld und der Lautsprecher sind Ausgabemodule. Bei der gemeinsamen Erarbeitung wird das Tafelbild entwickelt. Hilfekarten geben Formulierungshilfen. - S notieren auf dem Arbeitsblatt „Das EVA-Prinzip – Arbeitsblatt 1“ zu (mindestens) einem ihrer Programme Eingabe, Verarbeitung und Ausgabe. Das Hilfeblatt gibt Formulierungshilfen (siehe auch Tafelbild). - Eingabe: Umgebungslautstärke über den Lautstärkesensor, gemessen mithilfe des Mikrofons des Calliope mini - Verarbeitung durch den Prozessor: Der Variablen „Lautstärke“ wird der Wert der Umgebungslautstärke zugewiesen und mit dem Wert 100 verglichen. Wenn der Lautstärkewert größer gleich 100 ist, dann wird das mittlere C für einen Schlag gespielt (Ausgabe) und danach gibt es eine Pause von 1000ms = 1s. Es wird immer wieder geprüft, ob der Wert der Umgebungslautstärke größer oder gleich 100 ist und die entsprechenden Anweisungen (je nach Lautstärkewert) werden dann ausgeführt. - Ausgabe (bei entsprechender Umgebungslautstärke, Lautstärkewert größer gleich 100): Das mittlere C wird für einen Schlag gespielt und über das Mikrophon ausgegeben.		EVA-Schilder (hängen an seitlicher Tafel, parallel zur Projektionsfläche, EVA-Karten) AB EVA-Prinzip EVA-Schilder, Magnete, Hilfekarten geben Formulierungshilfen
Austausch 10 min	- Basar-Methode: pro 2er-Team bleibt eine Person am Platz und stellt ein Projekt vor (mit Calliope mini und Arbeitsblatt 01), jeweils andere Person bekommt ein Projekt vorgestellt, Wechsel von einer Person zur nächsten nach ca. 2 Minuten. - Nach ca. Hälfte der Zeit Rollenwechsel im 2er-Team	Basar	Calliope mini Arbeitsblatt „Das EVA-Prinzip – Arbeitsblatt 1“

Aufräumen 2 Min.	• S räumen auf.	S-Aktivität	
---------------------	---	-------------	--

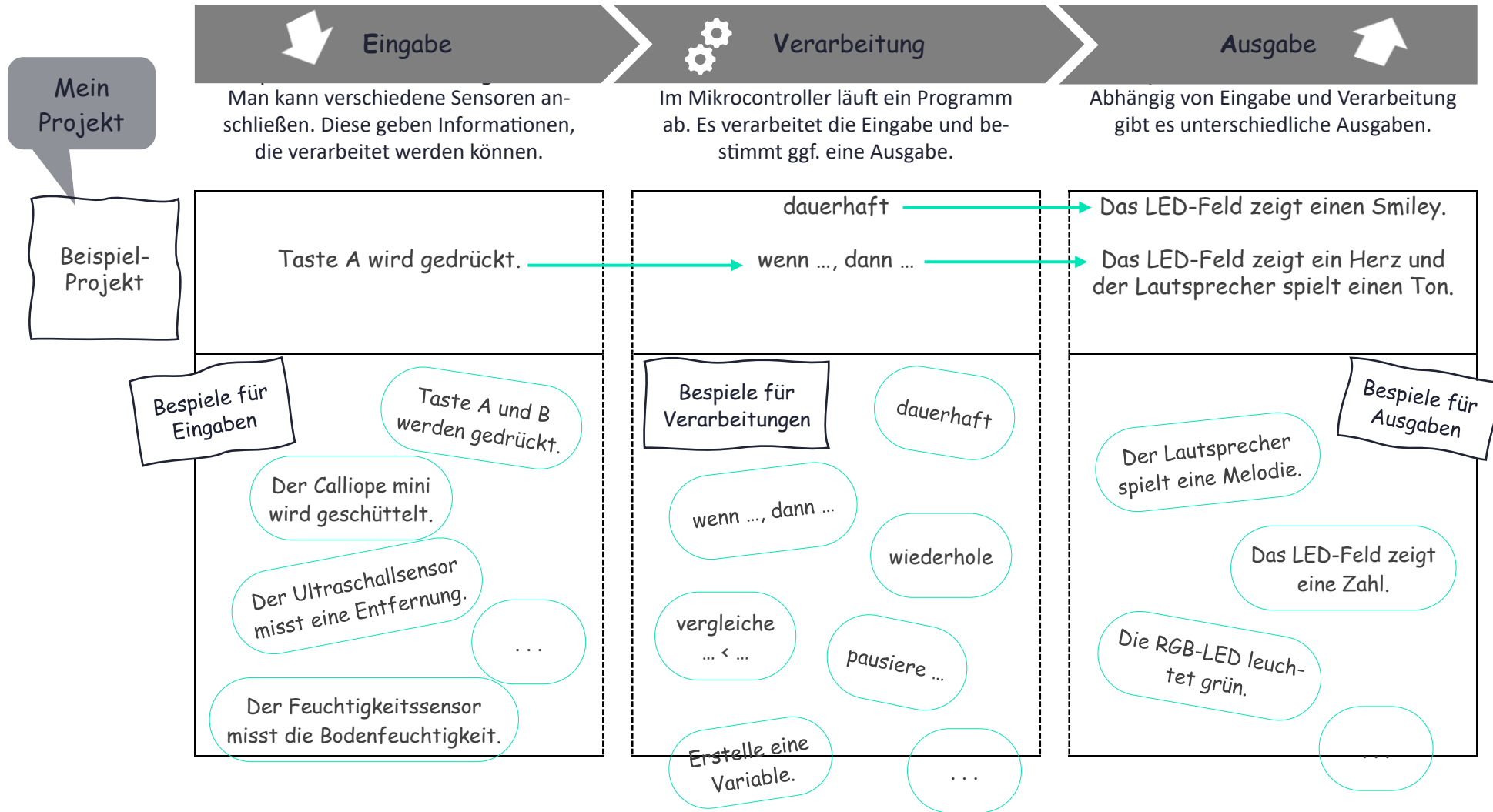
Abkürzungen: L: Lehrkraft, S: Schüler:innen, PA: Partnerarbeit, GA: Gruppenarbeit

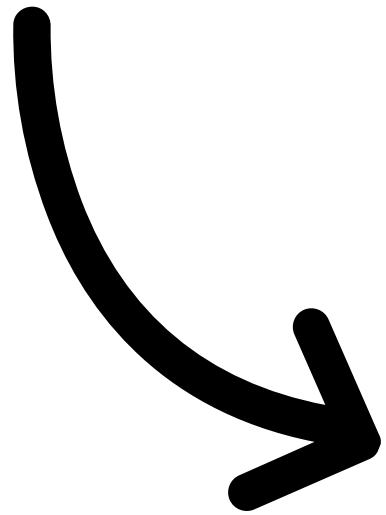
2.5 Arbeitsblätter und EVA-Schilder

Das EVA-Prinzip - Arbeitsblatt 1



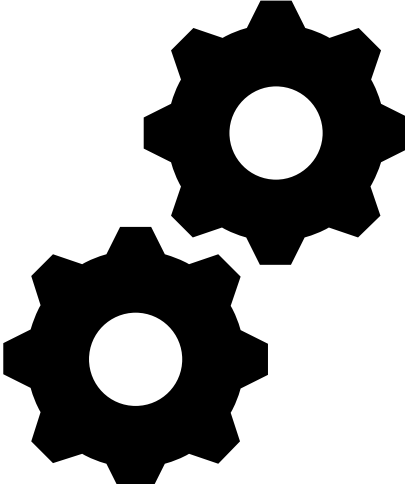
Das EVA-Prinzip – Arbeitsblatt 1, Lösungshinweise

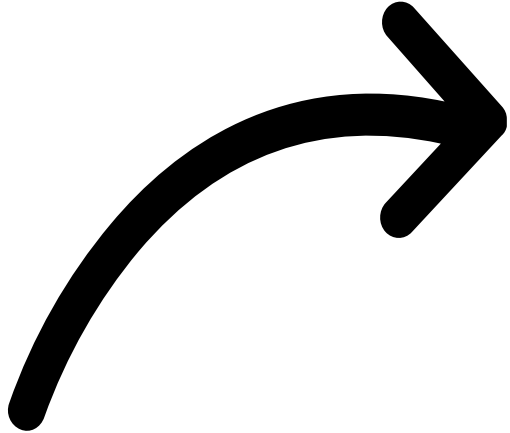




Eingabe

Verarbeitung

Two black interlocking gears are positioned in the upper right quadrant of the slide, above the word 'Verarbeitung'. The gears are of different sizes and are meshed together, symbolizing mechanical processing or data processing.



