

Archiv der 2020 ausgemusterten Internetseiten zum Thema Calliope mini

Calliope

- Calliope mini
- Coden und Lernen
- Offline Coding
- Open Roberta Lab
- Makecode-Editor
- Calliope-Pins

Mobil mit Calliope

- Fernsteuerung
- Servolenkung
- Kollisionsschutz
- Linienfolger

Mobil mit Arduino

Die Übersicht ist zu den einzelnen Kapiteln hin verlinkt.

Ein Link hierher zurück findet sich am Ende eines jeden Kapitela

Calliope

Wenn man der Professorin und ehemaligen Internetbotschafterin der Bundesregierung, Gesche Joost, Glauben schenkt, gehören neben neuen Formen des digitalen Recherchierens und der Mediengestaltung auch das Programmieren zum Rüstzeug, das Grundschulkindern vermittelt werden sollte.



Die „Medienerziehung digitaler Art“ ist schon seit geraumer Zeit bedeutsam für das Bildungsgeschehen an den Grundschulen in Schleswig-Holstein. Seit dem 28. Juni 2018 sind nun die konkreten Inhalte in den „Fachanforderungen“ und der dazugehörigen Ergänzungsschrift „Medienkompetenz – Lernen mit digitalen Medien“ verbindlich festgelegt. So dürften in aller Regel Grundschüler im Umgang mit Computern geschult und mit den dazu nötigen Hardware-Kenntnissen vertraut gemacht werden. Sie lernen, was ein Betriebssystem ist, können mit Bild- und Textverarbeitungen umgehen, recherchieren im Internet, sprechen über Gefahren, die dort lauern und anderes mehr.

Dabei sollte es aber nicht bleiben. Vielmehr muss zu der Fähigkeit, digitale Medien „kritisch“ zu nutzen, noch hinzukommen, sie auch „technisch“ zu verstehen. Gemeint ist damit, dass zum Verständnis der digitalen Medien „informatische“ Grundkenntnisse nötig sind, wie sie beispielsweise beim Programmieren zur Anwendung kommen.

Über ein derart spezifisches Wissen verfügen Lehrkräfte an Grundschulen nur in Ausnahmefällen, sodass sich hier ein erheblicher Fortbildungsbedarf zeigt.

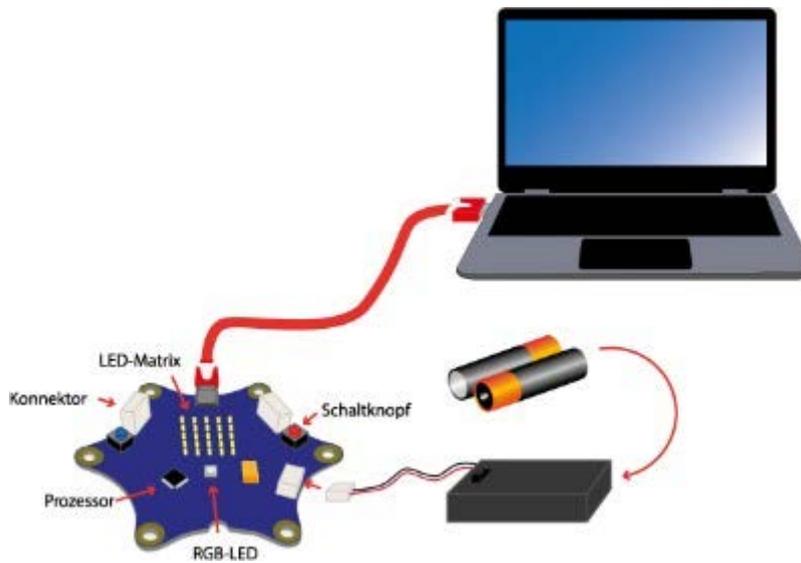
Hinzu kommt, dass die zum Programmieren erforderlichen Anwendungen oft nur über das Internet verfügbar sind, sodass eine entsprechende Ausstattung der Schulen nötig erscheint.

Dass beides - Qualifizierung der Lehrkräfte und schnelles Internet für alle Schulen - eine Mammutaufgabe darstellt, ist auch der Bildungsadministration klar. Im Jahr 2016 hat sie darauf mit einer „Bildungsoffensive für die digitale Wissensgesellschaft“ reagiert.

Es wird aber wohl noch einige Zeit dauern, bis diese Pläne realisiert sind. Bis dahin werden sich die Schulen auf ihre eigenen Möglichkeiten und die bestehenden Angebote besinnen müssen.



Calliope mini



Der Calliope mini kann bei verschiedenen Anbietern im Internet für etwa 35 € erworben werden. Er wird mit einem USB-Kabel, zwei AAA-Mignonzellen und dazu passendem Batteriehalter ausgeliefert. Im Unterschied zu anderen Mikrocontrollern wie beispielsweise dem Arduino sind neben dem Prozessor weitere Sensoren und Aktoren auf der Platine verbaut. Am auffälligsten neben den Konnektoren sind eine Matrix aus 25 LEDs, eine RGB-LED und zwei Schaltknöpfe.

Verbindet man den Calliope über

das USB-Kabel mit einem Computer, wird man von einer Laufschrift und blinken Pfeilen auf der LED-Matrix begrüßt, die dazu auffordern, die Schaltknöpfe zu betätigen.

Von einem Windows-Rechner wird der Calliope mini wie ein normaler USB-Stick erkannt und im Explorer als Laufwerk mit der Bezeichnung „MINI“ angezeigt.

Der Calliope mini hebt sich durch seine Ausstattung mit Sensoren und Aktoren von vielen anderen Experimentalplatinen ab. Leitend für diese Umsetzung ist wohl der Gedanke, mit geringem Materialaufwand viele Programmiergelegenheiten zu schaffen und damit Schulkindern am Anfang ihrer Bildungslaufbahn erfahrbar zu machen, dass sie in der digitalen Welt Gestalterinnen und Gestalter sein können und nicht nur Konsumentinnen oder Konsumenten.

Ob allerdings die auf der Calliope-Homepage vorgehaltenen Projekte diesem hehren Ziel dienen, darf bezweifelt werden. Oft ist dort das Programmieren reiner Selbstzweck und dient weder der Optimierung von Dingen noch von Prozessen.

Beispielsweise dürfte ein [digitaler Eierlauf](#) für Kinder enttäuschend sein, wenn sich nach dem Programmieren herausstellt, dass das Bastelobjekt für einen Eierlauf weniger taugt als das Ei selbst. Auch werden sich Kinder nicht ernstgenommen fühlen, wenn sie durch Programmierung eines Mikrocontrollers den [Füllstand](#) eines Messbechers ermitteln sollen und weder die Messwerkzeuge noch die Anzeige annähernd die Genauigkeit erreichen, die durch Augenmaß oder mit dem Eintauchen eines Zentimetermaßes zu erzielen wäre. Und so überzeugt es auch nicht, wenn zur [Temperaturmessung](#) einer Flüssigkeit statt eines einfachen Thermometers ein Calliope programmiert und eine Powerbank samt Thermofühler herangezogen werden sollen.

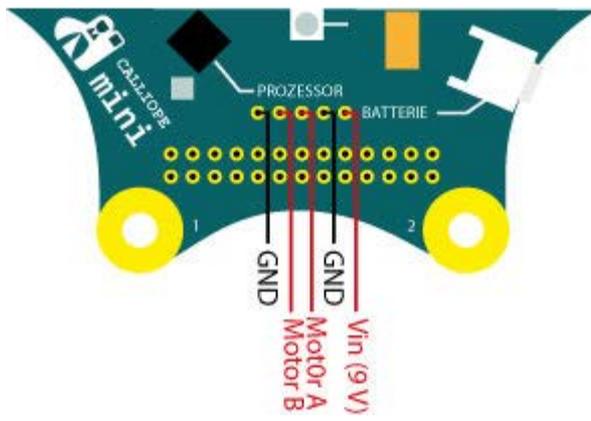
Die freistehenden Angebote sollten also mit Bedacht genutzt werden. Ein Schultauglichkeitsnachweis steht noch aus.

Vorsicht:

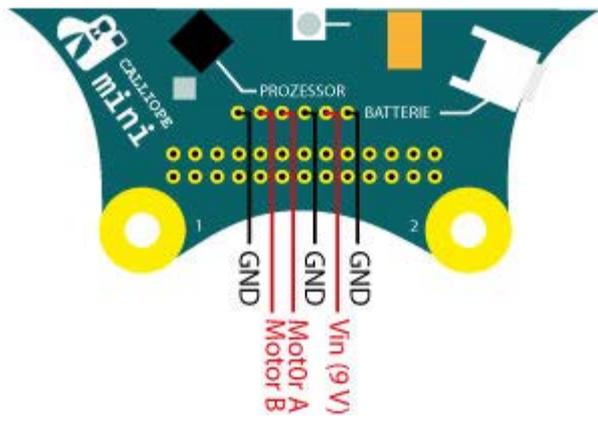
Es sind unterschiedliche Versionen des Calliope mini im Umlauf. Beim Anschließen externer Hardware, beispielsweise von Motoren, muss auf die unterschiedliche Pinbelegung geachtet werden. Die zurzeit neueste Version ist an den Lötösen zu erkennen. In der obersten Reihe sind es nun sechs statt wie bisher 5. Hinzugekommen ist ein GND-Pin.

Zu empfehlen ist, das jeweils neueste auf der Calliope-Homepage vorgehaltene Pinout mit dem eigenen Board abzugleichen.

ältere Version



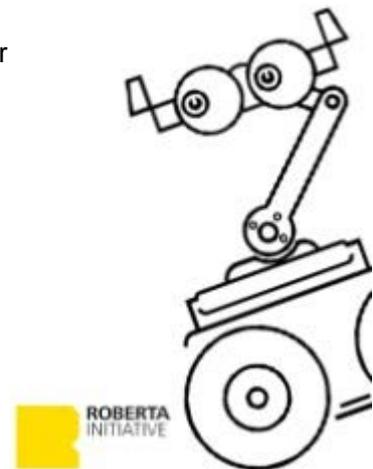
neuere Version



Coden und Lernen

In Deutschland geht die Angst um, das Land verbaue sich die Zukunft, wenn nicht schon Kleinkinder digital lernten. Andere Länder machen es vor: In England ist „Computing“ ein Pflichtfach für Erstklässler, in Estland und Luxemburg ebenso. Und auch hierzulande fluten Projektideen und Unterrichtsvorschläge das Internet. Leider wagt kaum jemand dem Code-Dogma zu widersprechen und es gibt auch keine Instanz, die sich damit beschäftigt, Spreu vom Weizen zu trennen. So sind interessierte Lehrkräfte nicht selten mit Vorschlägen konfrontiert, die meist nur wenig gemein haben mit gutem Unterricht.

Im Rahmen der „Roberta Initiative“ wird beispielsweise die Unterrichtseinheit **"Sachunterrichtsstunde: Belebte Natur"** vorgehalten, die zeigen soll, wie Sach- und informatisches Wissen sinnvoll miteinander verknüpft werden können.



Wer sich selbst ein Bild machen möchte, kann die Einheit [hier](#) oder [hier](#) herunterladen.

Wer darüber hinaus interessiert daran ist, wo da der "Hase im Pfeffer" liegt, sollte die nachfolgende Gegenüberstellung lesen:

Unterrichtsplanung

Thema

Verhaltensregeln im Umgang mit Pflanzen, Pflege einer Pflanze im Klassenraum

Material (für je 2 Kinder)

- Notebooks oder PCs mit Internetanschluss (<https://lab.open-roberta.org>)
- Calliope mini mit Batterie-Pack und USB-Kabel

Lehrerzählung als möglicher Einstieg

„[...] Mila ist von Beruf Gärtnerin [...] Sie hat zwei große Säcke mit Samen: Sonnenblumen und Primeln [...] aber sie weiß nicht genau, was die Pflanzen brauchen [...] Immer, wenn sie einen Samen einpflanzt, geht die Pflanze ein. Deshalb bittet sie dich um Hilfe.“

Problemstellung als Lernanlass

Die Kinder wissen nicht, was die Pflanzen brauchen. Deshalb ist es notwendig, dass sie sich informieren, um Mila helfen zu können.

Kommentierung

„Verhaltensregeln“ sind für die Aussaat und Aufzucht von Pflanzen so bedeutungslos wie PC und Mikrocontroller. Keimversuche und die Aufzucht von Pflanzen sollen dem Interesse für das Lebendige dienen. PCs, Internet und Mikrocontroller werden dazu nicht benötigt, sondern Anzuchttopfe, Blumenerde und ein Gießgeschirr.

Kinder sollten nicht für dumm verkauft werden. Sie haben ein natürliches Gespür dafür oder auch Kenntnis darüber, dass Gärtnerinnen genau wissen, "was die Pflanzen brauchen". Sie durchschauen auch das Hirngespinnst, sie könnte bei Fragen eine Grundschulklasse um Beistand bitten. Peinlich ist, dass die Lehrerin erzählt, Samen würden gepflanzt. Wie sich später zeigt, ist das nicht nur ein Versprecher.

Hat diese "Problemstellung" einen besonderen Aufforderungscharakter? Wohl kaum. Etwas nicht zu wissen ist in dieser Alterstufe allgegenwärtig und Wissbegierde kann nicht "einfach so" vorausgesetzt werden.

