

# Das Bergner · Franken · Kleeberger · Leonhardt · Lukas · Pesch · Prost · Thar · Wassong Calliope-Buch

Spannende Bastelprojekte  
mit dem Calliope-Mini-Board



**CALLIOPE  
mini**

mediothek@phgr.ch

plus\_d8d429aa8bf84ea828a2-16566-13068-1



**dpunkt.verlag**

## **Was sind E-Books von dpunkt?**

Unsere E-Books sind Publikationen im PDF- oder ePub-Format, die es Ihnen erlauben, Inhalte am Bildschirm zu lesen, gezielt nach Informationen darin zu suchen und Seiten daraus auszudrucken.

Sie benötigen zum Ansehen den Acrobat Reader oder ein anderes adäquates Programm bzw. einen E-Book-Reader.

E-Books können Bücher (oder Teile daraus) sein, die es auch in gedruckter Form gibt (bzw. gab und die inzwischen vergriffen sind). (Einen entsprechenden Hinweis auf eine gedruckte Ausgabe finden Sie auf der entsprechenden E-Book-Seite.)

Es können aber auch Originalpublikationen sein, die es ausschließlich in E-Book-Form gibt. Diese werden mit der gleichen Sorgfalt und in der gleichen Qualität veröffentlicht, die Sie bereits von gedruckten dpunkt.büchern her kennen.

## **Was darf ich mit dem E-Book tun?**

Die Datei ist nicht kopiergeschützt, kann also für den eigenen Bedarf beliebig kopiert werden. Es ist jedoch nicht gestattet, die Datei weiterzugeben oder für andere zugänglich in Netzwerke zu stellen. Sie erwerben also eine Ein-Personen-Nutzungslizenz.

Wenn Sie mehrere Exemplare des gleichen E-Books kaufen, erwerben Sie damit die Lizenz für die entsprechende Anzahl von Nutzern.

Um Missbrauch zu reduzieren, haben wir die PDF-Datei mit einem Wasserzeichen (Ihrer E-Mail-Adresse und Ihrer Transaktionsnummer) versehen.

Bitte beachten Sie, dass die Inhalte der Datei in jedem Fall dem Copyright des Verlages unterliegen.

## **Wie erhalte ich das E-Book von dpunkt?**

Sobald der Bestell- und Bezahlvorgang abgeschlossen ist, erhalten Sie an die von Ihnen angegebene Adresse eine Bestätigung. Außerdem erhalten Sie von dpunkt eine E-Mail mit den Downloadlinks für die gekauften Dokumente sowie einem Link zu einer PDF-Rechnung für die Bestellung.

Die Links sind zwei Wochen lang gültig. Die Dokumente selbst sind mit Ihrer E-Mail-Adresse und Ihrer Transaktionsnummer als Wasserzeichen versehen.

## **Wenn es Probleme gibt?**

Bitte wenden Sie sich bei Problemen an den [dpunkt.verlag](mailto:dpunkt.verlag@dpunkt.de)  
e-mail: [ebooks@dpunkt.de](mailto:ebooks@dpunkt.de)  
fon: 06221/1483-0.

Papier  
plus<sup>+</sup>  
PDF.

Zu diesem Buch – sowie zu vielen weiteren dpunkt.büchern – können Sie auch das entsprechende E-Book im PDF-Format herunterladen. Werden Sie dazu einfach Mitglied bei dpunkt.plus<sup>+</sup>:

**[www.dpunkt.plus](http://www.dpunkt.plus)**

[mediothek@phgr.ch](mailto:mediothek@phgr.ch)

plus\_d8d429aa8bf84ea828a2-16566-13068-1

**Nadine Bergner • Patrick Franken • Julia Kleeberger •  
Thiemo Leonhardt • Mario Lukas • Mario Pesch • Natalia Prost •  
Jan Thar • Lina Wassong**

# **Das Calliope-Buch**

**Spannende Bastelprojekte mit dem  
Calliope-mini-Board**



**dpunkt.verlag**

Nadine Bergner • Patrick Franken • Julia Kleeberger • Thimo Leonhardt •  
Mario Lukas • Mario Pesch • Natalia Prost • Jan Thar • Lina Wassong

Lektorat: Volker Bombien, Boris Karnikowski  
Lektoratsassistentin: Stefanie Weidner  
Copy-Editing: Dr. Dorothee Leidig  
Satz: Frank Heidt  
Herstellung: Susanne Bröckelmann  
Umschlaggestaltung: Kraus, [www.exclam.de](http://www.exclam.de),  
unter Verwendung eines Artworks von [urbn pockets \(urbn-pockets.com\)](http://urbn-pockets.com)  
Druck und Bindung: Grafisches Centrum Cuno GmbH & Co. KG, 39240 Calbe (Saale)

Bibliografische Information der Deutschen Nationalbibliothek  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;  
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:  
Print 978-3-86490-468-4  
PDF 978-3-96088-263-3  
ePub 978-3-96088-264-0  
mobi 978-3-96088-265-7

1. Auflage 2017

dpunkt.verlag GmbH  
Wieblinger Weg 17  
69123 Heidelberg

Die vorliegende gedruckte Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten.  
Die Verwendung der hierin verwendeten Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.  
Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.  
Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0



# Vorwort



## • Liebe Leserinnen und Leser,



dieses Buch ist nicht nur die allererste Sammlung über und für den Calliope mini, sondern noch dazu eine ganz besondere.

Denn erfahrene Maker, Bastlerinnen, Autorinnen und Autoren haben dafür gemeinsam mit Didaktik-Fachleuten über 25 Projekte von »einfach« bis »komplex« untereinander abgestimmt, beschrieben und zum Schluss in einem sinnvollen Aufbau versammelt, der Eltern, Lehrkräfte und natürlich die jungen Calliope-Fans von Einsteigern bis Fortgeschrittenen eine ganze Weile lang beschäftigen kann. Und damit das auch möglichst umfassend passieren kann, stehen die Inhalte des Buchs unter einer freien Creative-Commons-Lizenz. Was nichts anders bedeutet, als dass die Arbeit an den Projekten in Schulen, an Ausbildungsplätzen und allen anderen Einsatzorten problemlos auch mit Kopien der Seiten möglich ist. Und dass es ebenso erlaubt ist, auf den beschriebenen Projekten aufzubauen, sie weiter zu entwickeln und die Ergebnisse wiederum der Community zukommen zu lassen (mehr dazu auf der Verlagswebsite unter <http://bit.ly/2pYTA8G>).

Genauso wie beim gesamten Calliope-Projekt steht nämlich eben dieser Aspekt der Gemeinsamkeit auch beim ersten Calliope-Buch im Vordergrund. Nicht nur bei seiner Entstehung, sondern vor allem bei der Nutzung. Die Macherinnen und Macher haben ein Werk nicht für das Regal, sondern für die Praxis vorgelegt mit dem Ziel, den Spaß, den sie selbst mit dem 5-Gramm-Minicomputer haben, mit anderen zu teilen und sie zu animieren, es ihnen gleich zu tun.

Bei aller Ernsthaftigkeit, die das Thema »Digitalisierung« speziell in den Schulen mit sich bringt, wird dieser Spaß schließlich oft vergessen, obwohl es doch eine der bekanntesten didaktischen Lehren ist, dass wir mit Spaß viel besser lernen. Und Spaß macht der Calliope mini noch einmal mehr mit den praxiserprobten Anleitungen aus diesem Buch, mit dem das kleine elektronische Wunderwerk zu einem digitalen Experimentierbaukasten wird, der die sonst so

verborgene Welt der digitalen Technologien erfahrbar, erlebbar und damit begreifbar macht. An Schulen oder auch zuhause. Und das ist bitter nötig und überfällig.

Denn während vielerorts hierzulande noch darüber gegrübelt und auch gestritten wird, ab welchem Alter Schulkinder mit digitalen Medien und Geräten in Berührung kommen sollten, ab wann also (und ob überhaupt!) die Lehre über das Digitale nötig wäre, ist die Welt um uns herum längst in einem fortgeschrittenen digitalen Zeitalter angekommen, das über Desktop-Rechner, Laptops und Smartphones weit hinaus geht. Sie ist in einem Zeitalter angekommen, in dem grenzenlose Vernetzung, Robotik, Sprach- und Stimmerkennung, virtuelle Realität, künstliche Intelligenz, Massendatenverarbeitung und automatisierte Verfahren rund um die Personenerkennung und -bewertung unseren Alltag, unser Leben und sogar unser Verhalten bestimmen.