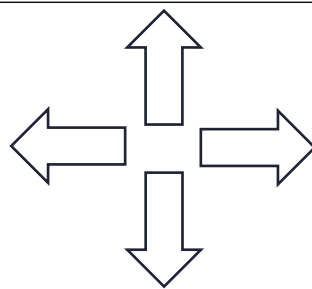
Ausgabe

2.6 Programmierbausteine und EVA-Karten

Die hier genutzten Programmierbausteine sind nicht eindeutig definiert und lassen daher Raum für individuelle Interpretationen. Gerade diese Mehrdeutigkeit macht den Umgang mit ihnen spannend und motivierend. Missverständnisse oder Fehler, die beim Programmieren entstehen, wirken dadurch eher amüsant – denn die Ursache liegt nicht eindeutig bei der programmierenden Person, sondern oft in der Offenheit der Bausteine selbst.

Gehe nach

links, rechts, oben, unten



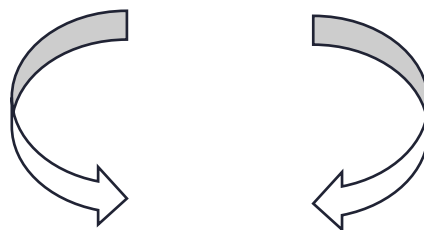
Zeichne

(Kreis, Dreieck, Quadrat)



Drehe

(links, rechts)

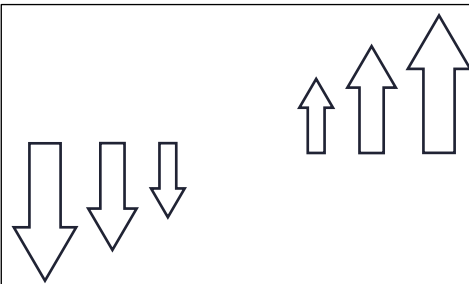


Wiederhole — mal



Größe

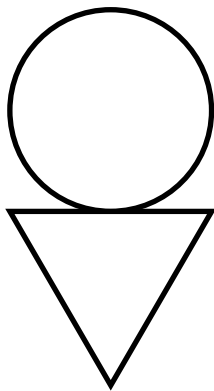
(verkleinere, vergrößere)



2.7 Mögliche Tafelbilder zum Einstieg

Im Plenum startet man zunächst mit einer einfachen Zeichnung, damit sich alle Lernenden bei der Programmierung einbringen können. Im Verlauf kann die Komplexität der Zeichnung steigen, um den Kindern Mut zu machen, sich auch an anspruchsvollere Programmierherausforderungen zu stellen. Beim Entwerfen der eigenen Zeichnungen in der Gruppe bzw. in Partnerarbeit wählen die Schüler:innen selbstdifferenzierend die Komplexität. Gegebenenfalls kann man die Kinder auch dazu ermutigen, sich eigene Programmblöcke auszudenken (z. B. „zeichne Smiley“), um ihre Zeichnungen zu realisieren

Mögliche Zeichnung

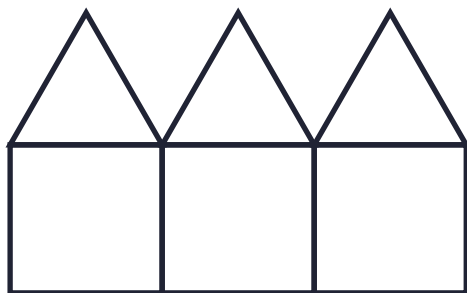


Mögliche Programmierbausteine, die die Schüler:innen nutzen können, damit gezeichnet werden kann:



Schwierigkeiten beim Zeichnen des Dreiecks: Wohin zeigt die Spitze?

Zeichnung auf dem Rücken:

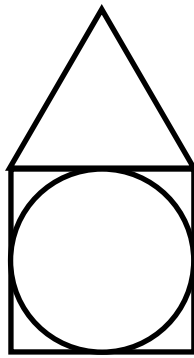
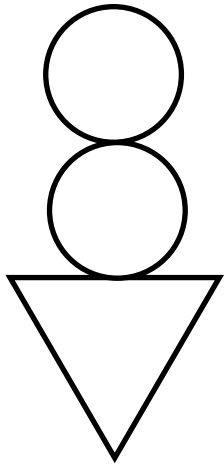
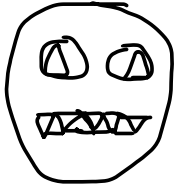


Mögliche Programmierbausteine, die die Schüler:innen nutzen können, damit gezeichnet werden kann:

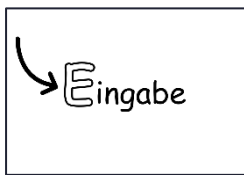


Schwierigkeiten: Positionen der Quadrate und Dreiecke nicht eindeutig bestimmt.

2.8 Weitere Beispielbilder



2.9 Mögliches Tafelbild – Sicherungsphase

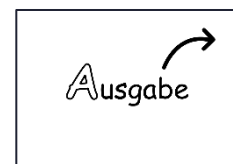


Der Lautstärkesensor (das Mikrofon) **misst** die Umgebungslautstärke.



Verarbeitung durch den Prozessor:

1. **Wenn** der Umgebungslautstärkewert größer gleich 100 ist, **dann** wird das mittlere C für einen Schlag gespielt (Ausgabe) und dann 1000ms = 1s pausiert.
2. **Wenn** der Lautstärkewert **nicht** größer gleich 100, dann gehe über zu Schritt 3.
3. Fange wieder bei Schritt 1 an („dauerhaft“).



Der **Lautsprecher spielt** das mittlere C für einen Schlag oder nicht.

Weitere Anregungen für den Unterricht finden Sie auch unter:
<https://calliope.cc/en/school/unterrichtsmaterial>

2.10 Weitere Erläuterungen zum spielerischen Einstieg: Algorithmen ohne Computer erleben

Zu Beginn der Unterrichtsstunde wird das Prinzip des algorithmischen Problemlösens mit Hilfe von Programmen spielerisch erfahrbar gemacht – ganz ohne digitale Geräte. In einem Rollenspiel übernehmen die Schüler:innen die Rolle von Programmierenden, während eine zweite Lehrkraft oder ein freiwilliges Kind als „Computer“ agiert. Die Aufgabe der Schüler:innen ist es, die an der Tafel stehende Person mithilfe vorgegebener Anweisungen so zu steuern, dass eine Figur aus Kreisen, Rechtecken und Dreiecken gezeichnet wird, die sich eine/einer der Schüler:innen vorher ausgedacht hat.

Die Person, die als „Computer“ fungiert, sollte vorher instruiert werden, die Anweisungen exakt und ohne Interpretation auszuführen. Fehlen beispielsweise Bewegungskommandos, kann es dazu führen, dass zwei Kreise direkt übereinander gezeichnet werden, anstatt nebeneinander. So erleben die Schüler:innen unmittelbar, wie Anweisungen präziser gestaltet werden können, damit ein Algorithmus korrekt und reproduzierbar ausgeführt wird.

Dieser spielerische Ansatz ermöglicht einen niedrigschwelligen Zugang zum blockbasierten Programmieren und zum Verständnis von Algorithmen. Gleichzeitig werden zentrale Herausforderungen des Programmierens deutlich: Ein „Computer“ führt Anweisungen so aus wie sie gegeben werden. Unklare Anweisungen wie „Drehe um“ oder „Vergrößere die Figur“ führen zu Missverständnissen, da die exakte Drehung oder Skalierung nicht genauer spezifiziert sind.

Durch diese Erlebnisse wird den Kindern auf anschauliche Weise vermittelt, warum präzise und eindeutige Anweisungen in der Programmierung essenziell sind – eine Grundlage, auf der sie später beim eigenen Programmieren mit dem Calliope mini aufbauen können.

2.11 Mögliche Stolperstellen in der explorativen Phase

„Das wird mir zu laut“

Ein besonders reizvoller Programmierblock für die Schüler:innen ist die Funktion „spiele Ton/Melodie“. Sobald dieser Block entdeckt wird, erklingen oft die unterschiedlichsten Melodien gleichzeitig im Klassenzimmer – der Geräuschpegel steigt entsprechend. Dass die Kinder solche Programmiermöglichkeiten selbstständig erkunden und kreativ für ihre Projekte nutzen, ist ausdrücklich erwünscht! Herausfordernd kann jedoch werden, dass sowohl die digitalen Endgeräte bereits während des Programmierens eine akustische Vorschau der Töne ausgeben als auch die Calliope mini nach der Übertragung des Programms.