Darüber hinaus ist der Anspruch, die Kinder sollen "sich informieren" abwegig. Zum einen sind sie des Lesens noch nicht oder nur

Formulierung des Stundenthemas

„Heute hilfst Du Mila dabei herauszufinden, um welche Pflanzen es sich handelt und was sie zum Wachsen brauchen. Dabei hilft dir auch der Calliope mini.“

Arbeitsbogen: So geht es meiner Pflanze

Sieh jeden Tag nach deiner Pflanze.

Lies den Temperatur- und Lichtwert ab.

Trage beide Werte in die Tabelle ein.



Datum	Temperatur	Licht	So geht es meiner Pflanze
Beispiel: 01.01.18	23°C	80%	

ansatzweise mächtig (Klasse 1-4), zum anderen ist es viel zu schwierig klarzustellen, "was Pflanzen brauchen" (schwammige Problemstellung).

Die Frage, warum Milas Pflanzen eingehen (siehe Einstieg) wird aus unerfindlichen Gründen fallengelassen. Statt um Milas Problem soll es nun um Pflanzenbestimmung und darum gehen, was Pflanzen "zum Wachsen brauchen".

Soll die Keimesentwicklung (ein für die Grundschule geeignetes Thema) außer Acht gelassen werden?

Die Vermutung liegt nahe, weil "Lichtwerte" gemessen werden sollen. Bei der Keimesentwicklung macht das keinen Sinn, weil dazu Licht entbehrlich ist.

Beim Wachsen dagegen ist Licht bedeutsam: Es wird zur Bildung von Chlorophyll benötigt, das die Pflanze befähigt, aus Kohlendioxid und Wasser neue Stoffe zu bilden, die zur Größenzunahme (Wachstum) nötig sind.

Dieser Sachverhalt erscheint aber nur wenig tauglich für die Alterstufe.

Jeden Tag nach einer Pflanze zu sehen (siehe Arbeitsauftrag) beinhaltet nicht, ihren Zustand und mögliche Veränderungen zu dokumentieren. Dass das gewollt ist, wird nur deutlich, weil neben den Temperatur- und Lichtwerten auch ein Emoticon die Beispieltabelle zierte.

Zu erwarten steht, dass nur ein Smiley gebraucht wird, weil die Keim- und Anzuchtversuche unter den für einen Klassenraum realistischen und in der Beispieltabelle bereits genannten Werten von 23° C und 80% Licht gut gelingen und schädliche Auswirkungen ausbleiben sollten.

Die Bekanntgabe des Lichtwerts dürfte für die Kinder Hinweis genug sein, die Anzuchttöpfe nicht der prallen Sonne auszusetzen, was sie sowohl vor übermäßigem Licht als auch vor Hitze schützt. Mit Kälte muss in einem Klasserraum wohl kaum gerechnet werden.

Dass unter solch stabilen und im Detail abschätzbaren Umständen jede Lerngruppe für sich immer wieder Temperatur- und Lichtwerte ermitteln soll, ist schlichtweg unverstündlich und entlarvt ein solches Tun

als reinen Selbstzweck. Nicht die belebte Natur, sondern der Calliope mini steht im Mittelpunkt des unterrichtlichen Geschehens.

Es wäre bitter, wenn die Vermittlung digitaler bzw. medialer Kompetenzen zu Qualitätsverlusten führte im Hinblick auf curriculare Vorgaben und pädagogische Grundsätze.



Offline Coding

Die [Mission](#) des Calliope-Entwicklerteams, „jedem Schulkind ab der 3. Klasse einen spielerischen Zugang zur digitalen Welt zu ermöglichen“ trifft auf die unerfreuliche Realität, dass vielen Schulen über keine, zu langsame oder nicht störungssichere Internetanbindung verfügen.

Ist den Entwicklerinnen und Entwicklern dieses Problem verborgen geblieben oder ist es ihnen nicht bedeutsam genug? Auf der [offiziellen Homepage](#) wird es jedenfalls nicht thematisiert, obwohl Lösungen verfügbar sind.

Hier soll diesem Manko nun begegnet werden. Wer es sich zutraut und die Mehrarbeit nicht scheut, findet im folgenden gangbare Wege, um auch ohne Zugriff auf das Internet die benötigten Editoren zur Programmierung des Calliope mini vorhalten zu können.

Lösung 1:

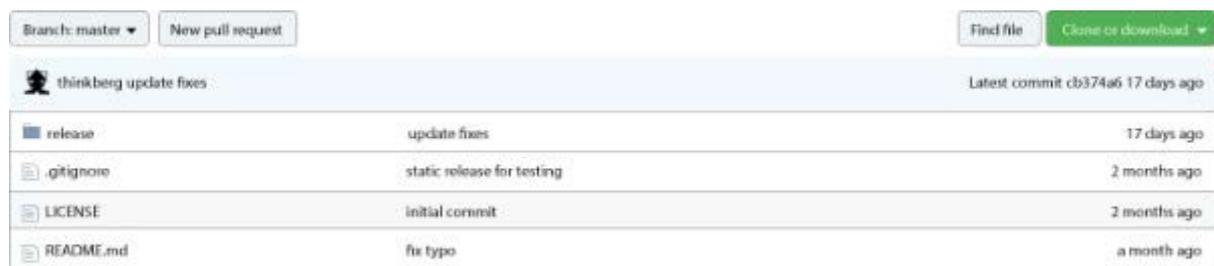
Man braucht einen WLAN-Router und einen Raspberry Pi. Wird auf dem Pi ein vom Fraunhofer-Institut zur Verfügung gestelltes Betriebssystem installiert, erhält man Zugriff auf eine lokal ausführbare Version des Open Roberta Labs.

Beim Hochfahren des Raspberry Pis wird der Server automatisch gestartet. Durch Eingabe von Benutzerkennung und Passwort kann er dann mit dem WLAN-Router verbunden und für andere Rechner innerhalb des WLAN-Netzes verfügbar gemacht werden. Eine detaillierte Anleitung zum Einrichten eines Raspberry Pis findet sich [hier](#). Spezielles für den Zugriff auf das Open Roberta Lab von Fremdrechnern aus kann in folgendem PDF-Dokument nachgelesen werden: [Download](#).

Lösung 2:

Man lädt sich die Programmdateien des PXT-Editors aus dem Internet herunter und führt sie lokal auf dem eigenen Rechner aus. Zu finden ist das Paket auf dieser [Internetseite](#).

Zum Herunterladen auf das grüne Feld "Clone or download" und in dem sich öffnenden Dialog auf "Download ZIP" klicken.

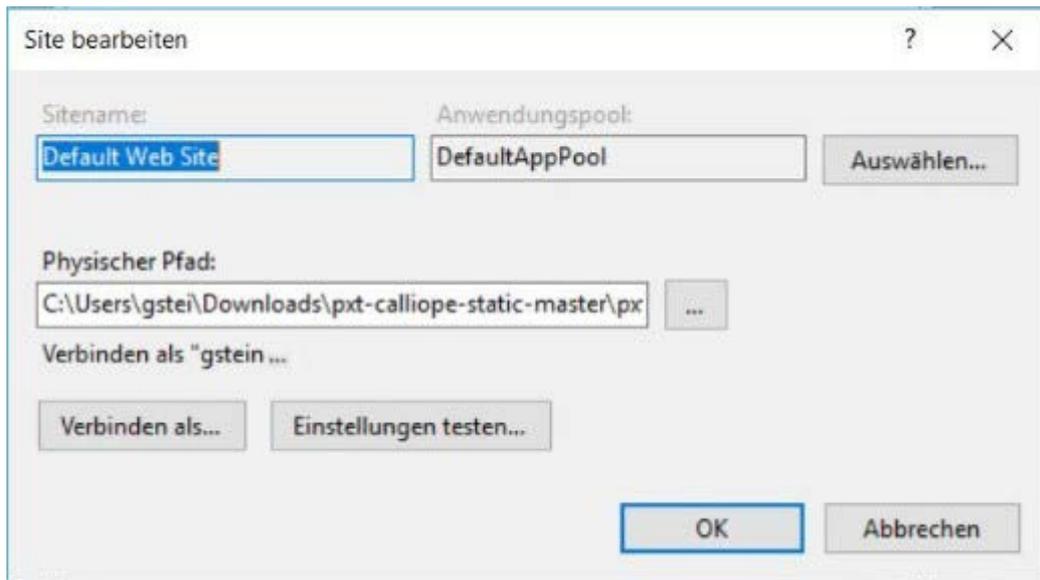


Damit wird die gepackte Datei "pxt-calliope-static-master" in den Ordner "Downloads" kopiert, aus dem sie danach an einen Ort der eigenen Wahl entpackt werden kann. Die zur Ausführung des Editors benötigten Dateien befinden sich in dem Ordner mit der Bezeichnung "release".

Um sie zur Anzeige zu bringen, können die "Internetinformationsdienste" des Betriebssystems benutzt werden, die unter Windows 10 wie folgt aktiviert werden:

1. Windows-Taste drücken und über das "A" oder den "#" zum Buchstabenmenü gehen. Das "W" führt zum Ordner "Windows-System", in dem sich der Zugang zur "Systemsteuerung" befindet. Nach dem Start über "Programme" ⇒ "Programme und Features" die "Windows-Features aktivieren ..." und in dem sich öffnenden Fenster "Internetinformationsdienste" anwählen.
2. Nach dem Klick auf die OK-Taste und erfolgreicher Installation die Windows-Taste erneut drücken und wie bei 1. über das "A" bzw. den "#" und das "W" zum Ordner "Windows-Verwaltungsprogramme" navigieren. Den Ordner öffnen und die "Computerverwaltung" starten.
3. In dem sich öffnenden Fenster über "Dienste und Anwendungen" die "Internetinformationsdienste ..." starten.
4. Es öffnet sich ein weiteres Fenster, in dem links oben unter "Verbindungen" auf die ">" geklickt werden muss, um zur "DefaultWebSite" zu gelangen. Bei der Anwahl wird sie blau hinterlegt. Im Fenster rechts unter "Aktionen" muss dann durch einen Klick auf "Grundeinstellungen" ein weiteres Fenster (siehe unten) geöffnet werden, über das der Pfad zum Ordner "release" gelegt werden kann.

5. Hinter "Verbinden als..." verbirgt sich ein Dialog, bei dem man sich mit seinen Microsoft-Benutzerdaten (Benutzername und Passwort) als autorisiert zu erkennen geben muss.



Der IIS-Manager wird nach einem Neustart des Computers aktiv. Der PXT-Editor kann danach ohne Internetverbindung von jedem Browser durch Eingabe von "http://localhost:80" in die Adresszeile gestartet werden.

Leider ist es nun so, dass zwar das Coden und die Simulation einwandfrei offline funktionieren, zur Kompilierung der Programmdatei (speichern als HEX-Datei) aber eine (einmalige) Internetanbindung nötig ist - es sei denn, man trifft gewisse Vorkehrungen mit Hilfe des Webbrowsers "Firefox" von "Mozilla":

Dieser Browser speichert die zur Kompilierung nötigen Informationen dauerhaft, wenn er die Prozedur einmal online durchlaufen hat. Danach kann die Internetverbindung unterbrochen und der Computer sogar ausgeschaltet werden, ohne dass die Fähigkeit verloren geht, den PXT-Editor offline voll umfänglich zu nutzen.

Und so ist bei der Einrichtung vorzugehen:

1. Firefox starten und nach einem Klick auf die 3 waagerechten Striche rechts oben über "Einstellungen" den Menüpunkt "Datenschutz & Sicherheit" aufrufen und unter "Chronik" die Einstellung vornehmen "Firefox wird eine Chronik anlegen".
2. Etwas weiter unten auf der Seite findet sich der Eintrag "Websitedaten". Hier das Feld "Einstellungen anwählen und in dem sich öffnenden Fenster alle Einträge löschen".
3. Bei funktionierender Internetanbindung nun in die Adresszeile des Firefox-Browsers "http://localhost:80" eintippen und in dem sich öffnenden Editor das Beispielprogramm speichern.
4. Anschließend die Internetverbindung unterbrechen und mit einem erneuten Klick auf die 3 waagerechten Striche rechts oben erst zu "Einstellungen" und dann zum Menüpunkt "Datenschutz & Sicherheit" wechseln. Danach unter "Websitedaten" und "Einstellungen" prüfen, ob sich dort der Eintrag "localhost" zusammen mit einer Zahlenangabe in KB befindet.
5. Wenn das so ist zur Überprüfung den Rechner ohne Netzwerk neu starten, Firefox aufrufen und in die Adresszeile "http://localhost:80" eintippen. Damit müsste der PXT-Editor starten und nun auch ohne Anbindung ans Internet voll funktionstüchtig sein, was bedeutet, dass neben dem Coden und der Simulation auch das Kompilieren der Daten funktionieren sollte.



Open Roberta Lab

Um den Calliope zu programmieren, wählt man die Internetseite der Calliope gGmbH an. Dort werden 3 Editoren vorgehalten, mit denen der Prozessor und seine Komponenten programmiert werden können.

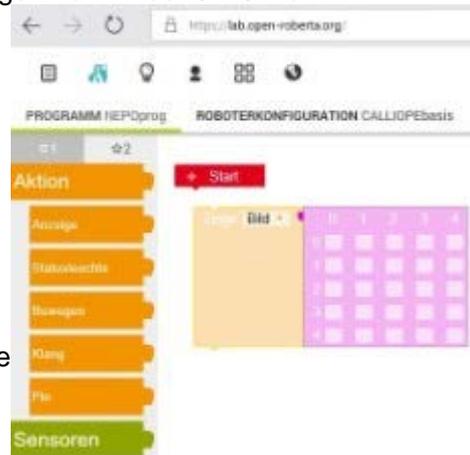
Zur Wahl stehen der „MINI EDITOR“, der MAKECODE von Microsoft und das OPEN ROBERTA LAB. Die letztgenannten Editoren heben sich in Bezug auf ihren Funktionsumfang deutlich vom erstgenannten ab. Versteht man sie, erklärt sich der Übrige von selbst.