Nicht jeder Mensch muss programmieren können. Doch jeder Mensch sollte über ein Grundwissen darüber verfügen, wie digitale Technologien funktionieren, wie sie gesteuert werden, was sie können – und was (noch) nicht. Kein Beruf der nahen Zukunft wird ohne diese Grundkenntnisse auskommen, an keiner Branche gehen die Entwicklungen vorbei.

Die junge Generation (die davon noch viel mehr betroffen ist und beeinflusst wird als wir Erwachsene, und die nicht nur in einer digitalen Zukunft leben wird, sondern sie auch zum Wohle aller gestalten soll) über diese technologischen und gesellschaftlichen Entwicklungen nicht zu informieren, sie nicht dazu auszubilden, wäre ein grober Fehler. Einer, den das Projekt Calliope mini verhindern möchte. Indem es speziell den jungen, natürlich aber auch erwachsenen Menschen digitale Technologie mit viel Spaß und Experimentierfreude beibringt.

Das erste Calliope-Buch hilft dabei. Als Bastel-Anleitung für einen für jede Person erschwinglichen Kleinstcomputer, der vor nicht allzu vielen Jahren noch Tausende von Euro gekostet hätte, als Ideen-Vorlage für eigene Projekte und als Lehrbuch zum Kennenlernen der digitalen Welt, in der wir leben.

Wir wünschen viel Spaß dabei!

*Tanja Haeusler, Johnny Haeusler*

Autorin und Autor des Elternratgebers »Netzgemüse«, Gründerin und Gründer der TINCON, der ersten Jugendkonferenz rund um die digitale Gesellschaft.



# Wie dieses Buch entstand

Im Januar 2017 lud ich erfahrene Elektronik-Bastler und Leute aus der Bildungsarbeit zu einem Wochenende nach Lemiers/Holland ein. Ziel des Wochenendes war, ein Buch zum Calliope mini zu planen und es dann in den Folgewochen zu schreiben. Das Ergebnis liegt nun gedruckt vor.

Das Calliope mini-Board und die hinter ihm stehende gemeinnützige Calliope-GmbH tritt mit dem Ziel an, den Informatikunterricht an deutschen Schulen zu verbessern und attraktiver zu gestalten. Bundesweit sollen zukünftig alle Drittklässler mit dem Calliope-Board programmieren lernen. Die Autorinnen und Autoren dieses Buches wissen um die große Bedeutung von Programmierkenntnissen, um die Gesellschaft zu verstehen und zu gestalten. Deshalb haben sich alle Autoren gern an diesem Buchprojekt beteiligt.

Das vorliegende Buch soll zu einem spielerischen Einstieg in das Programmieren mit dem Calliope mini-Board verhelfen. Die Bastelprojekte beginnen sehr einfach, werden dann aber immer komplexer. Der Bastelspaß steht dabei im Vordergrund; die Vermittlung von Programmiergrundkenntnissen und technisches Grundlagenwissen werden an den Stellen vermittelt, an denen sie gebraucht werden.

Ein Hemmnis für die intensive Nutzung von Lernmaterialien ist häufig eine eingeschränkte Nutzungslizenz. Deshalb war es ein besonderes Anliegen der Autoren, die in diesem Buch vorliegenden Texte und Abbildungen unter eine freie Nutzungslizenz (<https://creativecommons.org/licenses/by-nc-sa/3.0/de/>) zu stellen. Die Autoren und ich hoffen, dass viele Lehrer und sonstige Bildungsarbeiter dadurch zusätzlich angeregt werden, das Calliope mini-Board im Unterricht einzusetzen (mehr dazu auf der Verlagswebsite unter <http://bit.ly/2pYTA8G>).

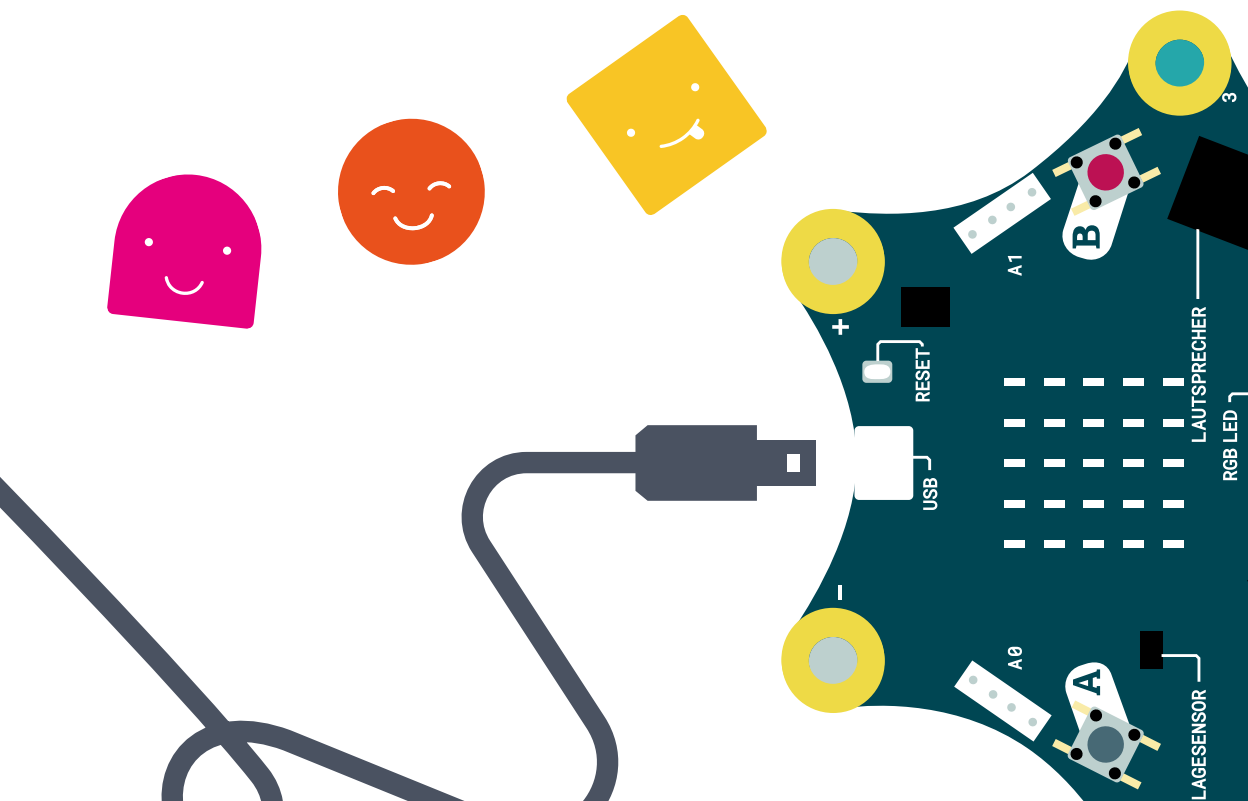
Dem dpunkt.verlag gebührt Dank dafür, dass er ohne Zögern einer freien Nutzung seines Buchtextes zugestimmt hat. Ein weiterer Dank geht an die Make Light Initiative des BMBF für ihre materielle und ideelle Unterstützung dieses Buchprojekts.

Der größte Dank geht an die Autoren und Autoren, die sich begeistert in dieses Buchprojekt gestürzt haben und innerhalb weniger Wochen faszinierende Calliope-Bastelprojekte gezaubert haben.