Gerade die direkte Tonwiedergabe beim Programmieren führt häufig dazu, dass manche Kinder beim Ausprobieren verschiedener Melodien „hängen bleiben“, anstatt ihren Programmcode abzuschließen, auf den Calliope mini zu übertragen und sich neuen Programmiermöglichkeiten zu widmen.

Um dieser Herausforderung zu begegnen, kann mit der Klasse eine gemeinsame Vereinbarung getroffen werden: Tonwiedergaben dürfen nicht. Jede Gruppe darf nur ein Projekt mit Tonausgabe programmieren. Zudem sollten während der Besprechungsphasen die Calliope minis ausgeschaltet werden, damit keine unerwarteten Töne die Aufmerksamkeit stören. Auch die digitalen Endgeräte sollten in diesen Momenten lautlos gestellt oder die Bildschirme geschlossen bzw. auf den Tisch gedreht werden. Diese Maßnahmen

helfen dabei, den Geräuschpegel zu reduzieren und die Fokussierung auf die Unterrichtsinhalte zu unterstützen.

„Ich sehe schon in der App, dass mein Programm funktioniert“

Die Programmierumgebung MakeCode bietet eine integrierte Simulation, die zeigt, wie der Calliope mini den geschriebenen Programmcode ausführen würde. Dies kann eine wertvolle Unterstützung beim ersten Programmieren sein: Grobe Fehler fallen frühzeitig auf, noch bevor das Programm auf den Mikrocontroller übertragen wird.

Allerdings besteht die Gefahr, dass sich die Schüler:innen allein auf diese Simulation verlassen und auf die Übertragung verzichten. Dadurch geht der praktische Umgang mit dem Calliope mini als programmierbarem Mikrocontroller verloren – ein wesentlicher Bestandteil der Unterrichtseinheit.

Empfehlungen: Ermutigen Sie die Kinder dazu, ihre fertigen Programmcodes aktiv auf den Calliope mini zu übertragen und dort zu testen. Manche Fehler zeigen sich erst auf der realen Hardware – sei es durch unvorhergesehene Reaktionen der Sensoren oder durch Verzögerungen in der Programmabfolge. Diese echten Erfahrungen sind ein wertvoller Impuls zur Fehlersuche (Debugging) und zur Weiterentwicklung der Programme. Indem die Schüler:innen den gesamten Programmierprozess durchlaufen – von der Idee über die Umsetzung bis zur Optimierung auf dem Mikrocontroller – entwickeln sie nicht nur technische Kompetenzen, sondern auch wichtige Problemlösestrategien.

3. Zweite Doppelstunde

3.1 Thema

Vertiefung des algorithmischen Problemlösens unter Nutzung des Ultraschallsensors des Calliope mini. Der Calliope mini wird im Laufe der Doppelstunde mit einem Ultraschallsensor verbunden und so programmiert, dass bei einem bestimmten Abstand ein Ton abgespielt wird.

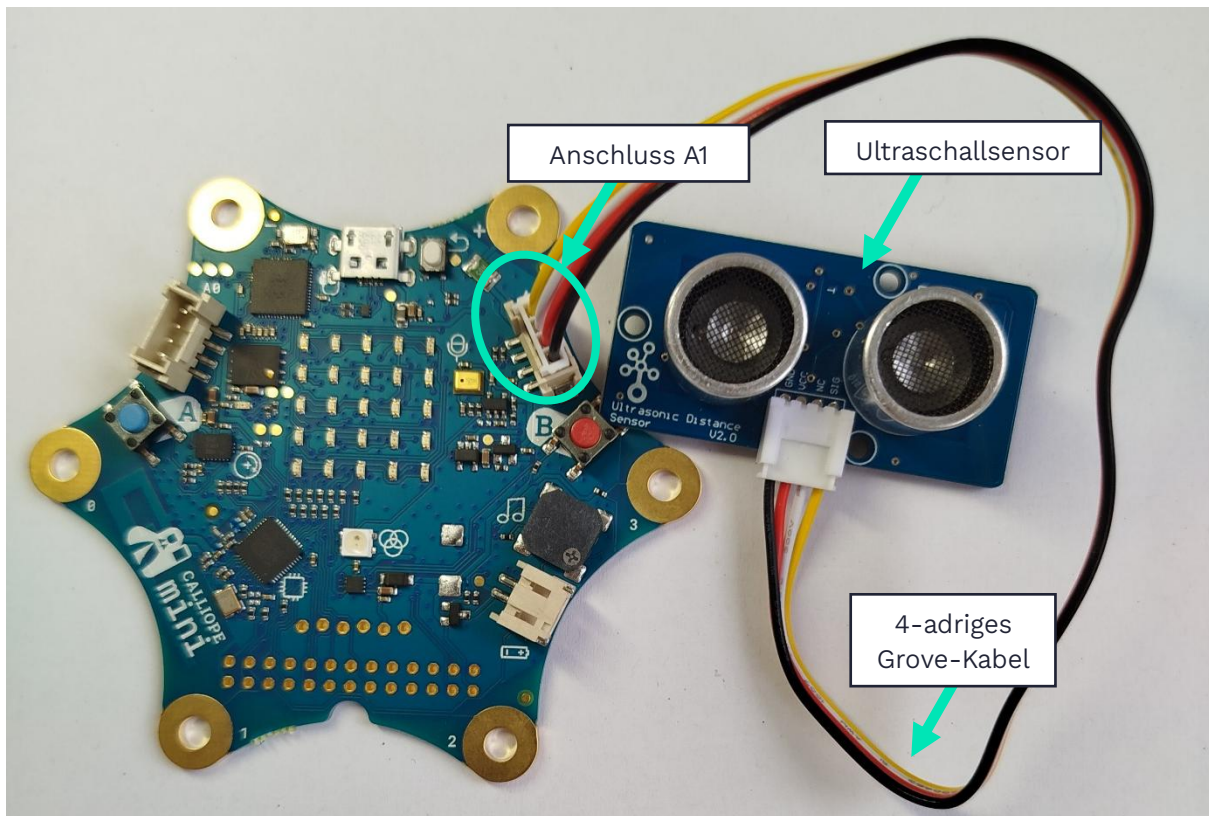


Abbildung 5 Der Calliope mini mit dem Ultraschallsensor (Bild: Autor:innen)

3.2 Benötigtes Material

- ein Calliope mini (hier angepasst an die Version 2.1) + Erweiterungsset mit Ultraschallsensor wie zum Beispiel „Sensorbox - 20 Grove Sensoren (10 Ultraschall- und 10 Feuchtigkeitssensoren)“ pro Person (siehe https://informatiksysteme.calliope.de/informatiksysteme/calliope/projekte/Ultraschallsensor_Auto/lernstrecke/Ultraschallsensor)
- digitale Endgeräte (Laptops, Tablets), bluetoothfähig oder mit USB-Anschluss
- installierte Calliope mini App oder Online-Aufruf des Editors von MakeCode (<https://makecode.calliope.cc/>)
- Installation der Grove-Erweiterung; <https://calliope.cc/calliope-mini/erweiterungen/sensoren/ultraschallsensor>
- Tafel/ digitale Alternative
- ausgedruckte oder digital zur Verfügung gestellte Aufgaben (siehe 3.6)

3.3 Kompetenzen

Die Schüler:innen

- führen gegebene Programme nach Anleitung aus, indem vorhandene Calliope-mini-Programme mit dem entsprechenden Werkzeug geöffnet, auf den Calliope mini übertragen und ausgeführt werden (AFB I),
- entwerfen, realisieren und testen Algorithmen unter altersgerechter Anwendung der Funktionsweise des Ultraschallsensors, indem sie bei unterschiedlichen Abständen zu Hindernissen das Abspielen von Tönen variieren (AFB II).
- entwerfen, realisieren und testen Algorithmen mit den algorithmischen Grundbausteinen Anweisung, Sequenz, Wiederholung, bedingte Anweisung und Verzweigung, indem sie Programme zur Lösung einfacher Aufgaben/Probleme für den Calliope mini entwerfen, implementieren und testen (AFB II/III).
- programmieren ein Informatiksystem, indem sie Programme für den Calliope mini schreiben und testen (AFB II/III).
- stellen Algorithmen dar und vergleichen diese unter Verwendung einer altersgerechten, informatischen Fachsprache (AFB II/III).
- geben grundlegende, allgemeingültige Beschreibungen der Funktion und Arbeitsweise von Informatiksystemen an (EVA-Prinzip), indem sie die Programme für den Calliope mini bzgl. der Komponenten des EVA-Prinzips untersuchen und den entsprechenden Teilbereichen begründet zuordnen (AFB II/III).