Wie eine Programmierung mit dem OPEN ROBERTA LAB gelingt, kann mithilfe kurzer Anleitungen und einfachen Beispielen sowohl interessierten Erwachsenen als auch Kindern schnell klargemacht werden.



Das Bedienkonzept des Editors ist denkbar einfach: Aus einem Pool unterschiedlicher „Blöcke“ wird der ausgewählt, der zu einer Komponente auf dem Calliope mini passt, also beispielweise das „Zeige Bild“ mit der 5 x 5 Matrix.

Der Block erscheint nach der Auswahl mit verblaster Farbe auf der

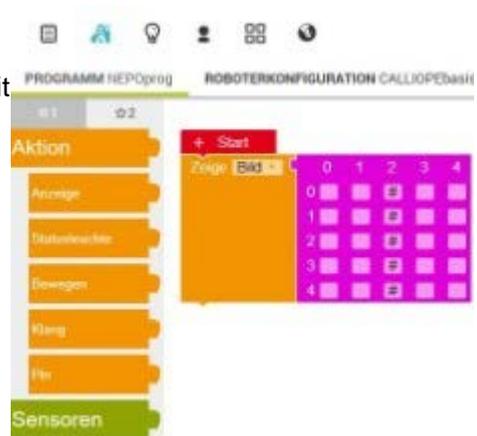


Programmieroberfläche. Bringt man ihn mit der Einkerbung in die Nähe der Spitze des „Start“-Feldes, verbinden sich die Teile und die Farbe normalisiert sich.

Auf der virtuellen Matrix kann nun ein Anzegebild entworfen werden. Dazu klickt man einzelne Felder an. Jeder Klick erzeugt ein Doppelkreuz. Aus der Summe aller Kreuze ergibt sich das gewünschte Bild, hier eine senkrechte Linie.



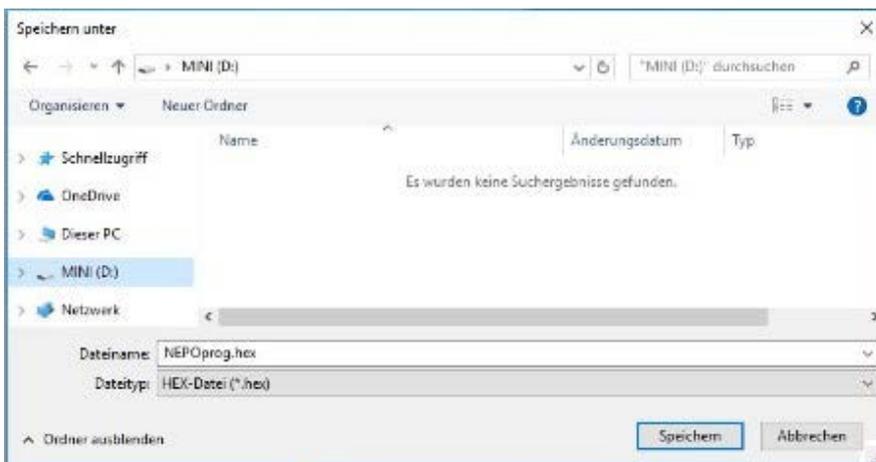
Um diese Linie auf dem Calliope mini anzuzeigen, wird er mit dem Computer verbunden, auf dem der Web-Editor läuft.



Ein Klick auf das schwarze Dreieck rechts unten öffnet ein Fenster mit Hinweisen zum Download.



Bei einem Rechtsklick auf "NEPÖprog" wird ein kleines Fenster in der unteren rechten Ecke eingeblendet, in dem auf den Menüpunkt "Link speichern unter" geklickt werden muss. Daraufhin öffnet sich das Fenster „Speichern unter“, das den Zugriff auf den Calliope möglich macht.



Dazu in der linken Spalte nach dem Eintrag „MINI“ suchen und ihn anwählen. Ist das geschehen, löst ein Klick auf „Speichern“ den Vorgang aus, bei dem der Programmcode auf den Prozessor übertragen wird. Eine flackernde LED auf dem Calliope mini links unterhalb des USB-Anschlusses zeigt das an.

Viel mehr gibt es nicht zu verstehen. Dennoch empfiehlt es sich, einige der auf der Calliope-Homepage bereitgestellten Beispiele auszuprobieren, um Sicherheit im Umgang mit dem Editor zu gewinnen.



Makecode-Editor

Der MAKECODE-Editor kommt in den vorgehaltenen Beispielen auf der Calliope-Homepage nur selten zur Anwendung. Ein sachlicher Grund ist dafür nicht auszumachen. Es wäre schade, wenn er ganz aus dem Blickfeld, geriete, besonders da er in einzelnen Funktionen dem OPEN ROBERTA LAB überlegen ist.

So zeigt sich MAKECODE nach dem Start:



Links oben findet sich ein Simulator, der die Auswirkungen der jeweiligen Programmierung auf die Hardware anzeigt. So lassen sich Fehlprogrammierungen bereits im Ansatz erkennen und ersparen damit manchen Probelauf auf der Hardware.

Der Blockstapel in der Mitte dient der Programmierung, wobei die eigentlichen Blockbausteine erst sichtbar werden, wenn ein Teil angeklickt wird. Mit gehaltener linken Maustaste können sie auf die Programmieroberfläche gezogen werden.

Ein Beispiel ist dort bei jedem Neustart bereits realisiert. Es besteht aus drei Teilen:

1. Einer zwei Blöcke umspannenden Klammer mit der Bezeichnung „dauerhaft“
2. Einem Block „zeige LEDs“ mit 7 roten Feldern, die einen Smiley symbolisieren
3. Einem weiteren Block „zeige LEDs“ mit ausschließlich weißen Feldern

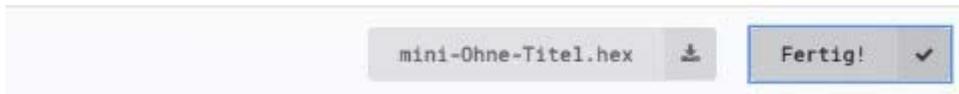
Die Klammer „dauerhaft“ legt fest, dass die eingebundenen Teile immer wieder angezeigt werden sollen: Zuerst das obere Blockbild, danach das untere, dann wieder das obere und so weiter ohne Ende. Der Smiley wird damit endlos an- und wieder abgeschaltet, er blinkt.

Die Darstellung auf dem Block „zeige LEDs“ ist durch einfaches Anklicken der Felder veränderbar: Weiß bedeutet LED aus, Rot LED an.

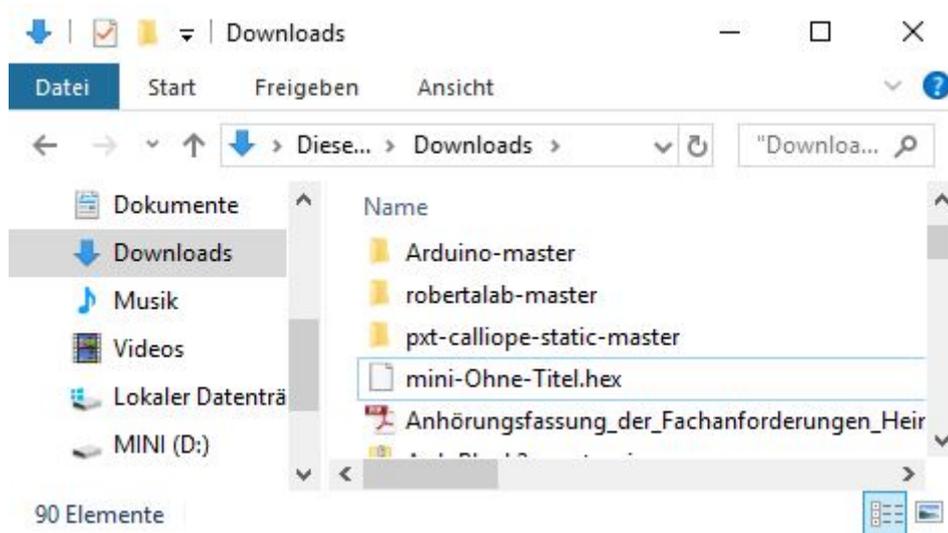
Um das Programm auf den Calliope mini zu übertragen, muss das Board über das USB-Kabel mit dem Rechner verbunden werden. Von diesem wird daraufhin ein Laufwerk mit Namen „MINI“ generiert. Wird vor dem Download kein Name und auch kein Speicherort festgelegt, wird bei einem Klick auf die grüne Schaltfläche „Herunterladen“ eine Datei mit der Bezeichnung „mini-Ohne-Titel.hex“ in den Ordner „Downloads“ kopiert und die folgenden Anzeigen eingeblendet:

Download abgeschlossen...

Bewege die Datei .hex auf das MINI-Laufwerk, um den Code auf deinen Calliope mini zu übertragen.



Um die Programmierung des Calliope mini abzuschließen, muss besagte Datei aus dem Downloadordner mit der Maus auf das Laufwerkssymbol mit der Kennzeichnung "MINI" gezogen werden.

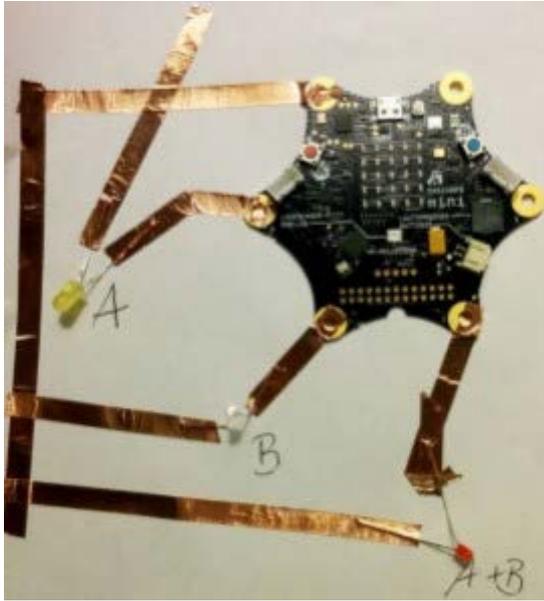


Der Download wird mit dem Flackern einer gelben LED auf dem Board quittiert.



Calliope-Pins

Um das Calliope-Mobil im Folgekapitel zu realisieren müssen Motoren und eine zusätzliche Stromquelle mit dem Calliope verbunden werden. Für solche Zwecke ist der Calliope weniger gut gerüstet. Nach den Vorstellungen der Entwickler soll ein Zugriff über die Ecken des Calliope erfolgen, wie das beispielhaft in folgender Anwendung realisiert ist:

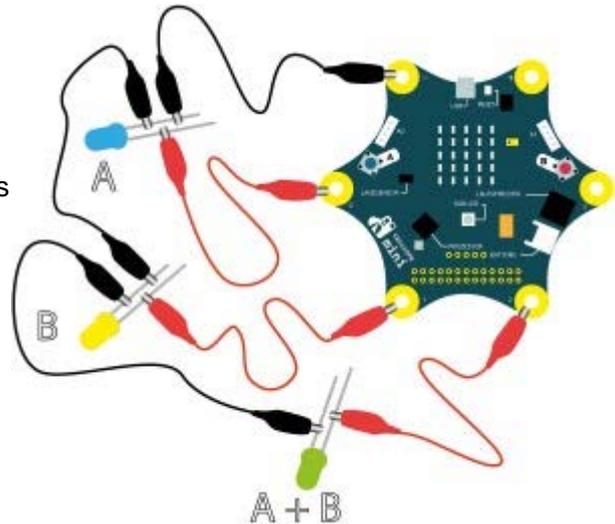


Die Kontaktflächen neben dem USB-Port liefern Gleichspannung von etwa 3 Volt. Die übrigen vier Ecken sind programmierbar und können im Makecode-Editor unter „... Fortgeschritten“ und „Pins“ als P0, P1, P2 und P3 angesprochen werden.

Die Abbildung links zeigt 3 LEDs, deren kurzen Beinchen allesamt über selbstklebendes Kupferband mit dem Minuspol verbunden sind. Die Anschlüsse P0, P1 und P2 sind jeweils mit den langen Beinchen verbunden. So gelingt es mit einer passenden Programmierung die LEDs an- und auszuschalten. Das dazugehörige Projekt findet sich [hier](#).

Klebeband können auch Verbindungsleitungen mit Krokodilklemmen verwendet werden, wie das aus der zweiten Abbildung ersichtlich ist.