Volker Bombien, im Juni 2017

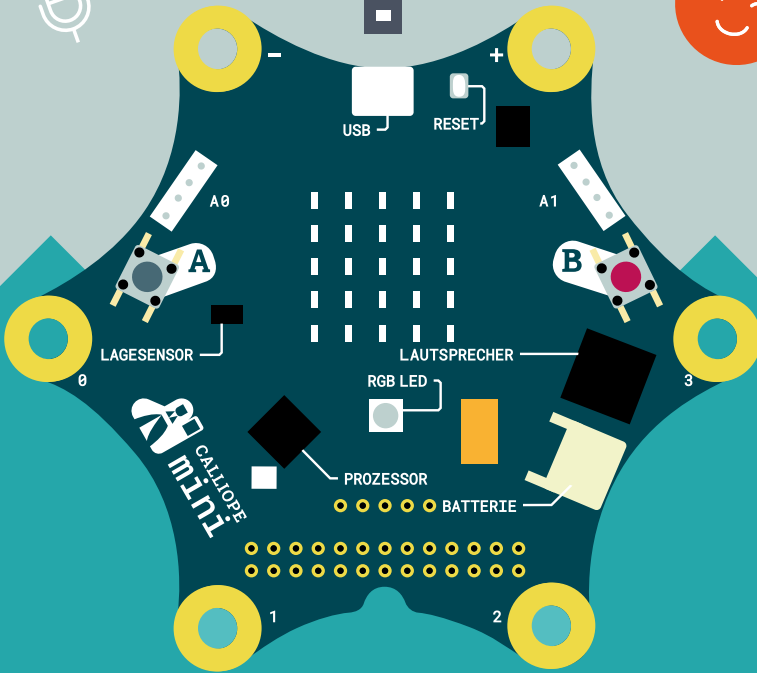
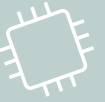
# Inhaltsübersicht

Kapitel 1:	Es geht los ...	1
Kapitel 2:	Erste Programme ...	9
Kapitel 3:	Mit dem Calliope mini interagieren ...	31
Kapitel 4:	Die Sensoren beim Calliope mini ...	49
Kapitel 5:	Dein rechter, rechter Platz ist leer ...	77
Kapitel 6:	Würfeln ohne Würfel ...	105
Kapitel 7:	Es wird logisch ...	149
Kapitel 8:	Mapping – so hinein und anders wieder hinaus ...	177
Kapitel 9:	Drahtlose Kommunikation ...	219
Kapitel 10:	Jetzt kommt Bewegung ins Spiel ...	267
Anhang .....		299
Über die Autoren .....		301
Index .....		303





# CALLIOPE mini



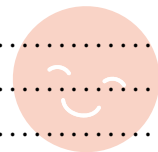
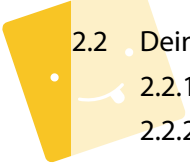
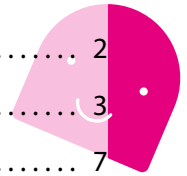
# Inhaltsverzeichnis

## Kapitel 1: Es geht los ... 1

- 1.1 Was dich in diesem Buch erwartet ..... 2
- 1.2 Du lernst den Calliope mini kennen ..... 3
- 1.3 Calliope mini erwacht zum Leben ..... 7

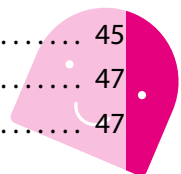
## Kapitel 2: Erste Programme ... 9

- 2.1 So wirst du zum Programmierer oder zur Programmiererin ..... 10
- 2.2 Dein erstes Programm »Calliope mini lacht« ..... 11
  - 2.2.1 Der PXT-Editor ..... 11
  - 2.2.2 Erste Befehle kennenlernen ..... 13
  - 2.2.3 Übertragen des Programms auf den Calliope mini ..... 16
- 2.3 Erstes eigenes Programm »Deinen Namen schreiben« ..... 20
  - 2.3.1 Löschen von Befehlen und Programmen ..... 21
  - 2.3.2 Dauerschleife anlegen ..... 21
  - 2.3.3 Text als Laufschrift ausgeben ..... 22
  - 2.3.4 Programm testen und erweitern ..... 24
- 2.4 Calliope mini macht Musik ..... 25
  - 2.4.1 Töne abspielen ..... 25
  - 2.4.2 Lieder komponieren ..... 26
- 2.5 Programme umbenennen, speichern und weiter bearbeiten ..... 27
  - 2.5.1 Programme benennen und speichern ..... 27
  - 2.5.2 Programme importieren und weiter bearbeiten ..... 28



**Kapitel 3: Mit dem Calliope mini interagieren 31**

- 3.1 Wenn, dann, was? ..... 32
- 3.2 Ein Klavier mit dem Calliope mini ..... 35
  - 3.2.1 Welche Materialien und Werkzeuge werden benötigt? ..... 35
  - 3.2.2 Die Vorbereitung ..... 36
  - 3.2.3 Das Basteln ..... 36
  - 3.2.4 Die Programmierung ..... 38
  - 3.2.5 Klavier spielen! ..... 39
  - 3.2.6 Weitere Ideen ..... 39
- 3.3 Angelspiel ..... 39
  - 3.3.1 Benötigte Materialien ..... 40
  - 3.3.2 Aquarium, Angel und Fische bauen ..... 42
  - 3.3.3 Mit dem Calliope mini verbinden ..... 45
  - 3.3.4 Die Programmierung ..... 47
  - 3.3.5 Andere Stromkreisspiele ..... 47

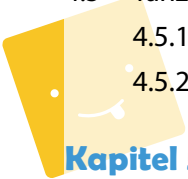


**Kapitel 4: Die Sensoren beim Calliope mini 49**

- 4.1 Was sind eigentlich Sensoren? ..... 50
  - 4.1.1 Der Temperatursensor ..... 51
  - 4.1.2 Der Beschleunigungs- und Bewegungssensor ..... 51
  - 4.1.3 Der Helligkeitssensor ..... 52
- 4.2 Die Wasserwaage ..... 53
  - 4.2.1 Die x-Achse wird gemessen ..... 54
  - 4.2.2 Die y-Achse wird gemessen ..... 56
- 4.3 Die Calliope-Alarmanlage ..... 57
  - 4.3.1 Alles, was du über die Calliope-Alarmanlage wissen musst .. 57
  - 4.3.2 Materialien und Werkzeuge ..... 57
  - 4.3.3 Wie funktioniert die Calliope-Alarmanlage? ..... 58
  - 4.3.4 Der Lichtsensor ..... 58
  - 4.3.5 ... und jetzt zum Programmieren ..... 59
  - 4.3.6 Wie soll sich deine Alarmanlage einschalten? ..... 59
  - 4.3.7 Die Schleife programmieren ..... 60

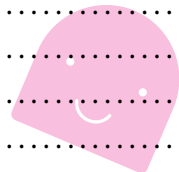


- 4.3.8 Wann ist es hell? Wann ist es dunkel? ..... 60
- 4.3.9 Den Alarmton programmieren ..... 61
- 4.3.10 Die Lichtstärke im Simulator ..... 62
- 4.3.11 Geschafft! ..... 63
- 4.4 Das Farbthermometer ..... 65
  - 4.4.1 Farbe und Temperatur? ..... 65
  - 4.4.2 Calliope mini unter der Lupe ..... 65
  - 4.4.3 Die Programmierung ..... 66
  - 4.4.4 Erweiterungen ..... 69
  - 4.4.5 Was du gelernt hast ..... 69
- 4.5 Tanzender Teddy ..... 70
  - 4.5.1 Die Programmierung ..... 70
  - 4.5.2 Der Aufbau des Teddys ..... 72



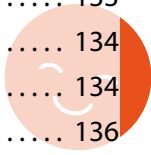
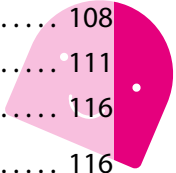
**Kapitel 5: Dein rechter, rechter Platz ist leer 77**

- 5.1 Variable »Platzhalter« ..... 78
- 5.2 Der Klickzähler ..... 79
  - 5.2.1 Platzhalter erstellen ..... 79
  - 5.2.2 Platzhalter anzeigen ..... 81
  - 5.2.3 Platzhalter mit Knopfdruck verändern ..... 84
  - 5.2.4 Anzeige beim Zählen verbessern ..... 85
  - 5.2.5 Von vorne zählen ..... 86
  - 5.2.6 Noch mehr zählen ..... 87
- 5.3 Die digitale Sanduhr ..... 88
  - 5.3.1 Platzhalter erstellen ..... 88
  - 5.3.2 Sanduhr startbereit machen ..... 89
  - 5.3.3 Eine einzelne LED ausschalten ..... 90
  - 5.3.4 Ganze Reihe mit Knopfdruck ausschalten ..... 91
  - 5.3.5 Jede Reihe nacheinander ausschalten ..... 93
  - 5.3.6 Musik abspielen ..... 95
  - 5.3.7 Sanduhr starten ..... 96
  - 5.3.8 Reihenfolge verändern ..... 97
- 5.4 Das Metronom ..... 98