3.4 Verlaufsplan zur zweiten Doppelstunde

U-Phase	SuS-/L-Aktivität (Grafiken: selbsterstellte Bilder bzw. selbsterstellte Screenshots der Programmierumgebung Microsoft MakeCode - © Microsoft/MIT-Lizenz)	Sozialform	Medien, Material
	Vor der Unterrichtsstunde: Verteilen der Tablets/Notebooks, Calliope minis und der Erweiterungssets mit Ultraschallsensor → 1 bis 2 pro 2er-Team, Geräte dürfen nicht ohne Erlaubnis der Lehrkraft ausgepackt und benutzt werden.		
Einstieg 15 Min.	- Begrüßung Spiel zur Orientierung im Raum und zum Abschätzen von Abständen über akustische Signale: - Fünf Schüler:innen (S1 bis S5) schließen die Augen, fünf Schüler:innen (S6 bis S10) verteilen sich nach und nach im Raum (Abstand und Richtung zu den fünf nicht sehenden S1 bis S5, die alle nah beieinanderstehen sollten, variieren). S6 sagt etwas, L bittet S1 bis S5 mit geschlossenen Augen auf die Stelle zu zeigen, an der sie S6 vermuten, jeder schätzt kurz den Abstand in Metern. S1 bis S5 öffnen die Augen und stellen fest, inwiefern sie richtig lagen. S6 setzt sich hin und es geht mit S7 weiter usw. Die übrigen sitzenden S können die Augen beim Platz ebenfalls schließen und mitmachen. - Praxisbezug: Ermittlung, wie weit ein Gewitter entfernt ist: Zählt man die Sekunden zwischen dem Blitz und dem Donner, z. B. 6 Sekunden, dann muss man diese Zahl, hier 6, durch 3 teilen, Ergebnis: 2. Schätzung: Der Blitz hat in etwa 2 Kilometern Entfernung eingeschlagen. - Ultraschall abgeben und wahrnehmen können Tiere wie Delfine und Fledermäuse. Ähnlich wie wir die Entfernung des Gewitters abschätzen können, sind die Tiere in der Lage, Entfernung, aber auch Richtung des Hindernisses oder der Beute zu „hören“. Ultraschall (für uns nicht mehr hörbares Tonsignal) wird von der Fledermaus ausgesendet, trifft auf einen Gegenstand und wird zurück reflektiert, Fledermaus nimmt reflektiertes Signal wahr und verarbeitet es zur Einschätzung des Abstands. Die halbe Laufzeit des Schalls ist mit der Schallgeschwindigkeit zu multiplizieren, ähnlich wie man die Entfernung eines Gewitters ableitet. Beim Aufwachen entwickeln Säugetiere so etwas wie ein „Ohrenmaß“ für ihr natürliches Sonar. Weitere Informationen dazu sind z. B. unter folgendem Link zu finden: https://www.weltderphysik.de/thema/hinter-den-dingen/ultraschall/ - Fachwort Ultraschall: Bei der Bezeichnung Ultraschall sind Schallwellen gemeint, die oberhalb des menschlichen Hörbereichs liegen und somit von Menschen nicht hörbar sind. - Überblick über Thema der Stunde geben: Abstände mithilfe eines Ultraschallsensors messen wie es zum Beispiel Delfine, Fledermäuse machen oder auch moderne Autos	S-S-Interaktion; Plenum	Bilder (siehe 3.5); Projektion via Beamer

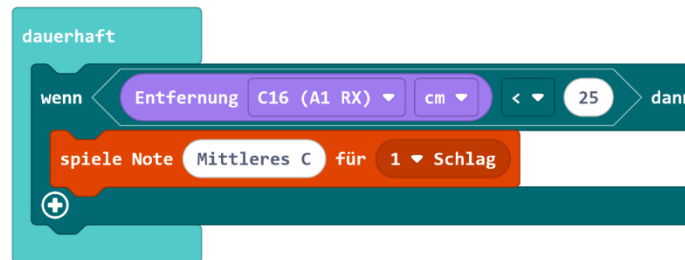
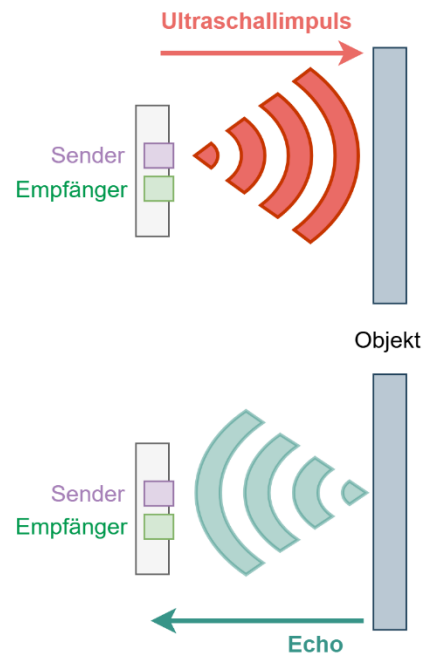


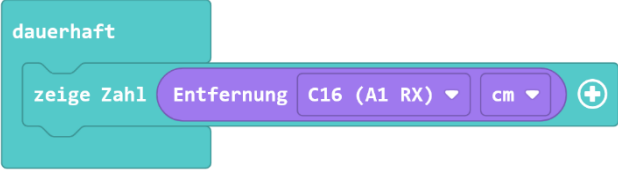
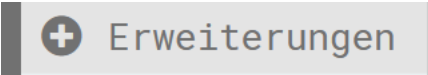
Bild: Autor:innen
Vereinfachte Darstellung des Ultraschallmessprinzips zur Abstandsmessung mithilfe des Ultraschallsensors (links) und das zugehörige Programm (rechts), erstellt mit dem MakeCode-Editor

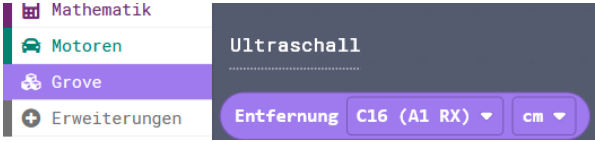
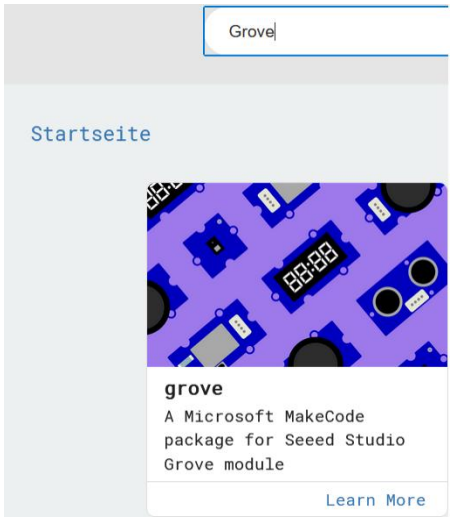
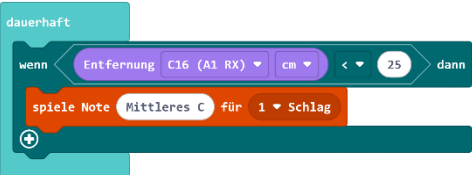
Erarbeitung 1
10 Min

- L zeigt S Calliope mini verbunden mit Ultraschallsensor (siehe Abb. 6).
- Aufgabe: Analysiere das hier angezeigte Programm und stelle Vermutungen an, was es macht.