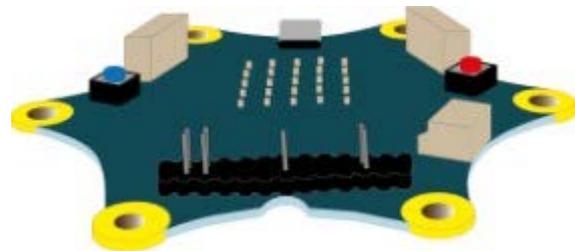
Statt



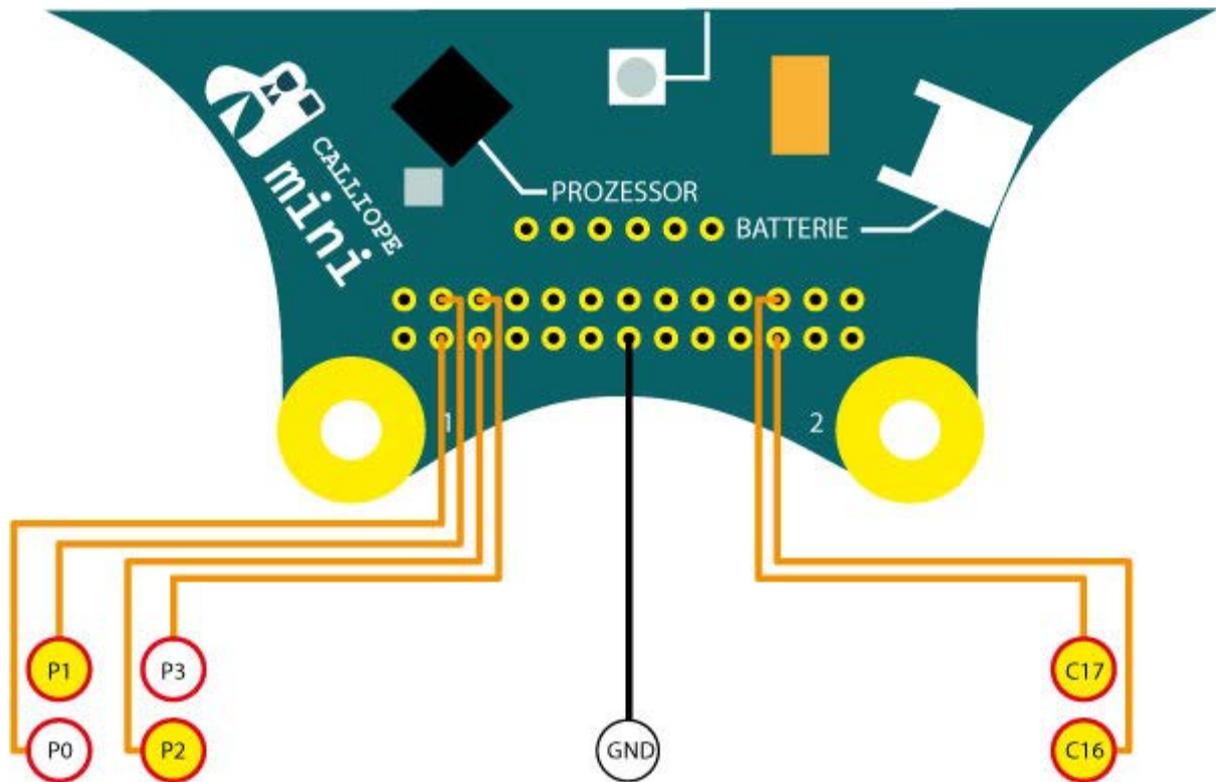
Wer es noch professioneller mag, lötet in die Ösen auf dem Calliope Stifte ein, die es ermöglichen mit Hilfe von steckbaren Drahtbrücken Verbindungen zu einem Breadboard oder anderen externen Komponenten herzustellen.



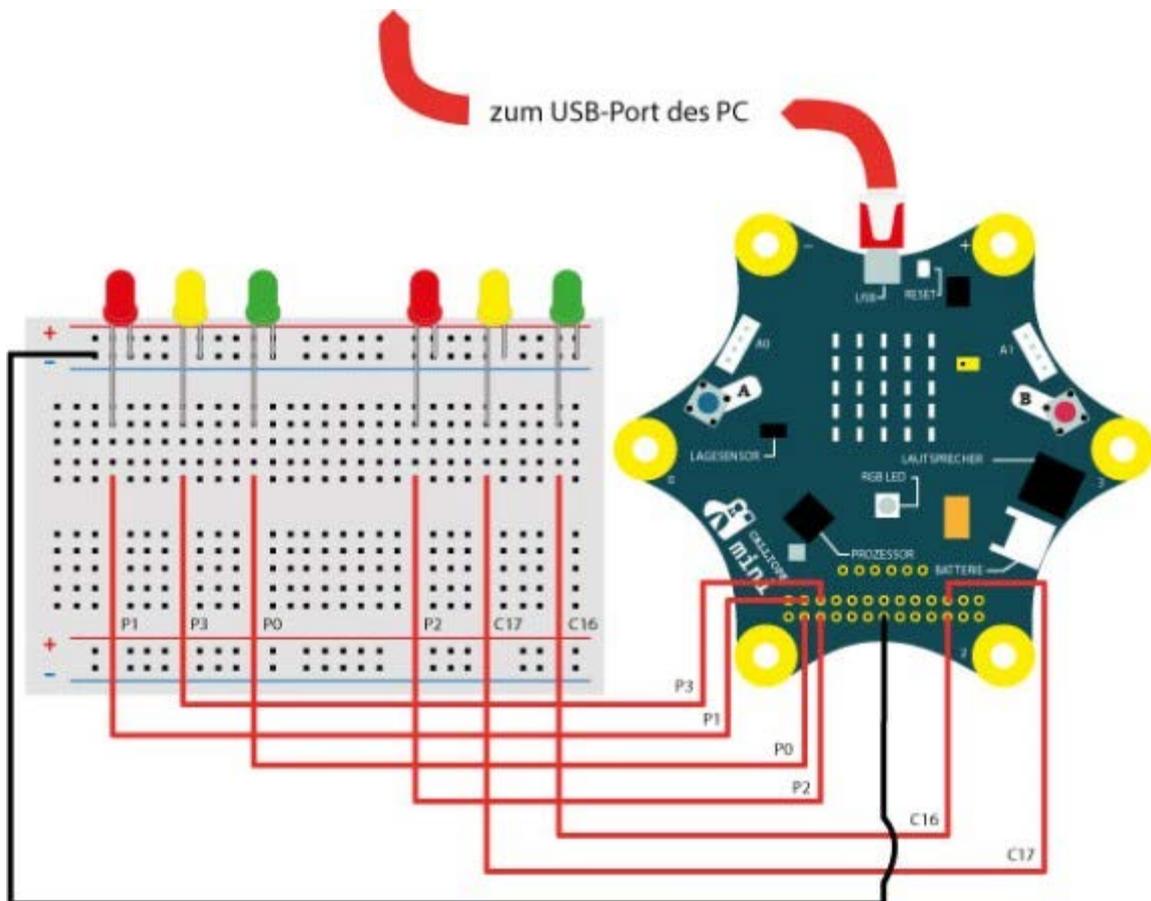
Zum



Programmieren muss man wissen, welche Pins ansprechbar sind und welche Kennung sie haben.



Eine mögliche Anwendung, bei der sechs programmierbare und ein GND-Pin benötigt werden, ist die Simulation einer Verkehrsampel. Hier der Aufbau mit Hilfe eines Steckbretts...



...und die dazu passende Programmierung. Ein Klick auf den Programmcode führt zum Download der Datei "Ampel.hex", die in einem gleichnamigen gezippten Ordner zu finden ist.

```
III dauerhaft
  schreibe digitalen Wert von pin P1 auf 1
  schreibe digitalen Wert von pin P2 auf 1
  III pausiere (ms) 5000
  schreibe digitalen Wert von pin P3 auf 1
  III pausiere (ms) 2000
  schreibe digitalen Wert von pin P1 auf 0
  schreibe digitalen Wert von pin P3 auf 0
  schreibe digitalen Wert von pin P0 auf 1
  III pausiere (ms) 5000
  schreibe digitalen Wert von pin P0 auf 0
  schreibe digitalen Wert von pin P3 auf 1
  III pausiere (ms) 2000
  schreibe digitalen Wert von pin P3 auf 0
  schreibe digitalen Wert von pin P1 auf 1
  III pausiere (ms) 2000
  schreibe digitalen Wert von pin C17 auf 1
  III pausiere (ms) 2000
  schreibe digitalen Wert von pin P2 auf 0
  schreibe digitalen Wert von pin C17 auf 0
  schreibe digitalen Wert von pin C16 auf 1
  III pausiere (ms) 5000
  schreibe digitalen Wert von pin C16 auf 0
  schreibe digitalen Wert von pin C17 auf 1
  III pausiere (ms) 2000
  schreibe digitalen Wert von pin C17 auf 0
```

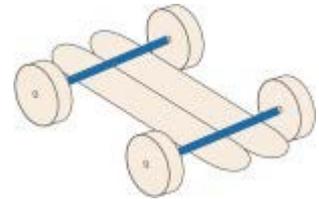
Kauftipps: [Anschlusskabel mit Krokodilklemmen](#), [Steckbrücken](#), [LEDs](#) und [Siftleisten](#).



Mobil mit Calliope

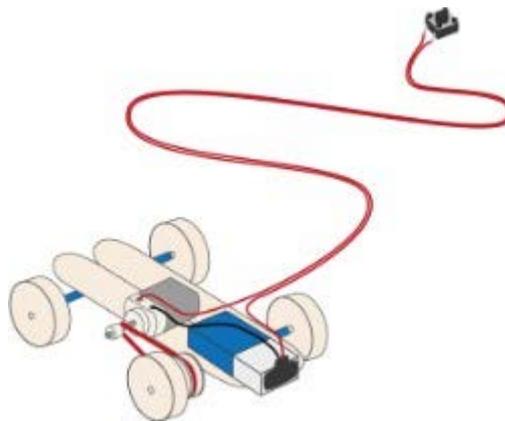
Da Grundschulen nur in Ausnahmefällen über zur Holzbearbeitung geeignete Werkzeuge und Maschinen verfügen, werden hier Fahrzeuge in den Blick genommen, die mit einfachen Mitteln hergestellt werden können. Die Entwicklung soll dabei vom einfachen Fahrgestell zum programmierbaren Fahrzeug führen.

Zur Herstellung solcher Gefährte genügen Heißkleber und Seitenschneider. Benötigte Konstruktionsmaterialien wie Trinkhalme, Schaschlikspieße, Holzspatel, Räder und Riemenscheiben können für wenig Geld im Internet bzw. Supermarkt erstanden werden (Bezugsquellen am Ende des Kapitels).



Zur Konstruktion des Fahrzeugs werden zwei Spatel der Länge nach zusammen- und zwei gleichlange Abschnitte eines Trinkhalms quer dazu geklebt. Danach werden in die Bohrungen der Räder Schaschlikspieße gesteckt, diese durch die Trinkhalme geführt und so eingekürzt, dass das gegenüberliegende Rad Halt findet, ohne die freie Drehbarkeit zu behindern. Wenn nötig, die Räder zusätzlich mit etwas Heißkleber sichern.

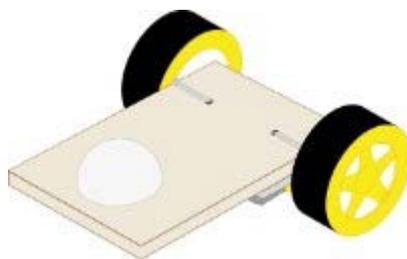
Die Funktionstüchtigkeit eines solchen Fahrzeugs kann im Rahmen eines kurzen Wettbewerbs erprobt werden: Über ein schräggestelltes Brett erfahren die Wettbewerbsfahrzeuge eine vergleichbare Beschleunigung, sodass über die zurückgelegte Strecke beim Ausrollen auf ebenem Terrain die Laufeigenschaften beurteilt werden können. Neben der zurückgelegten Strecke kann auch der Geradeauslauf in die Bewertung einbezogen werden.



In einem nächsten Schritt soll das Fahrzeug mit einem Elektromotor ausgestattet, dessen Drehbewegung über ein Gummiband und ein Schnurlaufrad auf eine der beiden Achsen übertragen wird. Der für den Betrieb des Motors nötige Strom kommt aus einem 9V Block. Mit wenig Heißkleber werden er und der Elektromotor fixiert. Ein schmaler Streifen Klebeband auf der Motorenachse verhindert das Abrutschen.

Damit das Fahrzeug kontrolliert werden kann, wird das Ein- und Ausschalten über ein Verlängerungskabel realisiert, an dessen Ende ein Taster (Schließer) angeschlossen ist. Ein Druck auf den Knopf startet den Motor. Wird der Knopf losgelassen, stoppt er.

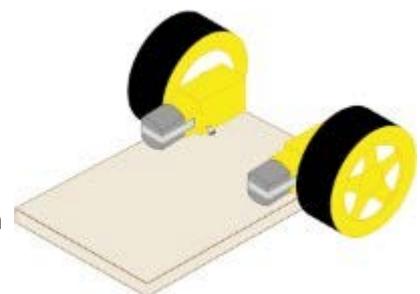
Da das Gefährt nur geradeaus fahren kann, stellt sich die Frage, wie ein solches Gefährt mit einfachen Mitteln gesteuert werden kann. Einfach meint hier den Verzicht auf eine separate Lenkung, bei der die Räder nach links oder rechts eingeschlagen werden.