**Kapitel 6: Würfeln ohne Würfel**

- 6.1 Zufallszahlen ..... 106
- 6.2 Würfeln mit Schwung ... 107
  - 6.2.1 Dein Calliope wird zu einem einfachen Würfel ..... 108
  - 6.2.2 Calliope-Schüttelwürfel mit Augenzahlen ..... 111
  - 6.2.3 Was ist noch möglich? ..... 116
- 6.3 Symbolschnapper ..... 116
  - 6.3.1 Symbolplättchen ausmalen ..... 117
  - 6.3.2 Beim Schütteln ein bestimmtes Symbol anzeigen..... 118
  - 6.3.3 Alle LED-Symbole malen..... 120
  - 6.3.4 Zufälliges Symbol anzeigen ..... 122
  - 6.3.5 Punkte zählen ..... 123
  - 6.3.6 Noch mehr Zufall..... 125
- 6.4 Das Reaktionsspiel ..... 126
  - 6.4.1 Wer ist die oder der Schnellste? ..... 126
  - 6.4.2 Symbol nach zufälliger Zeit anzeigen..... 127
  - 6.4.3 Platzhalter speichert, ob »Smiley an« ..... 128
  - 6.4.4 Auf Knopfdruck reagieren..... 129
  - 6.4.5 Das Smiley muss verschwinden ..... 129
  - 6.4.6 Punkte zählen ..... 131
  - 6.4.7 Gegenspieler B kommt ins Spiel..... 131
  - 6.4.8 Punkte anzeigen..... 132
  - 6.4.9 Erweiterungen ..... 132
- 6.5 Farben-Bingo ..... 133
  - 6.5.1 Was ist denn Farben-Bingo?..... 133
  - 6.5.2 Materialien und Werkzeuge ..... 134
  - 6.5.3 Wie funktioniert Bingo?..... 134
  - 6.5.4 Das Farben-Bingo-Programm ..... 136
  - 6.5.5 Auslösen des Farbwechsels ..... 136
  - 6.5.6 Was passiert zwischen den Farbwechseln? ..... 136
  - 6.5.7 Die zufällige Farbausgabe programmieren ..... 138
  - 6.5.8 Farben einfügen..... 139
  - 6.5.9 Erweiterung des Programms ..... 143



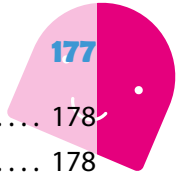
6.6	Das Wortschatzspiel.....	144
6.6.1	Wie viele Wörter kennst du?.....	144
6.6.2	Die Vorbereitung .....	144
6.6.3	Calliope mini unter der Lupe .....	144
6.6.4	Die Programmierung .....	145
6.6.5	Der Start .....	146
6.6.6	Das Erstellen der Buchstaben .....	146
6.6.7	Das Startsignal .....	146
6.6.8	Der letzte Schritt .....	147
6.6.9	Idee für den Schulunterricht .....	148

**Kapitel 7: Es wird logisch ...**

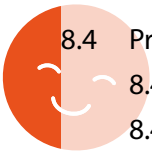
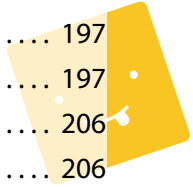
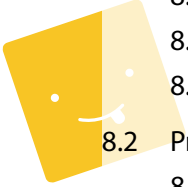
**149**

7.1	Würfelglück.....	154
7.1.1	Platzhalter steuert den Spielverlauf .....	155
7.1.2	Grundstruktur jeder Spielphase .....	156
7.1.3	Spielphase 1 – Spieler A würfelt .....	156
7.1.4	Das Würfeln .....	157
7.1.5	Spielphase 2 – Spieler B würfelt .....	158
7.1.6	Das Ergebnis .....	159
7.1.7	Nach dem Spiel ist vor dem Spiel.....	160
7.1.8	Erweiterungsmöglichkeiten.....	161
7.1.9	Unfares Spiel.....	164
7.2	Tischbillard .....	165
7.2.1	Materialien und Werkzeuge .....	165
7.2.2	Wie funktioniert Tischbillard? .....	166
7.2.3	Die Programmierung .....	166
7.2.4	Der Aufbau des Spiels .....	170
7.2.5	Erweiterung des Programms .....	175

## Kapitel 8: Mapping – so hinein und anders wieder hinaus

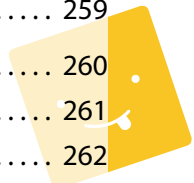
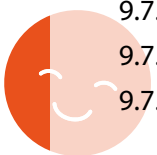
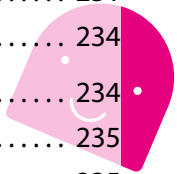
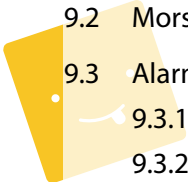


8.1	Mapping verstehen .....	178
8.1.1	Worum geht es hier?.....	178
8.1.2	Berechnung des »verteile«-Programmierbefehl.....	182
8.1.3	Sensoren und ihre Werte.....	183
8.2	Projekt »Farbverlauf«.....	184
8.2.1	Farbverlauf – Sensoren durch Farben sichtbar machen.....	184
8.2.2	Was brauchst du für dieses Projekt? .....	184
8.2.3	Alles noch einmal überprüfen, bevor es losgeht!.....	185
8.2.4	Es geht los.....	185
8.2.5	Testen.....	194
8.2.6	Experimentieren.....	194
8.2.7	Da geht noch mehr?! – Optimierte dein Projekt.....	195
8.3	Projekt »Luftgitarre«.....	196
8.3.1	Luftgitarre – Bewegung und Musik verbinden .....	196
8.3.2	Was brauchst du für dieses Projekt? .....	197
8.3.3	Alles noch einmal überprüfen, bevor es losgeht!.....	197
8.3.4	Es geht los.....	197
8.3.5	Testen.....	206
8.3.6	Basteln .....	206
8.3.7	Da geht noch mehr?! – Optimierte dein Projekt.....	207
8.4	Projekt »Fang den Dot«.....	208
8.4.1	»Fang den Dot« – Ein Geschicklichkeitsspiel .....	208
8.4.2	Alles noch einmal überprüfen, bevor es losgeht!.....	208
8.4.3	Es geht los.....	209
8.4.4	Testen.....	217
8.4.5	Da geht noch mehr?! – Optimierte dein Projekt.....	218



**Kapitel 9: Drahtlose Kommunikation** **219**

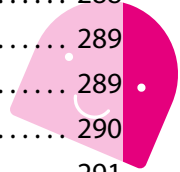
- 9.1 Die Funktionsweise von Funk..... 220
- 9.2 Morsen mit mehreren Calliope minis ..... 222
- 9.3 Alarmanlage vernetzt ..... 227
  - 9.3.1 Aufbau ..... 228
  - 9.3.2 Alarmanlage ..... 228
  - 9.3.3 Fernbedienung ..... 231
  - 9.3.4 Erweiterungen ..... 234
- 9.4 Hau den Lukas ..... 234
  - 9.4.1 Grundprinzip..... 235
  - 9.4.2 Die Programmierung ..... 235
  - 9.4.3 Verbesserte Punkteanzeige ..... 240
  - 9.4.4 Was man noch so machen könnte..... 241
- 9.5 Das Roboterballett ..... 241
  - 9.5.1 Bewegungsanweisungen senden ..... 242
  - 9.5.2 Bewegungsanweisungen empfangen und auswerten..... 244
  - 9.5.3 Einen einzigen Sender-Calliope bestimmen ..... 247
  - 9.5.4 Erweiterungen ..... 250
- 9.6 Eckstein, Eckstein ... alles muss versteckt sein! ..... 250
  - 9.6.1 Das Programm für den versteckten Calliope ..... 252
  - 9.6.2 Das Programm für den suchenden Calliope..... 254
  - 9.6.3 Was ist noch möglich? ..... 259
- 9.7 Theremin ..... 260
  - 9.7.1 Los geht's ..... 261
  - 9.7.2 Ein Calliope mini als Kopfhörer ..... 262
  - 9.7.3 Es wird lauter – Anschluss eines externen Lautsprechers ... 263
  - 9.7.4 Erweiterungen und Änderungen..... 265