Think-Pair-Share Methode

Calliope mini mit Ultraschallsensor, Projektion des Programmcodes via Beamer; Tablets/Notebooks mit MakeCode

	- Der Ultraschallsensor wird über den Grove-Anschluss A1 an den Calliope mini angeschlossen. Die Signale, die der Ultraschallsensor liefert, wird der violetten Variablen „Ultraschallsensor (in cm)“ zugewiesen. Die übertragenen Signale werden auf dem LED-Feld angezeigt z.B. 10. Das bedeutet, dass sich ein Hindernis in 10 cm Entfernung vom Ultraschallsensor befindet. - Weitere Informationen für die Lehrkraft: Über den Grove-Anschluss A1 können analoge Signale verarbeitet werden. Bei dem Calliope mini läuft das UART Protokoll (UART (Universal Asynchronous Receiver/Transmitter), eine serielle Kommunikationstechnologie) auf dem rechten Grove Port A1 mit den Pins C16 (RX) und C17 (TX). „RX“ steht einfach ausgedrückt für „Empfang“, „TX“ für „Übertragung“. Weiter Informationen dazu, sind unter https://docs.calliope.cc/tech/software/protocols/ zu finden. 		
Überleitung 15 Min.	- Tablets/Notebooks und Calliope minis werden gemeinsam für den Einsatz vorbereitet; S folgen den Schritten, die L vorne für alle durchführt (Einbindung der Grove-Erweiterung innerhalb der MakeCode-Umgebung zur Nutzung des Ultraschallsensors) - Schritte, die gezeigt werden: - Auf „Erweiterungen“ bei MakeCode klicken 	Plenum	Projektion via Beamer; Tablets/Notebooks mit MakeCode, Calliope mini

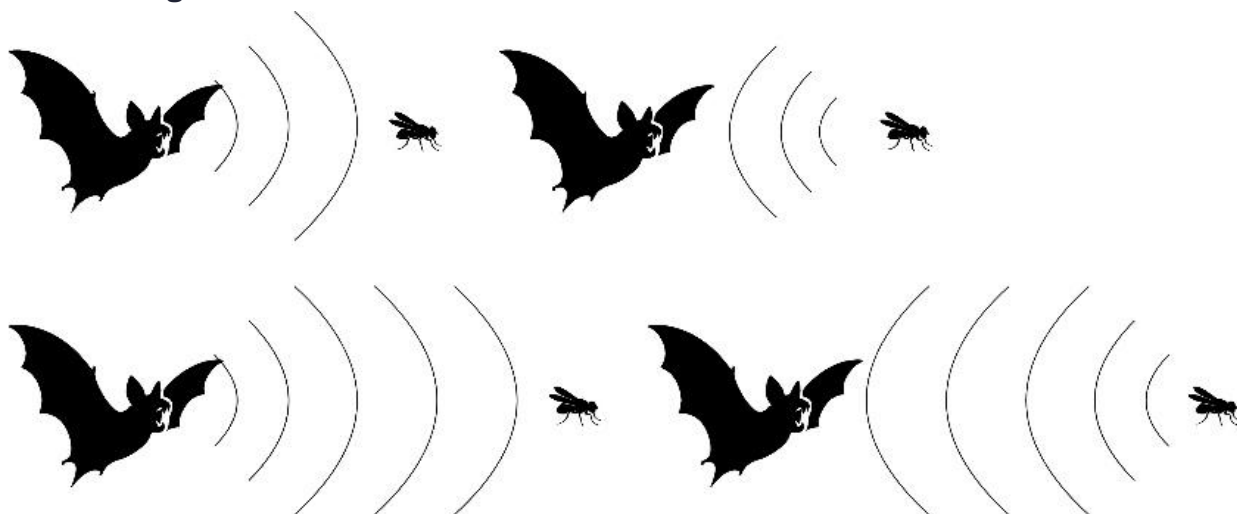
	- Im Suchfeld nach „Grove“ suchen. - Auf die „Grove“-Kachel klicken. - Die Erweiterung wird automatisch bei den zur Verfügung stehenden Befehlen hinzugefügt.  - S sollen selbstständig das Einstiegsprojekt verändern, testen und erneut verändern, so dass sie schließlich ein eigenes „Produkt“ entwickeln. - Anhand dieser ersten Aufgabe wird das EVA-Prinzip der letzten Doppelstunde wiederholt. - S sollen selbstständig das Programm zum Anzeigen des Abstandes programmieren und mit dem Calliope mini testen und eigenständig verändern und ergänzen. - Speicherung des Projekts durch die S. 		
Erarbeitung 2 20 Min.	- Bearbeitung der Aufgabe 2 (siehe 3.6) - Bearbeitung der Aufgaben 3 bis 4 (siehe 3.6, je nach individueller Leistungsfähigkeit)	PA	Calliope mini mit Ultraschallsensor, Tablet/Notebook, Musterlösungen zur Selbstkontrolle auf Pult ausgedruckt bereit halten
Sicherung 18 Min.	- Vorstellung der S-Lösungen von Aufgabe 2 - Wiederholung des EVA-Prinzips. - Je nach Lösungen werden die algorithmischen Grundbausteine Anweisung, Sequenz, Wiederholung, bedingte Anweisung und Verzweigung benutzt. In dem obigen Beispiel kann darauf eingegangen werden, dass es eine bedingte Anweisung (Begriff ggf. an die Tafel schreiben) verwendet wird. 	S-Aktivität, Plenum, L	Projektion via Beamer; Tablets/Notebooks mit MakeCode, Calliope mini EVA-Schilder (hängen an seitlicher Tafel, parallel zur Projektionsfläche, EVA-Karten)

	• Wenn die Bedingung „Wert der Variablen Ultraschallsensor < 25“ erfüllt ist, also wahr ist, dann wird die Anweisung „spiele Note Mittleres C für 1 Schlag“ ausgeführt. Ob diese Bedingung erfüllt ist, wird immer wieder also dauerhaft (durch eine Endlosschleife) überprüft. • if (wenn)-Anweisung: eine einfache Kontrollstruktur in der Informatik (siehe auch Kapitel 1.5)		
Austausch 10 min	• Basar-Methode: Pro 2er-Team bleibt eine Person am Platz und stellt ein Projekt vor (mit Calliope mini und Ultraschallsensor), jeweils andere Person bekommt ein Projekt vorgestellt, Wechsel von einer Person zur nächsten nach ca. 2 Min. • nach ca. Hälfte der Zeit: Rollenwechsel im 2er-Team	Basar	Calliope mini mit Ultraschallsensor
Weiteres für Schnelle	• Rückblick: Was haben wir alles am Calliope mini kennengelernt bzw. was kennen wir noch nicht: https://inf-schule.de/informatiksysteme/calliope/kennenlernen		
Aufräumen 2 Min.	• S räumen auf	S-Aktivität	

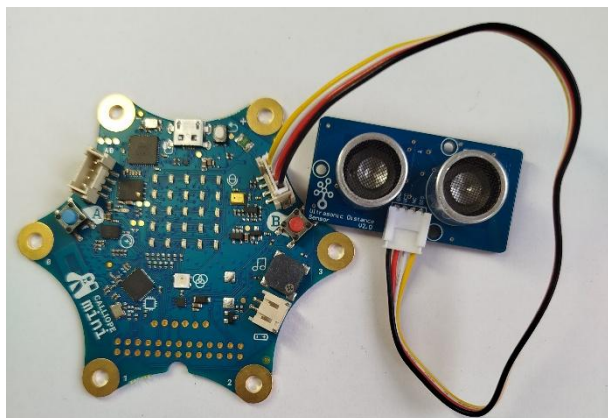
Abkürzungen: L: Lehrkraft, S: Schüler:innen, PA: Partnerarbeit, GA: Gruppenarbeit

3.5 Mögliche Tafelbilder

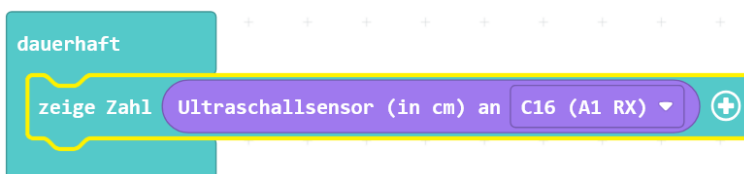
Zum Einstieg (Bild: Autor:innen)