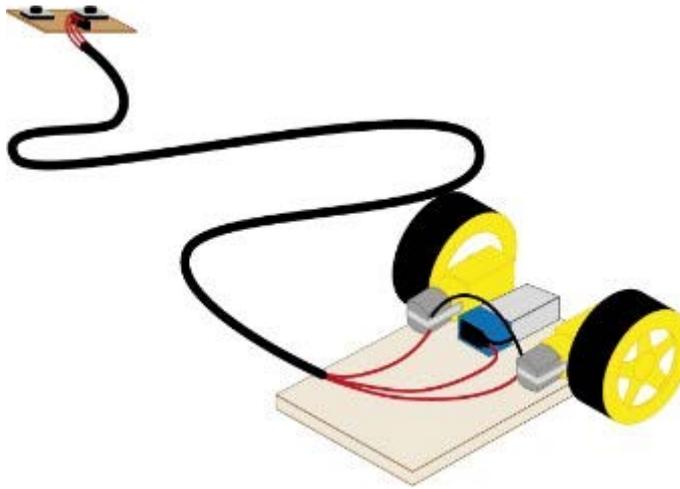
Der Weg dahin führt über die Verwendung von zwei Getriebemotoren, die samt Räder an einem Ende

eines 9 x 14 cm großen Sperrholzbrettchens befestigt werden. Das andere Ende wird mit der Hälfte eines Tischtennisballs so abgestützt, dass das Gefährt in eine

waagerechte Position kommt. Es spricht aber auch nichts gegen die Verwendung eines Bausatzes, der [hier](#) für moderates Geld erstanden werden kann.

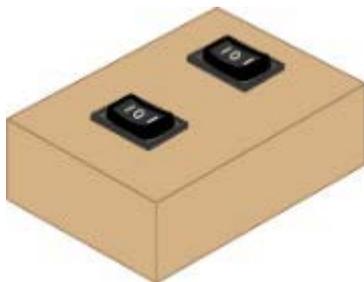
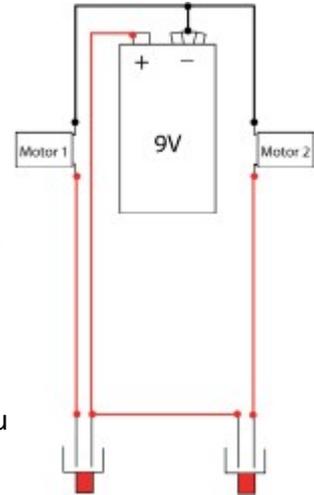


Die Lenkung funktioniert über das Ein- und Ausschalten der Motoren: Für den Geradeauslauf müssen beide Motoren, für die Kurvenfahrt ein Motor laufen, der linke für eine Rechts-, der rechte für eine Linkskurve.



Für die Fernsteuerung wird ein dreipoliges Verlängerungskabel benötigt. Die Batterie ist auf dem Fahrzeug befestigt. Ein Pol ist mit jeweils einem Pin beider Motoren verbunden. Der andere Pol wird zur Steuerkonsole mit zwei Tastern (Schließer) geführt und dort so verschaltet, dass über die eine Taste der Stromkreis zu dem einen und über die andere Taste der zu dem anderen Motor geschlossen werden kann. Bei

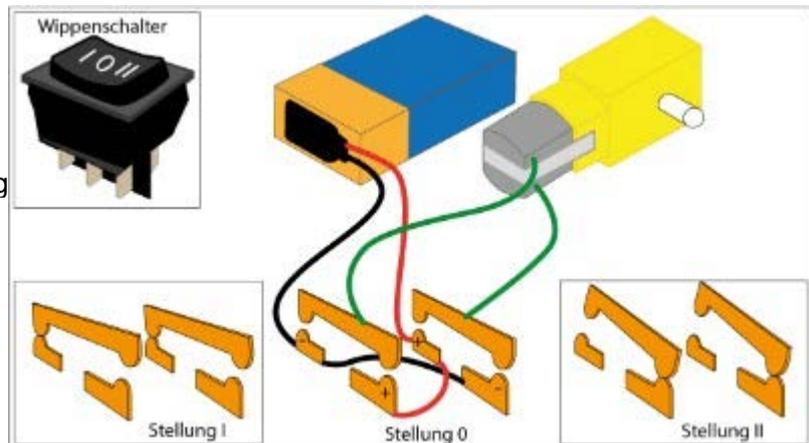
geschlossenem Stromkreis (gedrückter Taste) läuft der Motor an. Ist er unterbrochen (Taste nicht gedrückt) steht der Motor still. Für eine Geradeausfahrt müssen beide Tasten gedrückt werden, wird eine losgelassen, dreht das Fahrzeug entweder nach links oder rechts.



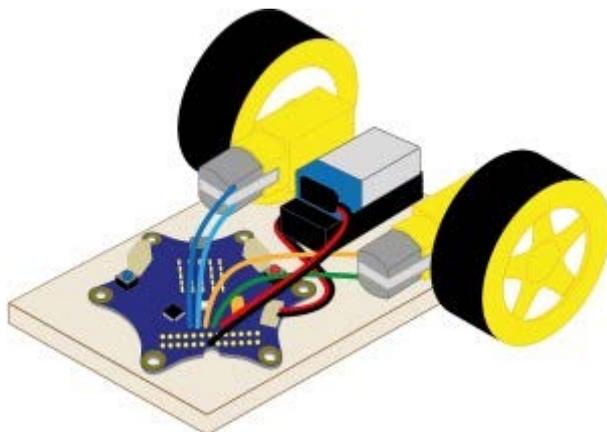
Um das Fahrzeug auch rückwärts fahren zu lassen, muss die Polung der Motoren umkehrbar sein. Das kann mit zweipoligen Wippenschaltern realisiert werden.

Die Batterie befindet sich zusammen mit den Schaltern im Steuermodul.

In der Stellung I und II bekommen die Motoren Strom und drehen sich. In Mittelstellung (Stellung 0) ist der Stromfluss unterbrochen und die Motoren kommen zum Stillstand. Da die gegenüberliegenden Kontakte der Wippe kreuzweise miteinander verbunden sind, werden beim Umschalten von Stellung I in Stellung II die Pole vertauscht. Die Motoren drehen sich dann in die jeweils entgegengesetzte Richtung.



Für den letzten Schritt, die Automatisierung der Lenkung, kommt ein Calliope mini zum Einsatz.

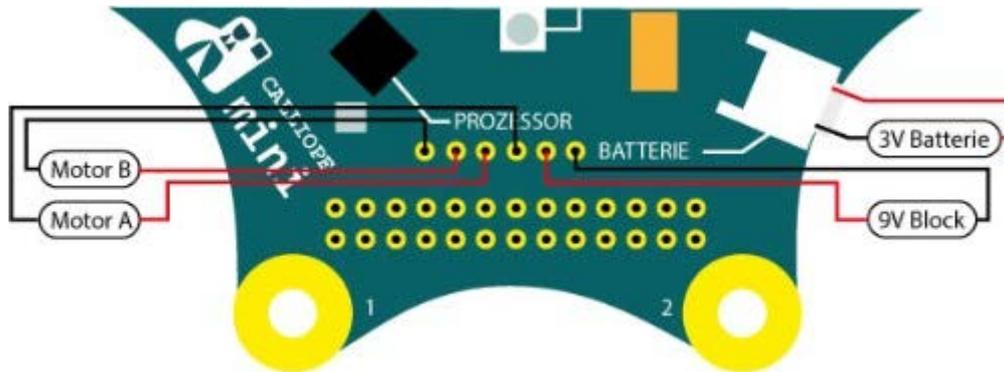


In Anlehnung an das Pinout unten werden die Kontakte vom ersten Motor mit GND und Anschluss A (Motor A), die vom zweiten Motor mit GND und Anschluss B (Motor B) verbunden.

Der 9V Block wird mit dem **Pluspol an Vin** und mit dem **Minuspol an GND verbunden**.

(Vorsicht! Ein Verwechseln der Batteriepole kann zu ernsthaften Schäden führen.)

Die 3V Batterie hat einen verpolungssicheren Stecker, der in die dazugehörige Buchse auf dem Calliope gedrückt wird.



Sind Batterien und Motoren richtig angeschlossen, wird der Editor (hier das Open Roberta Lab) gestartet und nach entsprechenden Programmierbefehlen gesucht. Fündig wird man, wenn unter **"PROGRAMM NEPOprog"** der zweite Reiter angewählt ist (siehe links). Nur dann öffnet sich bei einem Klick auf „Aktion“ ein Untermenü, in dem "Bewegen" an- und der Block "Motor Port A an Tempo % 30" ausgewählt werden muss. Er erscheint dann auf der Programmieroberfläche und wird schließlich über die Kerbe mit dem Block "Start" verbunden. Ist das geschehen, kann mit einem Klick auf das kleine Dreieck neben dem Buchstaben A ein Auswahlmü geöffnet werden (siehe hier).

Die Motoren können auf diese Weise einzelnen oder zusammen angesprochen werden, ganz ähnlich wie das mit den beiden Tasten der Kabel-Fernsteuerung möglich ist. Im Unterschied dazu können hier Zahlenwerte eingegeben werden, die Einfluss auf die Drehgeschwindigkeit der Motoren haben. Je kleiner die Zahl, desto langsamer der Motor. 0 bedeutet Stillstand.



Um die Laufdauer der Motoren festzulegen, muss unter „Kontrolle“ der entsprechende Steuerblock gefunden und an den bestehenden Programmabschnitt angegliedert werden (siehe links). Die Zahl im blauen Feld kann verändert werden. 500 entspricht einer halben, 1000 einer Sekunde.

Hier ein einfaches Programmierbeispiel, mit dem das Fahrzeug 5 Sekunden geradeaus, dann eine Kurve macht, um erneut 5 Sekunden geradeaus zu fahren und schließlich anzuhalten.



Bezugsquellen:

[Holzspatel](#) und [Riemenscheiben](#) aus Holz (Schnurlaufräder) gibt es bei Opitec, [DC-Motoren](#), zweipolige [Wippenschalter](#) bei Pollin, [Drucktaster](#), [Buchsen](#)- und [Stiftleisten](#) sowie [Batterieclips](#) bei Reichelt.

[Getriebemotoren](#) mit Rädern oder ein [kompletter Bausatz](#) sowie [Jumperkabel](#) können im Roboter Bausatz-Shop gekauft werden.

[Zwillingslitze](#) und vieradriges [Telefonkabel](#) gibt es als Meterware günstig bei der Firma Völkner.

Hochwertiges Gewebe-[Klebeband](#), das sehr gut klebt und leicht in Stücke und schmale Streifen gerissen werden kann, gibt es im Klebeshop24.

[Steckleisten](#), die leicht durch Abbrechen auf die benötigte Längen gekürzt werden können, und dazu passende [Kabelbrücken](#) gibt es bei Exp-Tech.

9V Batterieblöcke gibt es günstig bei Aldi und Lidl.



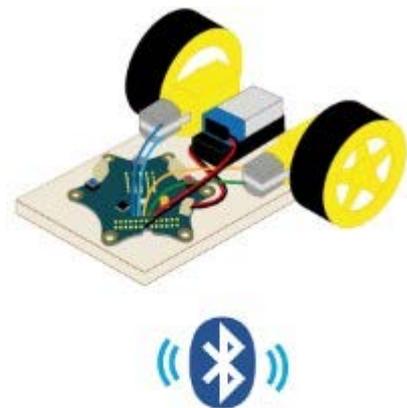
Fernsteuerung

Da Calliope minis über Bluetooth miteinander kommunizieren können, kann ein zweiter Controller als Fernsteuerung dienen: Einer steuert die Motoren, der zweite gibt die Befehle.

```

beim Start
  setze Gruppe 1
  lege Übertragungsstärke 7
  
```

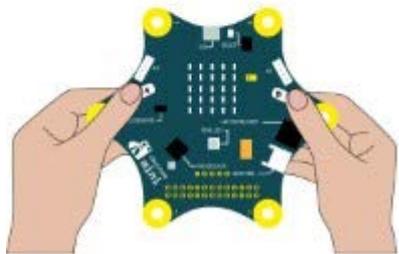
Zum Einrichten der Funkverbindung zwischen den beiden Mikrocontrollern dient der Befehlsblock links. Die Calliopes müssen der gleichen "Gruppe" zugeordnet sein. "7" ist die höchstmögliche Übertragungsstärke. Sind beide Mikrocontroller so programmiert, verbinden sie sich bei jedem Einschalten selbstständig.



```

dauerhaft
  wenn Knopf A ist gedrückt
  dann sende Zahl 0
  ansonsten sende Zahl 1
  wenn Knopf B ist gedrückt
  dann sende Zahl 2
  ansonsten sende Zahl 3
  
```

In Analogie zur 2-Tasten-Handsteuerung im Kapitel "Mobil mit Calliope" werden die Steuerbefehle über das Drücken der Tasten "A" und "B" auf dem Calliope mini generiert:



Taste "A" gedrückt erzeugt eine "0", losgelassen eine 1.

Taste "B" gedrückt eine "2" und losgelassen eine 3.

Empfängt der Controller auf dem Fahrzeug diese Zahlen, wird bei "0" (gedrückter Taste "A") Motor "A" gestartet und bei "1" (Taste A nicht gedrückt) Motor "A" gestoppt. Entsprechendes gilt für Motor "B" beim Empfang der Zahlen "2" und "3".

Werden beide Tasten gedrückt, fährt das Fahrzeug geradeaus. Wird eine Taste losgelassen, bewirkt das beim Fahrzeug - je nach Taste - eine Richtungskorrektur nach links oder rechts.