**Kapitel 10: Jetzt kommt Bewegung ins Spiel ... 267**



- 10.1 Wie funktioniert ein Elektromotor? ..... 268
  - 10.1.1 Wie funktioniert ein Servomotor?..... 270
  - 10.1.2 Motoren mit dem Calliope mini ansteuern ..... 271
  - 10.1.3 Servomotoren mit dem Calliope ansteuern ..... 274
- 10.2 Der Calliope-Tee-Roboter ..... 276
- 10.3 Das lichtscheue Borstentier..... 285
  - 10.3.1 Materialien und Werkzeuge ..... 286
  - 10.3.2 Wie funktioniert das Borstentier? ..... 287
- 10.4 Der Vibrationsmotor ..... 287
  - 10.4.1 ... und jetzt zur Programmierung deines Borstentiers ..... 288
  - 10.4.2 Zu Beginn ..... 288
  - 10.4.3 Die Logik im Spiel ..... 289
  - 10.4.4 Die Helligkeitswerte vergleichen ..... 289
  - 10.4.5 ... und jetzt zu den Motoren..... 290
  - 10.4.6 Und wie schläft das Borstentier wieder ein? ..... 291
  - 10.4.7 Das Borstentier basteln ..... 292
  - 10.4.8 Anschließen der Motoren ..... 292
  - 10.4.9 Befestigen der Motoren und der Batteriebox..... 294
  - 10.4.10 Zahnbürstenköpfe anbringen ..... 295
  - 10.4.11 Den Calliope in ein Borstentier verwandeln ..... 296
  - 10.4.12 Was ist noch möglich?..... 297



**Anhang 299**

- A.1 Calliope mini, 1. Generation ..... 299
- A.2 Calliope mini, 2. Generation ..... 300
- Über die Autoren ..... 301
- Index ..... 303

# 1

## Es geht los ...

In diesem Kapitel lernst du deinen Calliope mini näher kennen. Dabei wirst du erfahren, was der Calliope mini genau ist und was du mit ihm machen kannst.

Das einführende Kapitel wurde von Nadine Bergner geschrieben.

## 1.1 Was dich in diesem Buch erwartet

Du hast einen Calliope mini und fragst dich, was du damit tun kannst? Du hast schon Ideen und möchtest gerne deinen Calliope mini selbst programmieren?

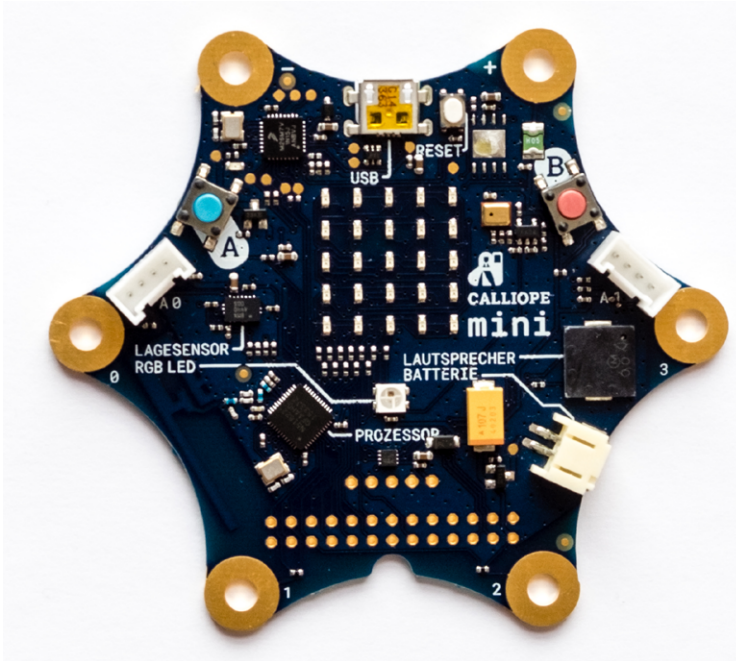


Abb. 1-1 Der Calliope mini und seine Bauteile

Herzlichen Glückwunsch, dann hältst du genau das richtige Buch in deinen Händen. In diesem Buch dreht sich alles um den Calliope mini, der dir im nächsten Abschnitt auch gleich noch genauer vorgestellt wird. Nachdem du erfahren hast, was der Calliope mini alles kann, lernst du Schritt für Schritt, wie du auch eigene Programme für den Calliope mini schreiben und mit ihm tolle Projektideen umsetzen kannst. In den späteren Kapiteln findest du viele ganz unterschiedliche Projekte, zu denen bereits eine Anleitung vorbereitet ist. Diese Projekte kannst du jederzeit mit deinen eigenen Ideen weiterentwickeln oder auch selbst neue Projekte erfinden.

Die vielen (hoffentlich) spannenden Projekte sind nach Schwierigkeit angeordnet. Wenn du also noch keine Vorerfahrung mit dem Calliope mini oder mit Programmierung hast, empfiehlt es sich, vorne anzufangen und sich zu den größeren Herausforderungen im hinteren Teil des Buches vorzuarbeiten. In diesem Buch findest du ganz unterschiedliche Projekte, die verschiedene Funktionalitäten deines Calliope mini nutzen.

Da du ganz viele neue Dinge kennenlernen wirst, findest du an einigen Stellen wichtige Merksätze wie zum Beispiel diesen hier:

## Programmieren

So nennt der Profi das Erstellen von Programmen. Auch du wirst im Laufe dieses Buches viele Programme für deinen Calliope mini programmieren. Dadurch wirst auch du zur Programmiererin oder zum Programmierer.

Wenn du neugierig bist oder auch schon ein wenig Vorerfahrung hast, kannst du gerne auch schon zu den komplexeren Projektideen im hinteren Teil des Buches vorblättern. Falls du bei der Umsetzung merkst, dass du eine bestimmte Sache nicht kennst, kannst du den Index (eine Art Stichwortverzeichnis ganz am Ende des Buches) benutzen, um die Stelle zu finden, an der diese Sache erklärt wird.

Jetzt wird es aber Zeit, dass du loslegst und dir deinen Calliope mini einmal ganz genau anschaust.

Ganz viel Spaß bei deinem ersten und allen weiteren Calliope mini-Programmen!

## 1.2 Du lernst den Calliope mini kennen

In diesem Kapitel lernst du deinen Calliope mini näher kennen. Dabei wirst du erfahren, was der Calliope mini genau ist, was du mit ihm machen kannst und was das Besondere an ihm ist.

Der Calliope mini ist eine Platine, auf der ein *Mikroprozessor* (praktisch das Hirn des Calliope mini) und zahlreiche weitere spannende *Bauteile* angebracht (gelötet) sind. Welche Bauteile es genau gibt und was du damit tun kannst, erfährst du ein paar Zeilen weiter unten.

Du kannst den Calliope mini später auch selbst programmieren, das heißt eigene Programme schreiben und auf den Calliope mini übertragen, sodass du ihm neue Sachen beibringen kannst. Keine Sorge, falls das für dich alles ganz neu ist. Es ist gar nicht so schwierig, und du findest auf den nächsten Seiten eine genaue Anleitung. Falls du dennoch Fragen hast oder etwas nicht funktioniert, frag doch deine Eltern oder älteren Geschwister um Hilfe.

Damit du erste Ideen bekommst, was du mit dem Calliope mini machen kannst, hier ein paar Vorschläge: Du kannst ...

- ▶ ... deinen Calliope mini zum Lächeln bringen oder auch traurig gucken lassen.
- ▶ ... den Calliope mini verschiedene Töne oder sogar ein ganzes Lied spielen lassen.
- ▶ ... mit dem Calliope mini ein kleines Spiel programmieren, indem du und eine Freundin oder ein Freund gegeneinander Würfeln können.
- ▶ ... eine eigene Alarmanlage für dein Zimmer bauen.
- ▶ ... noch ganz viele andere Projekte umsetzen, da fällt dir doch sicher noch einiges ein.

Damit du einschätzen kannst, welche Möglichkeiten du mit dem Calliope mini hast, erfährst du nun, wozu die verschiedenen Bauteile auf dem Calliope mini da sind. Im Anhang (Seite 299) findest du eine detaillierte Darstellung vom Calliope-mini-Board mit all seinen Anschlüssen.