Erarbeitungsphase

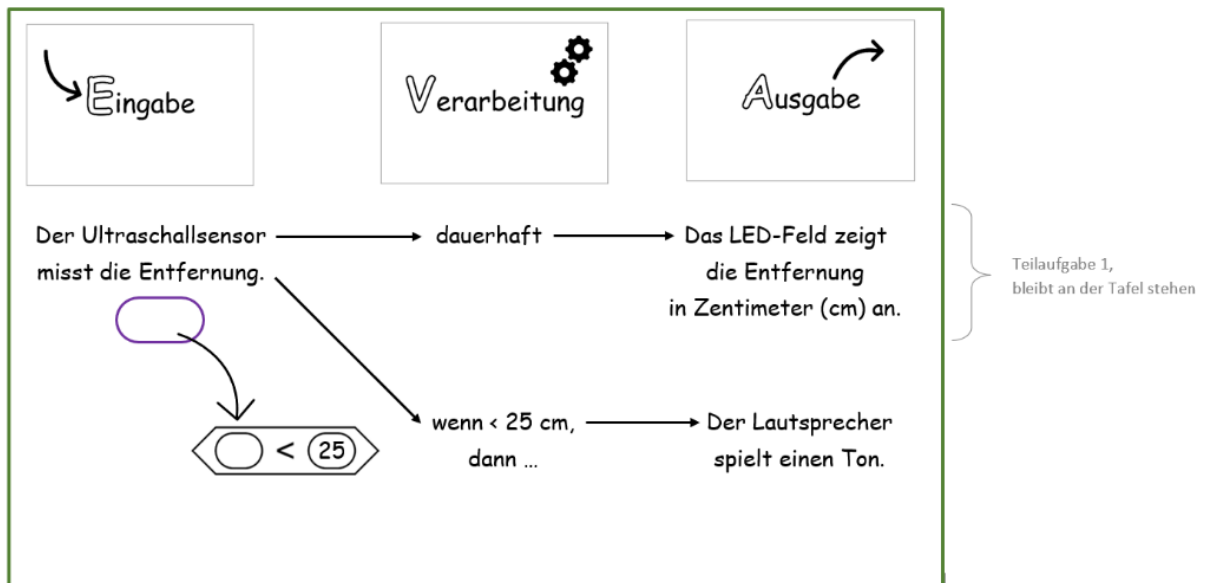


Tafelbild Teilaufgabe 2: Einen Abstand hören



Einen Abstand anzeigen

Tafelbild EVA-Prinzip



3.6 Aufgaben zur zweiten Doppelstunde

(Grafiken: selbsterstellte Screenshots der Programmierumgebung Microsoft MakeCode - © Microsoft/MIT-Lizenz)

Aufgabe 1: Mit dem Ultraschallsensor Abstände anzeigen

Der Ultraschallsensor soll immer wieder den Abstand (in cm) zu Hindernissen auf dem LED-Feld anzeigen. Die Anweisungen (siehe unten) helfen dir beim Programmieren.



Der Ultraschallsensor ermittelt die Entfernung zu Hindernissen und gibt das Signal an den Calliope mini weiter.

Die folgenden Anweisungen werden aufgrund des „dauerhaft-Blockes“ dauerhaft wiederholt:
Das Signal, das der Calliope mini vom Ultraschallsensor bekommt, wird der violetten Variablen „Ultraschallsensor (in cm)“ zugewiesen. Der Block „zeige Zahl“ stellt eine Anweisung an das LED-Feld dar, eine bestimmte Zahl anzuzeigen, hier die Entfernung.

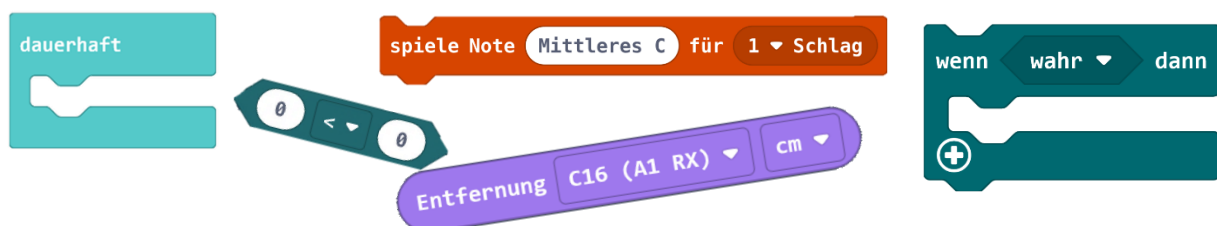
Das LED-Feld des Calliope mini zeigt die Entfernung in Zentimeter (cm) an.

Aufgabe 2: Einen „Abstand hören“

Der Ultraschallsensor soll immer wieder die Entfernung (in cm) zu Hindernissen messen. Wenn der Abstand zu dem Hindernis weniger als 25 cm beträgt, dann soll der Lautsprecher einen Ton abspielen.

Programmiere und teste danach dein Programm. Bei Bedarf verbessere dein Programm oder erweitere es nach deinen Bedürfnissen.

Die Anweisungen (siehe unten) helfen dir beim Programmieren.



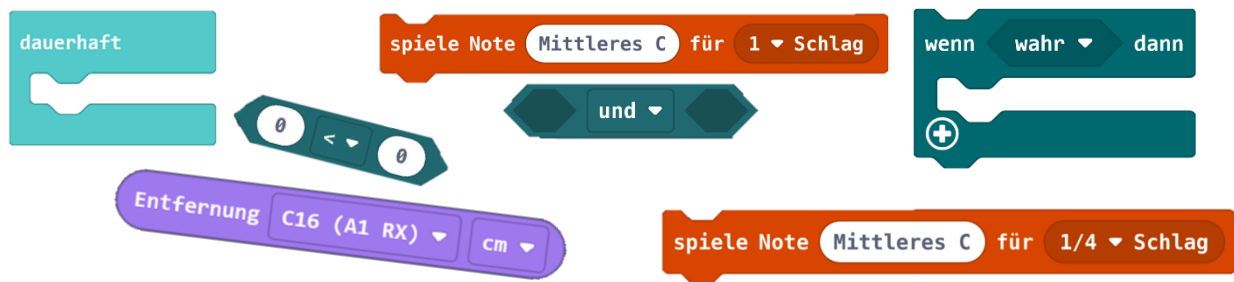
Aufgabe 3: Verschiedene „Abstände hören“

Der Ultraschallsensor soll immer wieder die Entfernung (in cm) zu Hindernissen messen. Wenn der Abstand zu dem Hindernis weniger als 25 cm beträgt, soll der Lautsprecher einen kurzen Ton abspielen.

Wenn der Abstand zu dem Hindernis größer gleich 25 cm und weniger als 50 cm beträgt, dann soll der Lautsprecher einen langen Ton abspielen.

Programmiere und teste danach dein Programm. Bei Bedarf verbessere dein Programm oder erweitere es nach deinen Bedürfnissen.

Die Anweisungen (siehe unten) helfen dir beim Programmieren.

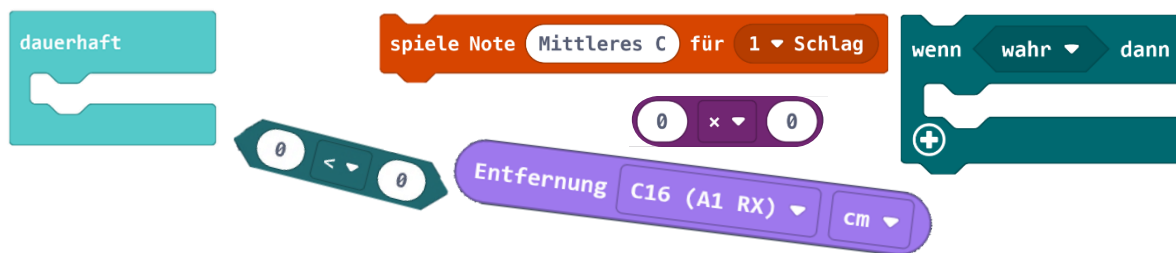


Aufgabe 4: Verschiedene „Abstände hören“

Der Ultraschallsensor soll immer wieder die Entfernung (in cm) zu Hindernissen messen. Wenn der Abstand zu dem Hindernis weniger als 50 cm beträgt, dann soll der Lautsprecher einen Ton abspielen. Aber: Je kleiner der Abstand wird, desto kürzer werden die Töne.

Programmiere und teste danach dein Programm. Bei Bedarf verbessere dein Programm oder erweitere es nach deinen Bedürfnissen.

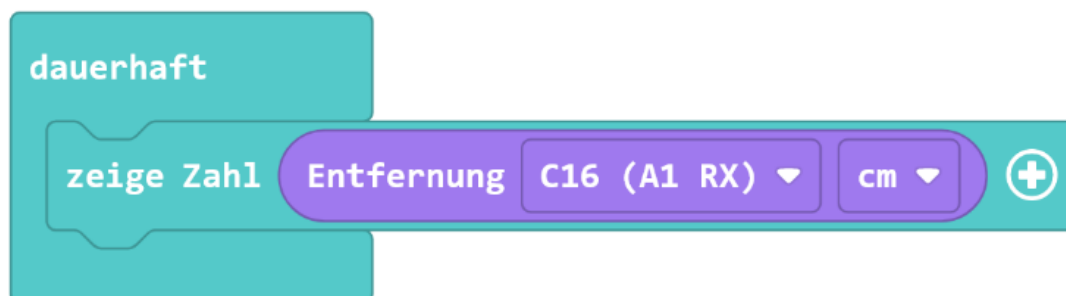
Die Anweisungen (siehe unten) helfen dir beim Programmieren.