Nicht vergessen: Zusätzlich zu dem rechts abgebildeten Code müssen - wie oben - für die Bluetooth-Funkverbindung "Gruppe" und "Übertragungsstärke" gesetzt werden.

```

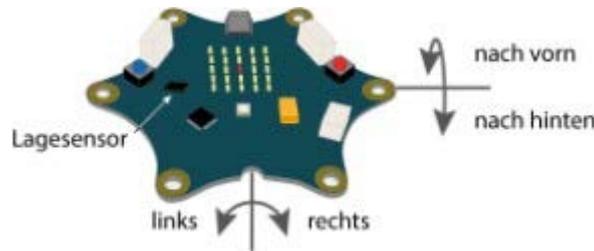
beim Start
  setze Gruppe 1
  lege Übertragungsstärke 7
  
```

```

wenn Datenpaket empfangen receivedNumber
  wenn receivedNumber == 0
  dann Motor A mit 100
  wenn receivedNumber == 1
  dann Motor A mit 0
  wenn receivedNumber == 2
  dann Motor B mit 100
  wenn receivedNumber == 3
  dann Motor B mit 0
  
```

Die Programmteile liegen auf der Programmieroberfläche unverbunden über- unter- oder nebeneinander.

Wer zudem an einer Steuerung mit Hilfe des auf dem Callipe mini verbauten Lagesensors interessiert ist, der findet die Details in einer Handreichung, die [hier](#) heruntergeladen werden kann.



Der Lagesensor kann Bewegungen des Controllers wahrnehmen: Durch Kippen nach vorn oder hinten können Impulse zum Beschleunigen oder Abbremsen gegeben werden, Drehungen um die Längsachse nach links und rechts machen Kurvenfahrten möglich.

Rückwärtsfahren geht leider nicht, weil die Ausstattung des Calliope das dazu nötige

Umpolen beider Motoren nicht zulässt.



Servosteuerung

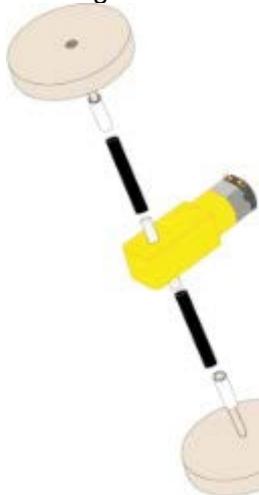
Der Bau des hier vorgestellten Fahrzeugs wurde im Rahmen des Technikunterrichts an einer Schule mit voll ausgestatteten Werkräumen realisiert.

Als die Schülerinnen und Schüler das Fahrzeug konstruierten, war ihnen nicht bekannt, dass es später zum Selbstfahrer umgestaltet werden sollte.

So waren das für Kinderhände gedachte "Lenkrad" samt beweglicher Vorderachse und starrer Hinterachse bereits realisiert, als die Idee zum Umbau geboren wurde.

Um das relativ schwere Fahrzeug bewegen zu können, war ein Getriebemotor mit entsprechendem Durchzugsvermögen nötig.

Die Wahl fiel auf auf einen Motor mit beidseitig herausgeführter Antriebswelle (Bezugsquelle am Ende der Ausführungen).



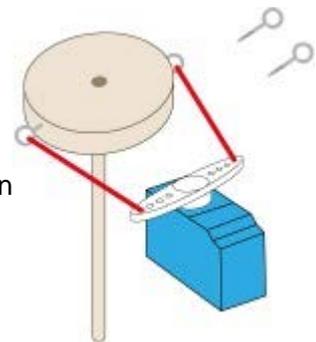
Um ihn einzubauen, musste die Hinterachse aufgetrennt und die Enden eingekürzt werden. Zwei Abschnitte eines dünnen Kunststoffrohrs und Klebeband ermöglichten das Anbringen der Räder. Etwas Heißkleber gab der Sache zusätzlichen Halt.

Der Motor samt Rädern wurde dann - ebenfalls mit mit Heißkleber - mittig auf dem hinteren Vierkantholz festgesetzt.

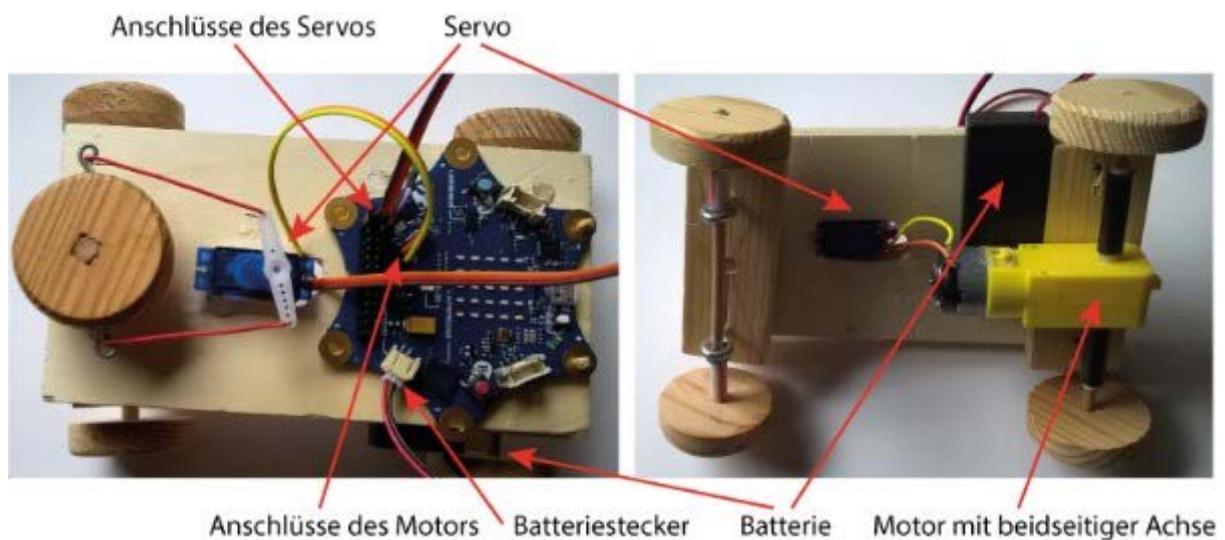
Für die Lenkung kam ein einfacher Servo-Motor (Bezugsquelle am Ende) zum Einsatz, dessen Hebelarme über Schraubösen und Klingeldraht mit dem Lenkrad verbunden wurden.

Die Steuerung der beiden Motoren übernahm ein Calliope mini.

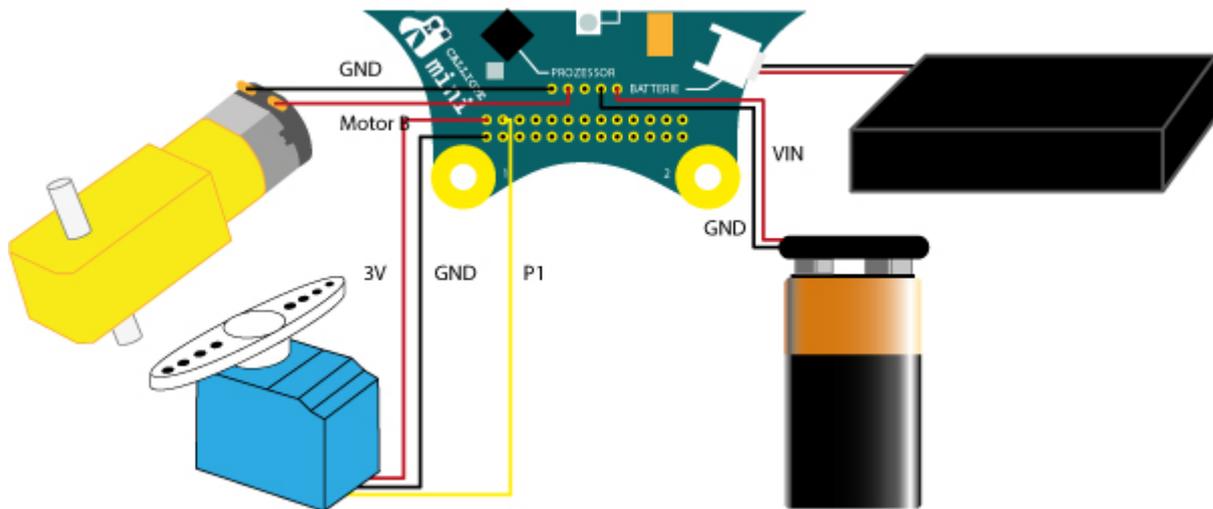
Nach erfolgreichem Zusammenbau präsentiert



sich das Gefährt so:



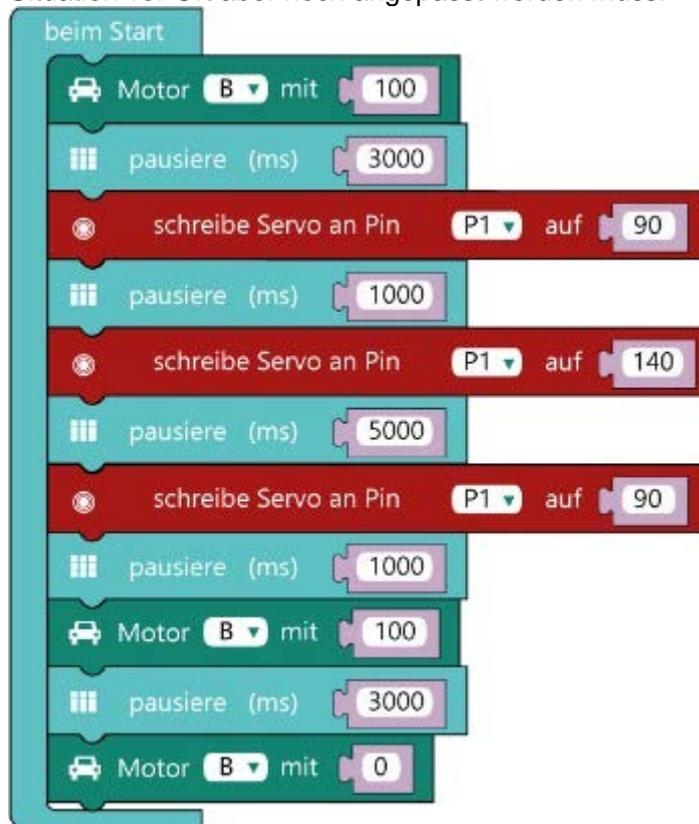
Wie der Calliope mit den Stromquellen und den Motoren verbunden wurde, zeigt der folgende Anschlussplan:



Wie [hier](#) näher ausgeführt, wurde ein Calliope mini verwendet, der bereits mit Stiftleisten ausgestattet war, sodass sowohl die Motoren als auch die zusätzliche Spannungsquelle (9V Block) dort angeschlossen werden konnten.

Für eine erste Probefahrt erschien es ausreichend, die Programmierung so anzulegen, dass das Fahrzeug eine bestimmte Strecke geradeaus, dann eine Kurve fährt, zum Startpunkt zurückkehrt und dort anhält.

Hier die dazu passende Programmierung, die alle Steuerkomponenten enthält, der konkreten Situation vor Ort aber noch angepasst werden muss:



Die Programmierblöcke zur Steuerung der Motoren finden sich im MAKECODE-Editor unter „Grundlagen“, „Motoren“ und durch Anklicken von „... Fortgeschritten“ unter „Pins“

Die Zahlenwerte und Buchstaben können auf den jeweiligen Blöcken durch Anklicken der entsprechenden Felder angepasst werden.

Hinweis: Beim Kurvenfahren, also solange der Servomotor eingeschaltet war, beschleunigte das Fahrzeug, was nicht zu erwarten war. Die Ursache konnte nicht geklärt werden, dürfte aber auf eine Fehlfunktion des Calliope mini hindeuten.

[Getriebe-](#) und [Servomotor](#) können bei "Roboter-Bausatz.de" preisgünstig erworben werden.

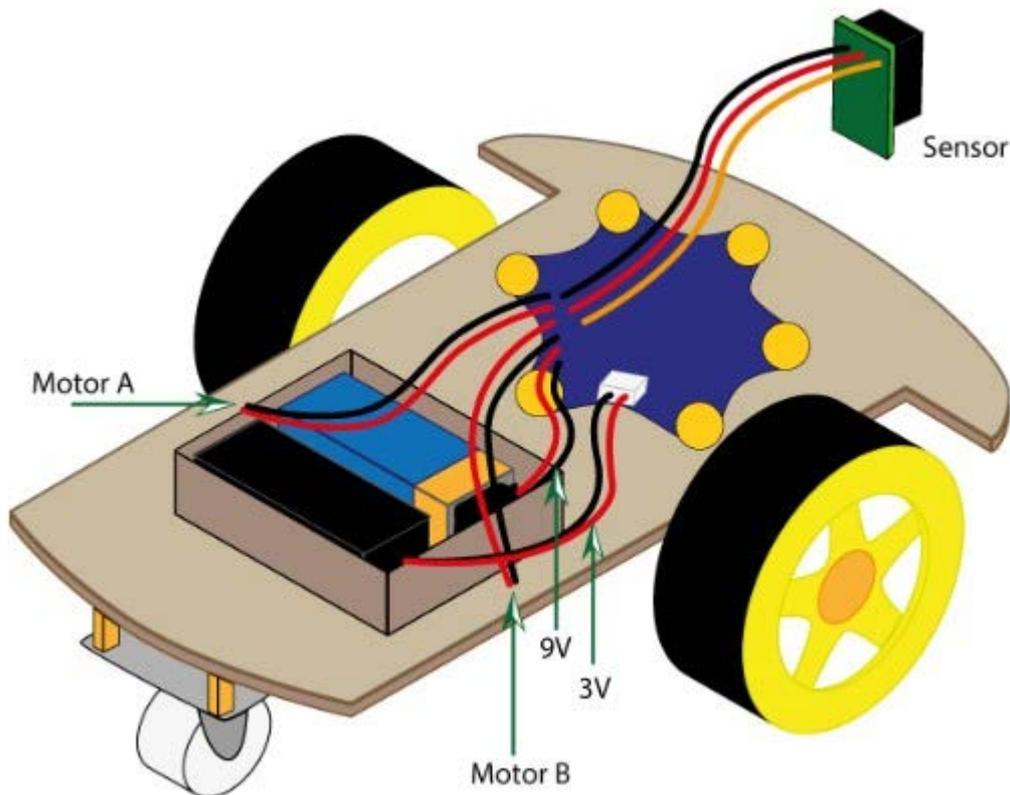


Kollisionsschutz

Ein nächster Schritt bei der Steuerung von Fahrzeugen ist, das Gefährt ein Stück weit autonom zu machen. Die Fähigkeit, ein Hindernis zu erkennen und rechtzeitig vor einem Zusammenprall zu stoppen, soll hier als Beispiel dienen.