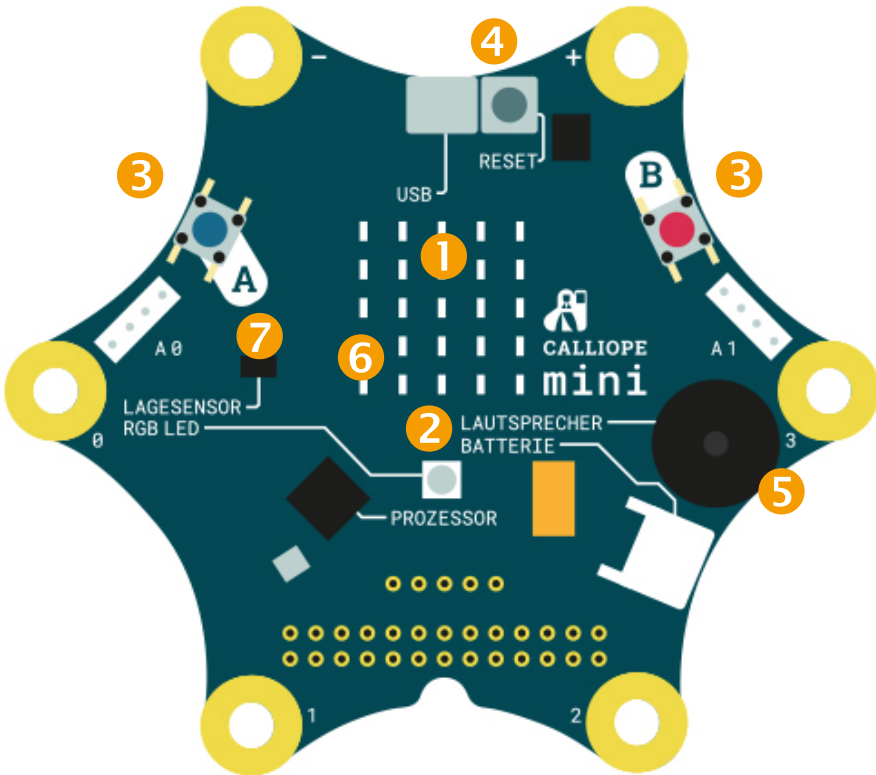


Abb. 1-2 Schematische Zeichnung des Calliope mini

- 1 Das *LED-Display*: In der Mitte des Calliope mini siehst du das LED-Display. LED ist die Abkürzung für den englischen Ausdruck »light-emitting diode«, was auf Deutsch »lichtaussendendes Bauteil« oder einfach ausgedrückt »kleine Lampe« bedeutet. Auf dem Calliope mini sind insgesamt 25 winzige rote LEDs in einem Rechteck angeordnet. Du kannst sie alle einzeln ansteuern und so zum Beispiel einen lachenden Smiley oder auch einen beliebigen Buchstaben anzeigen lassen.
- 2 Die *RGB-LED*: Die RGB-LED ist eine ganz besondere LED. Sie ist etwas größer und unterhalb des LED-Displays zu finden. Das Besondere ist aber, dass sie nicht nur rot, sondern in ganz vielen verschiedenen Farben leuchten kann. Die Abkürzung RGB steht für Rot, Grün und Blau. Aus diesen drei Grundfarben lassen sich nämlich alle anderen Farben mischen, ganz ähnlich wie bei einem Wasserfarbkasten.
- 3 Die *Knöpfe A und B*: Den blauen Knopf A (links auf dem Calliope mini) und den roten Knopf B (rechts) kannst du herunterdrücken. Dies merkt der Calliope mini, und du kannst ihn später so programmieren, dass der Calliope mini etwas Bestimmtes macht, zum Beispiel einen Ton abspielt, wenn ein Knopf gedrückt wird. Alle Bauteile, mit denen man dem Calliope mini ein Signal geben kann (z. B. das Signal »Knopf A wurde gedrückt«), nennt man Sensoren. Davon hat der Calliope mini einige verbaut, damit du mit ihm interagieren kannst.

## Sensoren

Sensoren sind Bauteile, die bestimmte Veränderungen der Umwelt messen. Das kann ein Knopf sein, der von dir heruntergedrückt wird, oder auch ein Temperatursensor, der registriert, ob es im Raum wärmer oder kälter wird.

- 4 Der *Reset-Knopf*: Wenn du genau hinschaust, erkennst du noch einen dritten Knopf, der mit »Reset« beschriftet ist. Er befindet sich oben in der Mitte. Das englische Wort »Reset« bedeutet auf Deutsch »zurücksetzen«. Wenn du also diesen Knopf drückst, wird das aktuelle Programm zurückgesetzt, also neu gestartet.
- 5 Der *Lautsprecher*: Der schwarze viereckige Kasten rechts unten ist ein Lautsprecher. Er kann verschiedene Töne abspielen, sodass du sogar ein kleines Lied komponieren kannst.
- 6 Der *Lichtsensor*: Fast unsichtbar versteckt sich dieser Sensor im LED-Display. Er misst die Lichtstärke (also die Helligkeit). Der Calliope mini kann also feststellen, ob es gerade hell oder dunkel ist. Auch wenn du deine Hand nah

über den Calliope mini hältst, wird es dunkler. So kannst du den Lichtsensor nutzen, um zu messen, ob jemand am Calliope mini vorbeigeht, und damit zum Beispiel eine Alarmanlage bauen.

- 7 Der *Lagesensor*: Der Lagesensor ist ein winziges Bauteil, das aber etwas ganz Tolles kann. Durch den Lagesensor kann der Calliope mini merken, wie herum er gehalten wird. Du könntest also ein Programm schreiben, sodass der Calliope mini piepst, wenn man ihn auf den Kopf dreht. Dieser Sensor misst auch ob sich der Calliope mini bewegt, also zum Beispiel geschüttelt wird.

Damit du später deine neuen Programme auf den Calliope mini übertragen kannst, gibt es noch einen *Mikro-USB-Anschluss* oben in der Mitte. Mit einem USB-Kabel kannst du den Calliope mini mit deinem Computer oder Laptop verbinden. Über das Kabel wird der Calliope mini gleichzeitig mit Strom versorgt. Du brauchst also zum Austesten deiner Programme keine Batterie, solange der Calliope mini am Computer angeschlossen ist. Wenn du den Calliope mini später auch allein benutzen möchtest, brauchst du ein *Battery-Pack*, das du dann an den *Batterieanschluss* ansteckst.

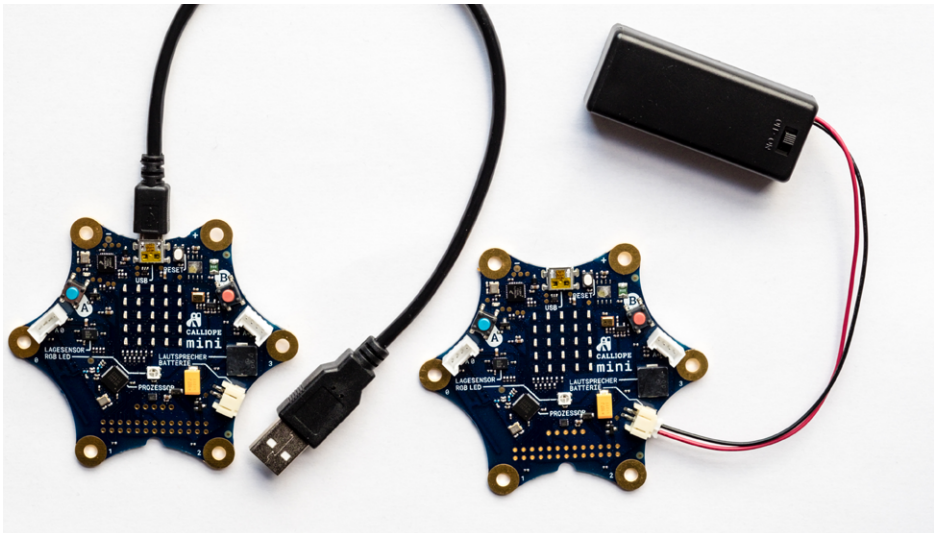


Abb. 1-3 Calliope mini mit USB-Kabel und Battery-Pack

## 1.3 Calliope mini erwacht zum Leben

Damit der Calliope mini zum Leben erwacht, musst du nun das andere Ende des USB-Kabels in den USB-Anschluss eines Computers stecken.