3.7 Lösungen zu den Aufgaben der zweiten Doppelstunde

(Grafiken: selbsterstellte Screenshots der Programmierumgebung Microsoft MakeCode - © Microsoft/MIT-Lizenz)

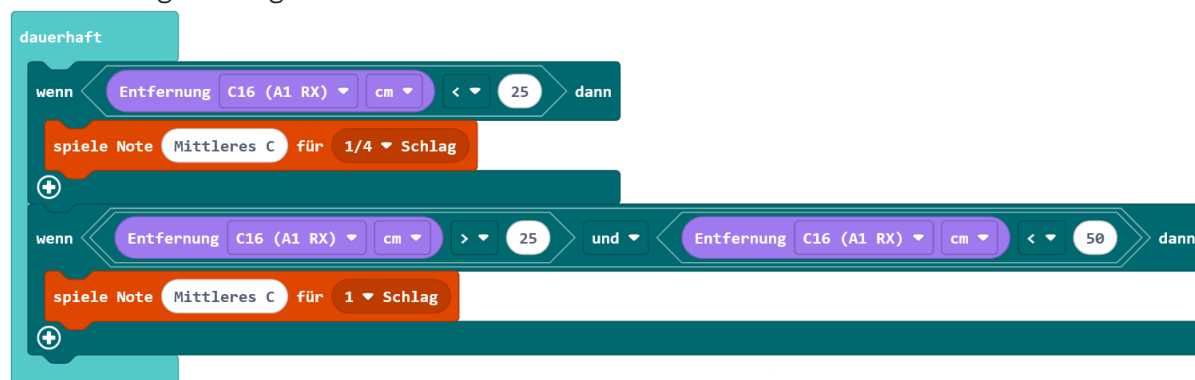
Lösung zu Aufgabe 1



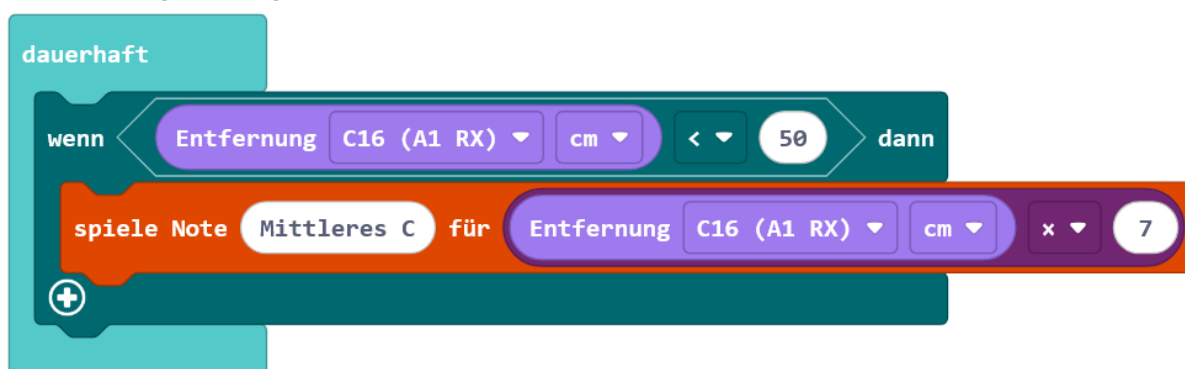
Eine Lösung zu Aufgabe 2



Eine Lösung zu Aufgabe 3



Eine Lösung zu Aufgabe 4



3.8 Die Funktionsweise eines Ultraschallsensors – Von der Natur zur Technik

Um das Prinzip eines Ultraschallsensors verständlich zu machen, wird zunächst ein Bezug zur Natur hergestellt. Die Schüler:innen lernen, dass Fledermäuse Ultraschall nutzen, um sich im Dunkeln zu orientieren: Sie senden hochfrequente Schallwellen aus, die von Objekten reflektiert werden. Aus der Zeitdifferenz zwischen dem Aussenden der Schallwellen und der Wahrnehmung des Echos berechnen Fledermäuse Entfernungen und navigieren sicher durch ihre Umgebung. Delphine und ein paar andere Tiere orientieren sich ebenso mit diesem Prinzip der Echolotation.

Dieses Prinzip findet sich auch in elektronischen Ultraschallsensoren wieder: Ein Sensor sendet Ultraschallwellen aus, die an Hindernissen reflektiert und anschließend wieder empfangen werden. Anhand der Zeitverzögerung wird die Entfernung zum Hindernis berechnet.

Um die Reflexion von Schallwellen anschaulich zu machen, können die Kinder ein einfaches Experiment durchführen: Sie halten ihre Hände oder ein Blatt Papier vor den Mund und sprechen. Dabei spüren und hören sie, wie sich der Klang ihrer Stimme verändert – ein Effekt, der ihnen hilft zu verstehen, wie Schallwellen auf Hindernisse treffen und reflektiert werden.

3.9 Programmieren mit dem Ultraschallsensor – Vom Messen zur akustischen Ausgabe

Nachdem die Funktionsweise des Ultraschallsensors geklärt wurde, setzen die Schüler:innen ihr Wissen praktisch in Partnerarbeit um. Ihr Ziel ist es, mit einem Ultraschallsensor Entfernungen zu messen und diese als Zahlenwerte auf dem LED-Display des Calliope mini auszugeben. Die Ultraschallsensoren unterliegen unterschiedlichen Limitierungen, die zu ungenauen Messergebnissen führen können. Auch diesen Erkenntnissen gilt es Aufmerksamkeit zu schenken und durch die Aufgaben und Experimente zu erkunden. Der Ultraschallsensor benötigt für genaue Messungsergebnisse einen Mindestabstand von bis zu 5 cm.

Teilaufgabe 1: Abstände auf dem Calliope mini anzeigen

- Die Kinder analysieren im Plenum, welche Eingabe, Verarbeitung und Ausgabe für dieses Programm erforderlich sind. Um diese Annahmen dann in der Realität zu testen.
- Dazu muss im ersten Schritt die Grove-Erweiterung in MakeCode importiert werden, um den Ultraschallsensor in die Programmierumgebung mit einzubinden. Die dafür nötigen Arbeitsschritte werden mit den Kindern zusammen durchgegangen.
- Die Lehrkraft geht mit den Schüler:innen das erste Ultraschallsensor-Beispiel durch um direkt auf Rückfragen reagieren zu können.
- Die Schüler:innen testen eigenständig ihre Programm mit Hilfe des Calliope mini und finden heraus, dass sich unterschiedliche Abstände in Zentimetern auf dem LED-Display (Matrix) anzeigen lassen.

Teilaufgabe 2: Von der Anzeige zur akustischen Rückmeldung

- Ein spielerischer Einstieg macht die Bedeutung akustischer Signale bewusst: Die Kinder orientieren sich mit geschlossenen Augen im Raum und versuchen, Personen anhand ihrer Stimmen zu lokalisieren.
- Im anschließenden Unterrichtsgespräch wird die praktische Relevanz akustischer Rückmeldungen diskutiert – sei es für Sehbeeinträchtigte oder beim Einparken eines Autos mit Parksensoren.
- Die Kinder erweitern in Partnerarbeit ihr Programm: Sobald ein bestimmter Abstand unterschritten wird, gibt der Calliope mini ein Tonsignal aus.

Teilaufgabe 3 & 4: Verschiedene „Abstände hören“

- Die Lernenden steigern die Komplexität ihrer Programme schrittweise:
- **Teilaufgabe 3:** Der Calliope mini gibt Töne in zwei Stufen aus, abhängig vom gemessenen Abstand.
- **Teilaufgabe 4:** Die Dauer des Tons wird proportional zum Abstand programmiert.

Während dieser Aufgaben vertiefen die Schüler:innen ihr Verständnis für die benötigten logische und mathematische Verknüpfungen und setzen diese selbstständig in Code um. Ihre Programmierfähigkeiten werden so durch eigenständiges Tüfteln und spielerisches Testen weiterentwickelt – beispielsweise, indem sie sich mit geschlossenen Augen einer Wand nähern und durch ein Tonsignal rechtzeitig gewarnt werden.