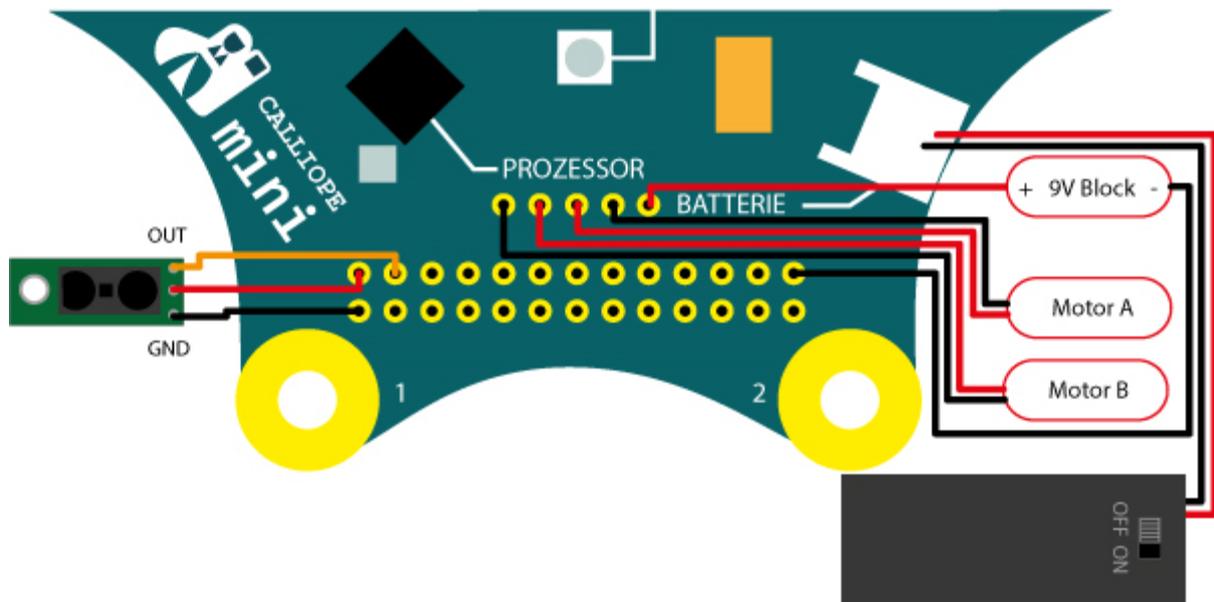
Als Sensor kommt der "GP2Y0D810" zum Einsatz, der ein Hindernis auf etwa 10 cm Entfernung erkennen kann. Das ist nicht viel, sodass die Geschwindigkeit des Fahrzeugs gedrosselt werden muss, damit es rechtzeitig zum Stehen kommt.

Als Fahrgestell kommt ein "2WD Motor Chassis" zum Einsatz, auf dem Sensor, Calliope mini und die Batterien wie in der Zeichnung unten Platz finden. Die Teile können mit wenig Heißkleber auf der Trägerplatte fixiert werden. Eine kleine Pappschachtel nimmt die Batterien auf.

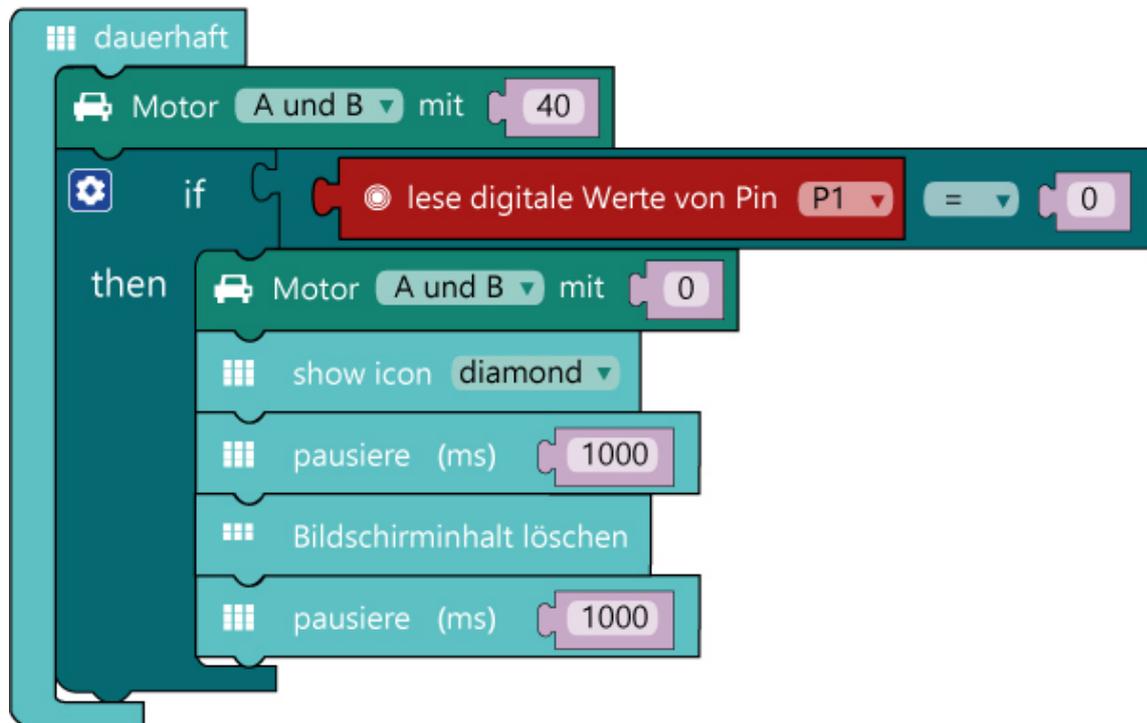


Um die Bauteile mit dem Calliope mini zu verbinden, sind kleinere Lötarbeiten erforderlich: Stiftleisten am Sensor und am Calliope mini, Steckbrücken an den Motoren und am Clip des 9V Blocks. Die Steckbrücken müssen mit weibliche Steckern versehen sein.

Die Zeichnung unten zeigt, wie die Verdrahtung vorzunehmen ist, wobei nicht dargestellt ist, dass die Anschlüsse über Stiftleisten erfolgen.



Hier der dazugehörige Programmcode, der ein wenig aufgepeppt ist durch die Einbindung einer Anzeige auf der LED-Matrix des Calliope: Eine rot blinkenden Raute soll auf das stehende Fahrzeug und das Hindernis aufmerksam machen.



Ein ähnliches Bastelprojekt unter Verwendung eines Ultraschallsensors zur Distanzmessung und eines Arduino-Mikrocontrollers auf einem Selbstbaufahrzeug findet sich [hier](#)

Das "2WD Motor Chassis" kann [hier](#), Steckbrücken [hier](#), Batterieclip [hier](#) und der Abstandssensor können [hier](#) preisgünstig erworben werden.

9V Blocks gibt es besonders günstig bei Aldi oder Lidl.



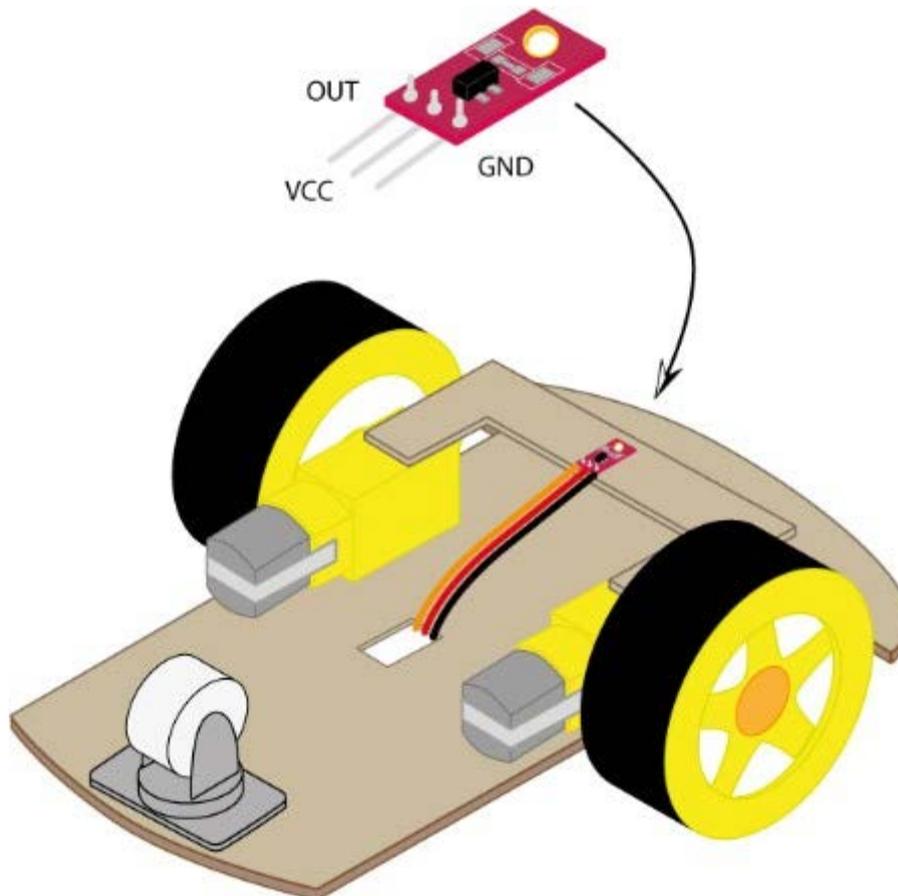
Linienfolger

Beim Zusammenstellen der Materialien sollte das Augenmerk stets in die Richtung gehen, Programmieraufwand und Basteln gleichermaßen Raum zu geben. Die Verwendung von natürlichen Materialien wie Holz und die damit verbundenen Erfahrungen mit Werkzeugen und Werkstoff sollten stets Vorrang haben vor der Verwendung vorgefertigter Teile. Das Zusägen von Platten und Leisten, die Auswahl geeigneter Komponenten wie Rundhölzer und Schraubösen zum Anbringen von Rädern und Achsen, die Konstruktion beweglicher Teile für die Lenkung passen in den Lehrplan und sind allemal lehrreicher als das Zusammenstecken vorgefertigter Normteile nach einem vorgegebenen Plan.

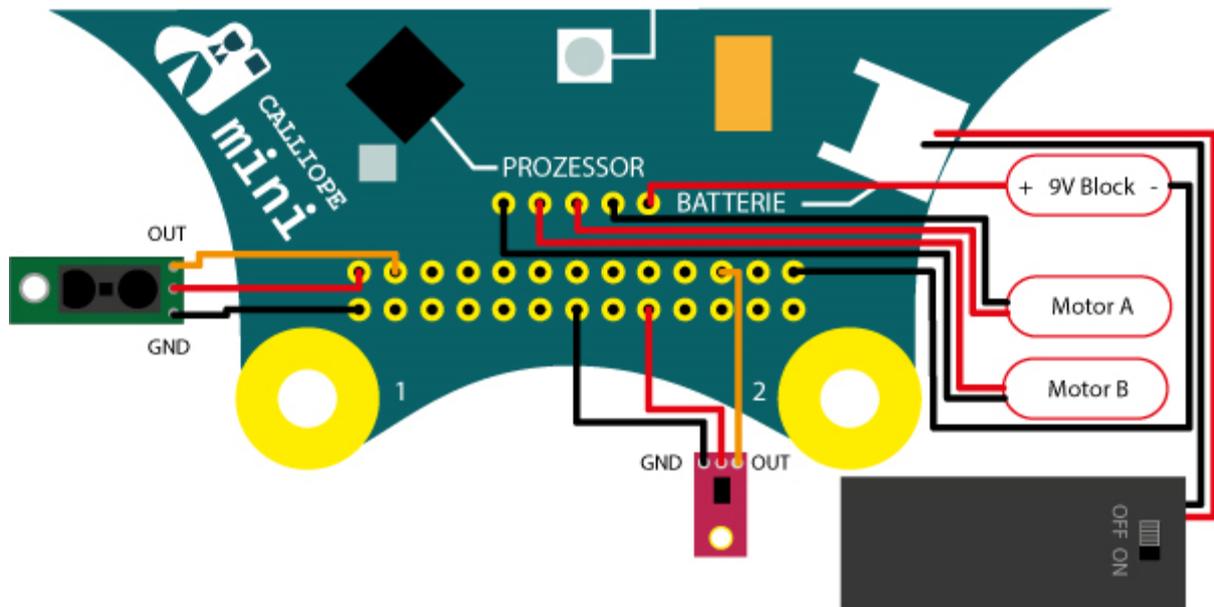
Auch wenn bei dem hier verwendeten Roboter-Bausatz bestimmte Komponenten wie Halterungen für Batterien und Sensoren selbst gefertigt werden müssen und auch Lötarbeiten nötig sind, wäre der Bau eines eigen Fahrzeugs die anstrebenswerte Alternative.