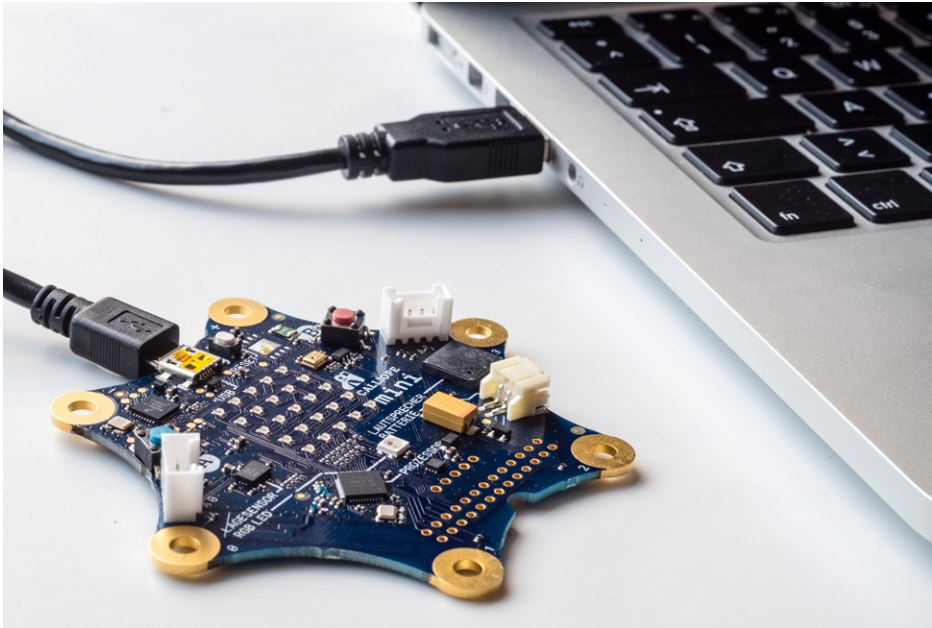


Abb. 1–4 Calliope mini mittels USB-Kabel am Computer angeschlossen

Auf dem Calliope mini wird immer das zuletzt aufgespielte Programm gespeichert. Wenn dein Calliope mini ganz neu ist und noch nicht vorher benutzt wurde, startet er nun ein Programm, das die Hersteller für dich aufgespielt haben. Mit diesem Programm kannst du die verschiedenen Sensoren ausprobieren.

Super, dein Calliope mini lebt! Damit er nicht nur lebt, sondern auch schick ist und sicher transportiert werden kann, wäre es eine gute Idee, wenn du dir eine Schutzhülle für deinen Calliope mini basteln würdest. Vielleicht hat du Lust, dir sogar eine richtige Tasche dafür zu nähen?! Oder – wenn dir Nähen nicht so liegt – dann könntest du auch eine Schachtel aus dicker Pappe oder aus Holz dafür verwenden. Deiner Fantasie sind dabei keine Grenzen gesetzt.

Viel Spaß dabei – und im nächsten Kapitel geht's auch schon ran ans Programmieren.



# 2

## Erste Programme ...

In diesem Kapitel schreibst du deine ersten Programme. Du erfährst, wie du Programme erstellst, abspeicherst und auf dein Calliope-Board überträgst.

Dieses Kapitel wurde von Nadine Bergner geschrieben.

## 2.1 So wirst du zum Programmierer oder zur Programmiererin

Du hast schon erfahren, dass *Programmieren* bedeutet, ein Programm für ein technisches Gerät, also zum Beispiel einen Computer oder auch deinen Calliope mini, selber zu schreiben. Menschen, die programmieren, nennt man Programmierinnen oder Programmierer. In diesem Kapitel wirst du selbst zu einer Programmiererin oder einem Programmierer.

Beim Programmieren ist es wie beim Lesen, Schreiben oder Rechnen: Es geht nicht immer alles glatt und manchmal passieren Fehler. Das ist auch gar nicht schlimm und passiert sogar den Profis jeden Tag.

### Ein Tipp zum Programmieren

Fehler passieren jedem; und das ist auch gar nicht schlimm, denn wenn ein Programm einmal nicht das tut, was du möchtest, kannst du ja jederzeit ein neues Programm auf deinen Calliope mini aufspielen. Also nicht gleich aufgeben, wenn etwas nicht klappt. Starte einen neuen Versuch und probier es noch einmal.

Wenn du ein tolles Programm fertiggestellt hast, kannst du es *speichern*, sodass du es immer wieder benutzen und weiter verbessern kannst. Du kannst deine Programme auch mit anderen teilen und tauschen. Dazu speicherst du dein Programm auf deinem Computer ab und schickst es einer anderen Person per E-Mail oder überreichst es auf einem USB-Stick. So wirst du ganz schnell viele tolle Projekte umsetzen können.

Alles, was du zum Programmieren brauchst, ist:

- ▶ der Calliope mini,
- ▶ einen Computer oder Laptop mit Internetzugang,
- ▶ ein USB-Kabel, um deinen Calliope mini mit dem Computer zu verbinden.

## 2.2 Dein erstes Programm »Calliope mini lacht«

Jetzt geht es aber richtig los! Hier programmierst du dein erstes Programm für deinen Calliope mini.

### 2.2.1 Der PXT-Editor

Ein solches Computerprogramm schreibt man nicht in einem normalen Textverarbeitungsprogramm wie zum Beispiel Microsoft Word, sondern in einem *Editor*.

#### Editor und Programmiersprachen

Ein Editor bzw. eine Programmierumgebung ist ein Werkzeug, mit dem man Programme in einer Sprache schreiben kann, die auch technische Geräte wie Computer und der Calliope mini verstehen. Solche Sprachen nennt man Programmiersprachen.

Programme für den Calliope mini kannst du in verschiedenen Editoren schreiben. Alle haben verschiedene Vor- und Nachteile. Manche sind einfacher zu bedienen, haben aber weniger Möglichkeiten. In diesem Buch lernst du, wie du im *PXT-Editor* programmierst.

#### Editoren für Calliope

Eine Übersicht über die verfügbaren Editoren für den Calliope mini findest du unter <https://calliope.cc/editor>.

Beachte bitte, dass wir in diesem Buch alle Projekte mit dem Calliope mini durchführen, das rechts das Logo hat. Schau bitte in den Anhang (Seite 299), wenn du nicht genau sicher bist, welches Calliope-Board du benutzt.

#### Der PXT-Editor

Um mit dem Programmieren zu beginnen, besuche die Webseite <https://mini.pxt.io/>

Falls du noch nicht weißt, wie man eine Webseite aufruft, bitte deine Eltern oder einen anderen Erwachsenen um Hilfe.

Wenn du die Webseite aufrufst, also den Editor startest, siehst du bereits sehr viele Knöpfe und auch einen künstlichen Calliope mini. Damit du dich auf der Webseite zurechtfindest, erfährst du nun, wozu die ganzen Knöpfe da sind.



Abb. 2-1 Der PXT-Editor

Wenn du diese Webseite zum ersten Mal öffnest, erscheint unten rechts eine Meldung in einer schwarzen Box.

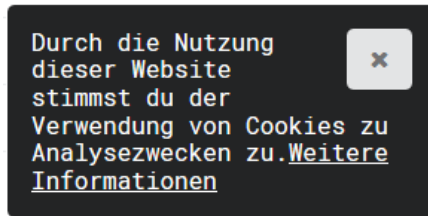


Abb. 2-2 Meldung zur Verwendung von Cookies

Diese Meldung zeigt dir an, dass die Webseite Informationen in einer Datei (ein sogenanntes Cookie) speichert. Diese Daten sind zum Beispiel Informationen über das Gerät, mit dem du die Webseite besuchst. Diese Daten werden vom Anbieter der Webseite genutzt, um Statistiken zu erstellen, zum Beispiel, wie viele Besucher der Webseite ein Windows-Betriebssystem nutzen. Für diese Webseite ist dies

nicht kritisch. Du musst zu keinem Zeitpunkt deinen Namen oder andere persönliche Daten eintragen. Somit kannst du die Meldung auf dem »X« in der oberen rechten Ecke wegklicken.