3.10 Mögliche Stolperstellen: „Die angezeigten Zahlen ergeben doch gar keinen Sinn!“

Nachdem die Schüler:innen die Funktionsweise eines Ultraschallsensors kennengelernt haben, sollen sie in Partnerarbeit untersuchen, wie der Calliope mini die gemessenen Abstände auf seinem LED-Feld ausgibt. Dabei kann es jedoch zu Verständnisschwierigkeiten kommen:

- Da das LED-Feld des Calliope mini nur einzelne Ziffern anzeigen kann, müssen mehrstellige Zahlen als laufende Anzeige von rechts nach links dargestellt werden. Dies kann für einige Kinder ungewohnt sein und zu Verwirrung führen, besonders im Vergleich zu statisch angezeigten Zahlen.
- Manche Schüler:innen fokussieren sich eher auf die Zahlenanzeige als auf die Sensorfunktion. Wenn der Ultraschallsensor nicht bewusst beachtet wird, erscheinen die ausgegebenen Zahlen für einige Kinder zufällig, da sie nicht sofort den Bezug zu den gemessenen Abständen herstellen.
- Einige Kinder versuchen, Muster in den Zahlen zu erkennen, indem sie sie mit bekannten Zahlenfolgen aus dem Mathematikunterricht in Verbindung bringen, anstatt den Zusammenhang zwischen Abstand und Zahlenwert zu erkennen.

Tipps zur Unterstützung

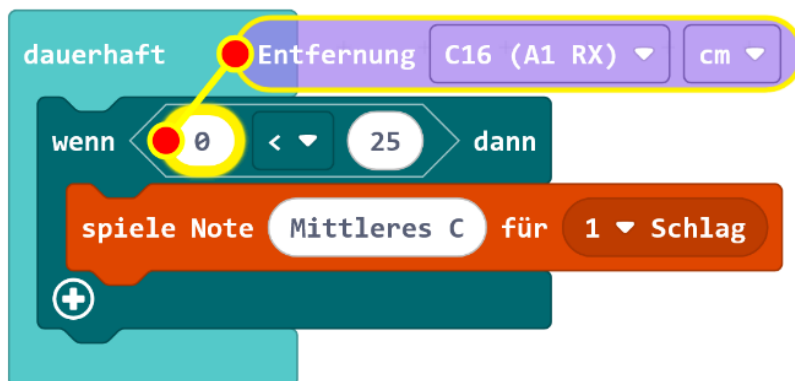
- Verständnisschwierigkeiten gezielt aufgreifen: Falls Kinder Schwierigkeiten mit der Darstellung mehrstelliger Zahlen haben, sollte dies präventiv in der Einführung thematisiert werden. Eine kurze Demonstration kann helfen, das Prinzip des durchlaufenden Zahlendisplays zu verstehen.
- Die Kinder zur aktiven Erkundung ermutigen: Erinnern Sie die Schüler:innen daran, dass der Ultraschallsensor Abstände misst. Fordern Sie sie auf, bewusst verschiedene Gegenstände in unterschiedlichen Entfernungen vor den Sensor zu halten. Durch dieses systematische Variieren können sie selbst entdecken, dass die ausgegebenen Zahlen tatsächlich den Abständen in Zentimetern entsprechen.

„Verstehen, dass der Ultraschallsensor Zahlenwerte liefert“

Der Ultraschallsensor misst Abstände und gibt diese als Zahlenwerte an den Mikrocontroller weiter. Einige Kinder haben Schwierigkeiten nachzuvollziehen, dass der Variablen „Ultraschallsensor (in cm)“ tatsächlich eine sich ständig aktualisierende Zahl zugewiesen wird.

- Wird der Block in eine „dauerhaft“-Schleife eingebaut, entspricht der Wert der Variablen „Ultraschallsensor (in cm)“ zu jedem Zeitpunkt dem aktuell gemessenen Abstand.
- Dadurch kann dieser Block sowohl direkt zur Anzeige von Zahlenwerten (vgl. Teilaufgabe 1) als auch für mathematische Vergleiche genutzt werden (vgl. Teilaufgaben 2-4).

- Diese Erkenntnis ist entscheidend, damit die Schüler:innen verstehen, dass der Ultraschallsensor nicht nur „irgendwie funktioniert“, sondern dass seine Messwerte konkreten Zahlen entsprechen, mit denen gerechnet werden kann.



(Grafiken: selbsterstellter Screenshot der Programmierungsumgebung Microsoft MakeCode - © Microsoft/MIT-Lizenz)

3.11 Zusammenführung der Erkenntnisse und Ausblick auf weitere Anwendungen

Zum Abschluss der Unterrichtseinheit reflektieren die Schüler:innen im Plenum über ihre Erfahrungen. Sie berichten von ihren Erfolgserlebnissen und den Herausforderungen, die sie während der Programmieraufgaben gemeistert haben. Der gemeinsame Austausch ermöglicht es, Lösungsansätze zu teilen und voneinander zu lernen.

Ein Ausblick auf die nächste Anwendung verdeutlicht die Übertragbarkeit des Gelernten: Der Feuchtigkeitssensor, der sich in manchen Erweiterungsset zusammen mit dem Ultraschallsensor befindet, bietet eine neue Möglichkeit zur Datenerfassung. Die Kinder diskutieren, wie sich das EVA-Prinzip auch hier anwenden lässt.

Ein Beispiel: Misst der Feuchtigkeitssensor in der Erde einer Zimmerpflanze einen Wert unterhalb eines festgelegten Wertes, erscheint auf dem LED-Display des Calliope mini ein trauriger Smiley – ein Hinweis darauf, dass die Pflanze gegossen werden muss.

Dieser Ausblick zeigt den Schüler:innen, dass sie ihre erworbenen Programmierfähigkeiten flexibel auf neue Problemstellungen übertragen können. Gleichzeitig wird ihr Interesse an weiteren kreativen Projekten mit dem Calliope mini geweckt.

Ihre Neugierde an weiteren Programmierprojekten mit Calliope mini wurde geweckt? Auf der Website von Calliope findet sich viel weiteres spannendes (Unterrichts-)Material:

<https://calliope.cc/schulen/unterrichtsmaterial>



Erschienen im

Kompetenzverbund lernen:digital
Marlene-Dietrich-Allee 16, 14482 Potsdam
Tel: 0331-977-256362
E-Mail: geschaeftsstelle@lernen.digital

Projektverbund

lernen:digital Kompetenzzentrum MINT
MINT-ProNeD

Datum der Erstveröffentlichung

16.09.2025

Autor:innen

Prof. Dr. Claudia Hildebrandt,
Matthias Matzner, Philipp Jung,
Jonas Braun, Larissa Nebelung,
Andreas Eller

Zitierhinweis

Hildebrandt, C., Matzner, M., Jung, P.,
Braun, J., Nebelung, L., Eller, A. (2025).
Mit dem Calliope mini Informatik entdecken – Einführung in das algorithmische Problemlösen. <https://www.hse-heidelberg.de/informatik-entdecken-calliope>

Die vorliegende Veröffentlichung ist im Rahmen des Projektverbunds MINT-ProNeD für das Kompetenzzentrum MINT im Kompetenzverbund lernen:digital entstanden.

Finanziert durch die Europäische Union – NextGenerationEU und gefördert durch das Bundesministerium für Bildung, Familie, Senioren, Frauen und Jugend (BMBFSFJ). Die geäußerten Ansichten und Meinungen sind ausschließlich die des Autors/der Autorin und spiegeln nicht unbedingt die Ansichten der Europäischen Union, Europäischen Kommission oder des Bundesministeriums für Bildung, Familie, Senioren, Frauen und Jugend wider. Weder Europäische Union, Europäische Kommission noch das Bundesministerium für Bildung, Familie, Senioren, Frauen und Jugend können für sie verantwortlich gemacht werden.



Dieses Produkt ist unter der Lizenz CC BY-NC-SA 4.0 veröffentlicht. Ausgenommene Inhalte sind an den einzelnen Inhalten angegeben. Die Urheber:innen sollen bei der Weiterverwendung wie folgt angegeben werden: Hildebrandt, C., Matzner, M., Jung, P., Braun, J., Nebelung, L., Eller, A., Kompetenzverbund lernen:digital, entstanden im Projektverbund MINT-ProNeD.

Dieses Dokument enthält Bilder, die aus Screenshots der Programmierumgebung Microsoft MakeCode erstellt wurden. Microsoft MakeCode unterliegt der MIT-Lizenz und erstellte Bilder & Inhalte dieser OER stehen unter der CC BY-NC 4.0 Lizenz. Eine kommerzielle Nutzung ist somit nur mit ausdrücklicher Genehmigung der Urheber gestattet. Alle im weiteren Dokument enthaltenen Abbildungen wurden von der Autorin/dem Autor, wenn nicht anders angegeben, selbst erstellt und stehen – wie das gesamte Dokument – unter der Creative Commons Lizenz CC BY-NC-SA 4.0 (Namensnennung – Nicht kommerziell – Weitergabe unter gleichen Bedingungen). Lizenztext: <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.de>