Wie schon in den vorherigen Kapiteln ist die Wahl ist auf dieses Fahrzeug gefallen, weil es sehr preiswert ist und in Bezug auf seine Steuerbarkeit den angepeilten Erfordernissen gerecht werden kann. Außerdem mangelt es an Anhaltspunkten, ob die Entwicklung eines solchen Fahrzeugs im Grundschulunterricht unabhängig von seiner Programmierung sinnstiftend ist.

Konkret handelt es sich hier um das gleiche Fahrzeug wie im vorangestellten Kapitel, das zusätzlich mit Reflexionssensor und Halterung ausgestattet werden muss. Dargestellt ist die Unterseite.



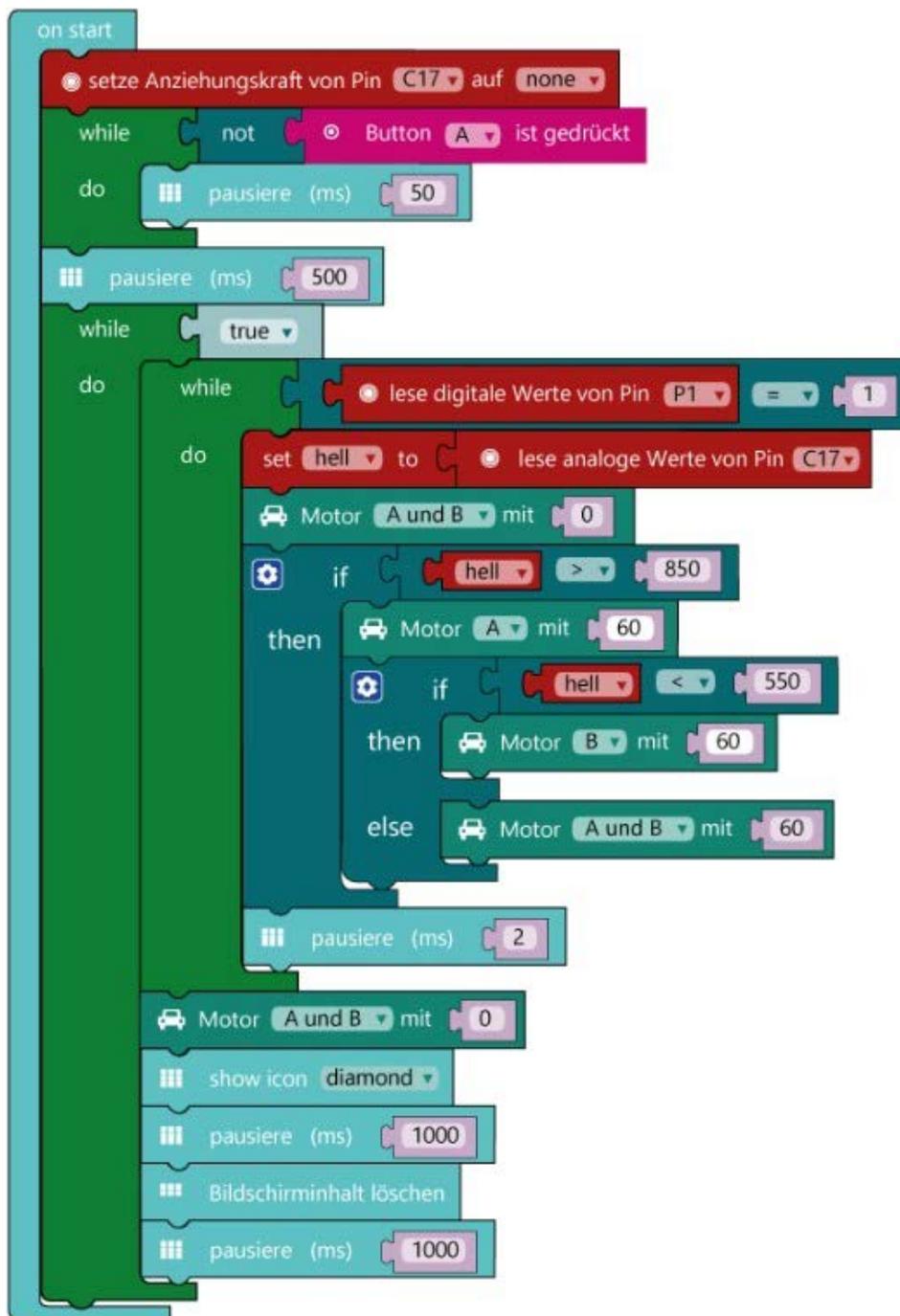
GND am Sensor wird mit einem GND-Pin am Calliope, VCC mit einem Pin, der 3V liefern kann, OUT mit dem Pin C17 verbunden.



Damit verfügt das Fahrzeug neben dem Kollisionssensor über einen zweiten Sensor, der erkennen kann, ob der Untergrund "hell" oder "dunkel" ist. Ausgestattet mit dieser Fähigkeit kann es einer Linie folgen, die von einer schwarzen Bahn auf der einen und einem weißen Untergrund auf der anderen Seite gebildet wird.

Die Idee für ein solches Fahrzeug ist an den Ausführungen von Werner auf "hackster.io" orientiert. Dort findet sich auch ein Video.

Der Programmcode ist recht anspruchsvoll und dürfte für Grundschul Kinder eine besondere Herausforderung darstellen.



Ein "echter" Linienfolger benutzt in der Regel zwei Sensoren, die jeweils einen Motor steuern. Liegt der schwarze Pfad genau zwischen den Sensoren, laufen beide Motoren und das Fahrzeug fährt geradeaus. Kommt ein Sensor über Schwarz, wird der jeweiligen Motor kurz gestoppt und so die Richtung korrigiert.

Das Einbinden eines zweiten Sensors machte die obige Programmierung noch komplizierter, sodass an dieser Stelle darauf verzichtet wird.

Wer mehr über Linienfolger wissen möchte, der findet [hier](#) und in diesem [Download](#) weitere Anregungen.

Der Reflexionssensor kann [hier](#) erworben werden.

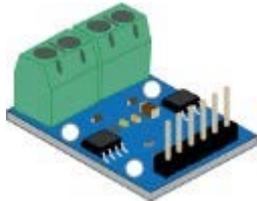


Mobil mit Arduino

Für die Steuerung von Fahrzeugen ist der Calliope mini nur eingeschränkt tauglich, da er lediglich über zwei Anschlüsse für Motoren verfügt. Folglich können bei der Verwendung von zwei Motoren diese nur an- und abgeschaltet. Das bedeutet, dass Fahrzeuge weder auf der Stelle wenden noch Rückwärtsfahren können. Ist aber genau das gesucht, muss ein anderer Controller eingesetzt werden, beispielsweise ein Arduino Nano. Dieses Mikrocontrollerboard kostet nur einen Bruchteil dessen, was für den Kauf eines Calliope mini aufgewendet werden muss.

Im Gegensatz zum Calliope mini verfügt der Nano über keinen Motorentreiber. Ein solches Bauteil muss nachgerüstet werden.

Infrage kommen dafür beispielsweise ein L9110 (links) oder L293D (rechts).



Der Zusammenbau der Komponenten gelingt am leichtesten mit Hilfe eines Breadboards.

Der L9110 wird über die Stiftleiste wie in der Abbildung gezeigt mit dem Breadboard verbunden. Mithilfe von Drahtbrücken werden die nötigen Anschlüsse hergestellt:

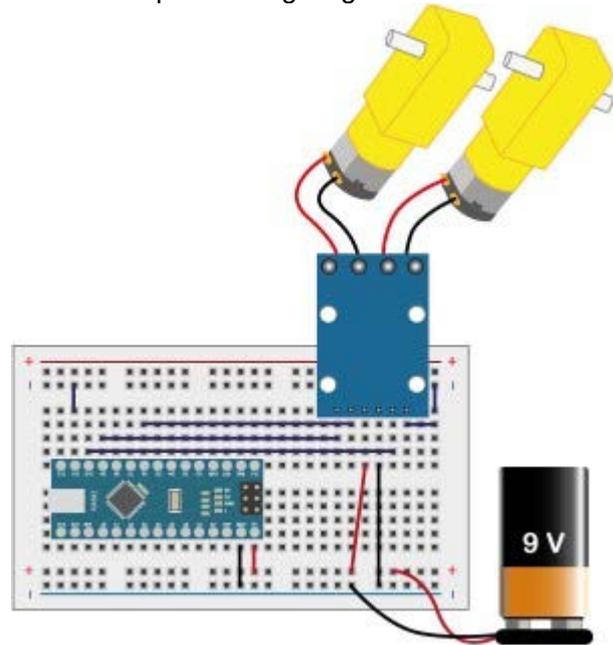
L9110 ↔ Nano

B-IA ↔ D11

B-IB ↔ D10

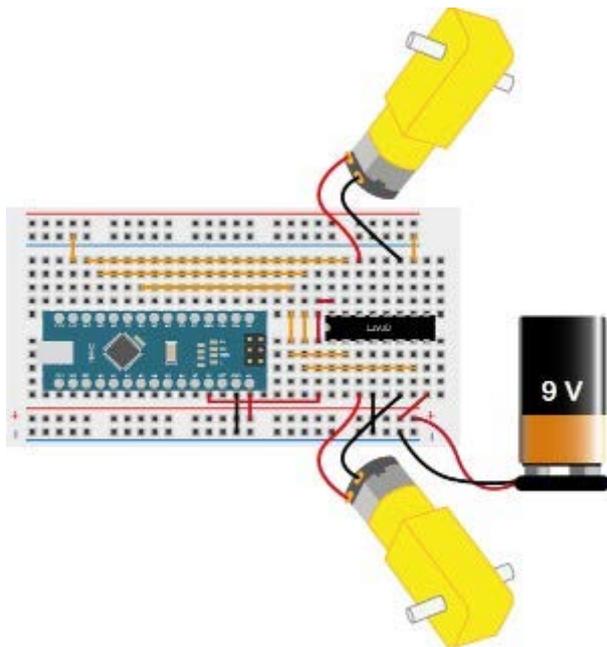
A-IA ↔ D9

A-IB ↔ D6.



Die Motoren werden mit den als "A" und "B" gekennzeichneten Schraubkontakten verbunden. Durch Einklemmen kurzer Drahtstifte können auch handelsübliche Jumperkabel benutzt werden.

Motorentreiber und Nano bekommen Strom von einem 9V Block. Der Pluspol der Batterie wird mit Vin am Nano und VCC am Motorentreiber verbunden. Der Minuspol an beiden Teilen mit GND.



Der L293D wird so auf das Breadboard gesetzt, dass dessen Pins unverbunden sind und einzeln angesteuert werden können.

Der Anschlussplan, soweit nicht aus der Abbildung links ersichtlich, kann [hier](#) nachgelesen werden.

Zum Programmieren empfiehlt sich eine dem "Open Roberta Lab" oder dem "Makecode Editor" vergleichbare Programmieroberfläche. Dazu ist eine Anleihe am [Institut für Geoinformatik](#) der Uni Münster zu empfehlen. Das Team des Schüler- und Forschungslabors hält nach den Prinzipien der [OpenScience](#) das Projekt "[senseBox](#)" eine [deutschsprachige Version](#) von Ardublockly im Netz vor.

Alternativ zu dieser internetbasierten Anwendung kann "Ardublockly" auch lokal auf jedem Rechner ausgeführt werden. Die dafür

nötigen Dateien stehen für die verschiedenen Betriebssysteme unter [GitHub](#) zum Download bereit.

Für ein 64-bitiges Windows ist das die Datei "ardublockly_Windows64_2017-06-26_13.55.zip". Diese Datei ist nach dem Download an einen Ort der eigenen Wahl entpacken. Zum Starten des Editors in dem neuen Ordner den Unterordner "ardublockly" öffnen und darin die Datei "index.html" mit einem Doppelklick starten.

Die zur Programmierung des Nano nötigen Blöcke finden sich unter "Funktionen" (Schleifen), "Eingang/Ausgang" (analoge und digitale Pins) und "Zeit" (Dauer einer Aktion).



Die in der Abbildung gezeigte Blöcke "Setze analogen pin# auf ..." sorgen dafür, dass der jeweils angegebene Pin in Intervallen vom μ Controller mit dem Pluspol der Stromquelle in Kontakt gebracht wird. Variiert werden kann das zwischen 0 und 255. Wird so beispielsweise der eine Pol des Motors auf 0, der andere auf 255 gesetzt, dreht der Motor mit voller

Geschwindigkeit. Ein Vertauschen der Zahlwerte kehrt die Drehrichtung um. Wird der Wert verringert, dreht der Motor langsamer. Beide Pins auf 0 (oder 255) führt zum Stillstand.

Unter "Zeit" findet sich der Block "Warte 1000 millisekunden", mit dem die Dauer einer Aktion festgelegt werden kann.

Die Auswahl eines Programmblocks erzeugt automatisch den Arduino-Quellcode. Über Copy & Paste kann dieser in die Arduino IDE übertragen und so zur Programmierung des μ Controllers genutzt werden.

Weitere Details zur Programmierung eines Arduino- μ Controllers finden sich [hier](#).

Bezugsquellen: Arduino [Nano](#) , [L9110](#), [L293D](#), [Breadboard](#) und [Steckbrücken](#) oder auch [Jumperkabel](#). Letztere gibt es in den Ausführungen Buchse-Buchse, Buchse-Stecker und Stecker-Stecker.