Jetzt kannst du dir den PXT-Editor erst einmal in Ruhe anschauen. Hier werden dir alle Bestandteile einzeln erklärt.

- ▶ Der *Simulator*: Den künstlichen Calliope mini links im Bild nennt man Simulator. Dort kannst du eine Vorschau auf dein Programm sehen, noch bevor du es auf den Calliope mini übertragen hast. Allerdings kann der Simulator nicht alles, was auch dein Calliope mini kann, zum Beispiel hat er keinen Kompass eingebaut.
- ▶ Die *Befehlsgruppen*: Die vielen bunten Schaltflächen in der Mitte stellen ein großes Menü dar. Alle Begriffe (zum Beispiel »Eingabe« oder »Funk«) stehen für eine Gruppe von Programmierbefehlen.

## (Programmier-)Befehl

Jedes Computerprogramm besteht aus wenigen oder auch ganz vielen Befehlen. Ein Befehl ist ein Baustein, der dem Programm sagt, was es tun soll. Der Calliope mini hört beispielsweise auf den Befehl »setze LED-Farbe auf Rot«.

## (Programmier-)Block

In manche Befehle kannst du weitere Blöcke einbauen. Solche Blöcke haben an der linken Seite eine Ausbuchtung wie ein Puzzleteil. Diese Blöcke kannst du nicht einzeln verwenden, du musst sie immer in einen Befehl einbauen.

- ▶ Die *Programmierfläche*: Auf der großen Fläche rechts erstellst du dein Programm, dies ist die Programmierfläche. Auf diese Programmierfläche ziehst du später alle Befehle, die dein Programm braucht, und setzt sie zu einem kompletten Programm zusammen.

## 2.2.2 Erste Befehle kennenlernen

Wenn du die Seite das erste Mal aufrufst, siehst du direkt ein erstes Beispielprogramm. Dieses Beispielprogramm besteht aus drei Programmierbefehlen, die wie Puzzleteile ineinandergreifen.

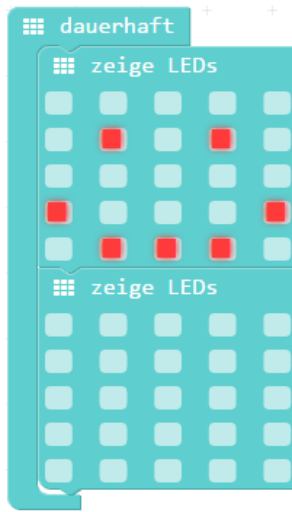


Abb. 2-3 Das erste Beispiel-Programm

Wenn du dir die drei Programmierbefehle einmal einzeln anschaust, hast du sicher schon eine Idee, was dieses Programm macht.

## Dauerschleife

Der oberste Befehl, der die anderen beiden umgibt, ist eine Schleife. Dies bedeutet aber nicht, dass sich der Calliope mini die Schuhe bindet, sondern dass ein bestimmtes Programmstück immer wieder ausgeführt wird, wie bei einer Schleife. Es gibt verschiedene Schleifen. Diese hier heißt »dauerhaft«, was bedeutet, dass das Programmstück unendlich lange ausgeführt wird. Es gibt auch Schleifen, die zum Beispiel nach einer gewissen Anzahl an Durchläufen automatisch enden. Eine Dauerschleife kannst du nur beenden, indem du das USB-Kabel herausziehst oder ein anderes Programm auf deinen Calliope mini spielst.



Abb. 2-4 Eine Dauerschleife

## LED-Display ansteuern

Der zweite und dritte Befehl sind eigentlich die gleichen, sie heißen »zeige LEDs« und dienen dazu, das LED-Display anzusteuern.

Wie du schon im Simulator links sehen kannst, sorgt dieses Programm dafür, dass auf dem Calliope mini ein Smiley blinkt.

Das Smiley blinkt, da der erste Befehl sagt, dass ein Smiley angezeigt werden soll. Also, dass genau die LEDs leuchten sollen, die ausgewählt wurden. Der zweite Befehl sagt aber, dass alle LEDs aus sein sollen. Und da die Dauerschleife diese beiden Befehle immer wieder abwechselnd aufruft, blinkt das Smiley.

Wenn du nun dem Smiley eine Nase hinzufügen möchtest, musst du einfach nur bei dem oberen Befehl die LED ganz in der Mitte anklicken.

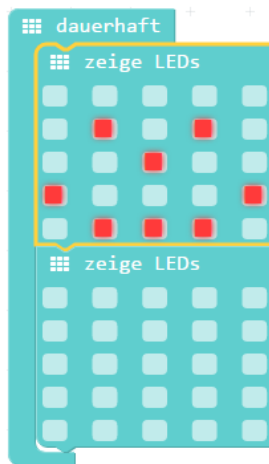


Abb. 2–5 Verändertes Programm: Smiley mit Nase

Sofort hat auch das blinkende Smiley im Simulator eine leuchtende Nase bekommen.

## 2.2.3 Übertragen des Programms auf den Calliope mini

Nun ist es an der Zeit, das Programm auf deinen echten Calliope mini zu übertragen. Das passiert in folgenden drei Schritten:

1. **Herunterladen** Du musst dein Programm herunterladen. Dazu drückst du auf den großen grünen Knopf unten links. Damit lädst du das Programm aus dem Internet auf deinen Computer herunter.

Gleichzeitig wird dir ein Fenster angezeigt, in dem dir erklärt wird, was die nächsten Schritte sind. Dieses Fenster öffnet sich immer, sodass du dort auch jederzeit nachschauen kannst, wenn du nicht mehr weißt, wie es genau geht.

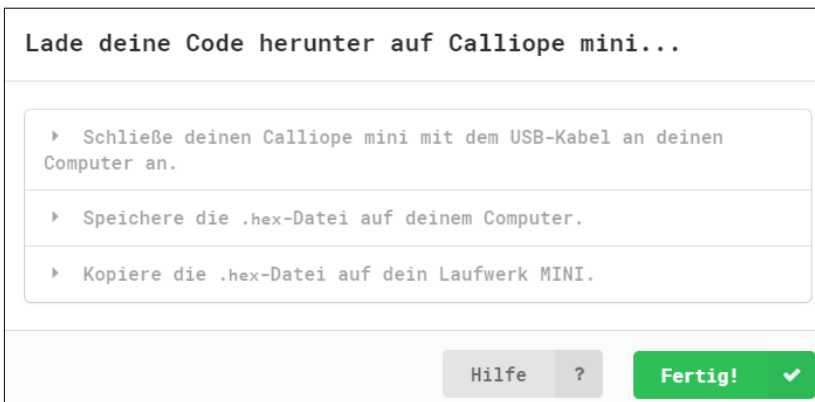


Abb. 2-6 Anleitung zum Herunterladen des Programms auf den Calliope mini

2. Als Nächstes musst du deinen Calliope mini mit dem USB-Kabel an den Computer anschließen (falls er nicht bereits angeschlossen ist).
3. Dann musst du das Programm, das nun in deinen Download-Ordner heruntergeladen wurde, auf den Calliope mini verschieben.

### Vorgehen bei anderen Betriebssystemen

Nicht alle Computer arbeiten an dieser Stelle gleich. Das liegt zum einen am Betriebssystem, also, ob du an einem Windows-, Linux- oder Apple-Gerät sitzt. Zum anderen laden verschiedene Browser (z. B. Mozilla Firefox, Google Chrome oder der Internetexplorer) die Dateien in unterschiedliche Ordner herunter. Es kann sogar sein, dass du beim Klicken auf »Herunterladen« gefragt wirst, wo das Programm gespeichert werden soll. Dann kannst du direkt deinen Calliope mini auswählen (siehe unten).

Dieser Schritt ist beim ersten Mal nicht ganz einfach und funktioniert auch nicht auf jedem Computer gleich, daher frag am besten einen Erwachsenen oder deine älteren Geschwister um Hilfe.



Zuerst öffnest du den *Explorer* auf deinem Computer. Das Symbol dazu sieht nicht immer ganz gleich aus. Wenn du an einem Windows-Computer sitzt, ist dies ein gelbes Ordnersymbol. Darauf musst du klicken bzw. doppelklicken.

Wenn du den Explorer geöffnet hast, sieht das Fenster etwa so aus wie in der nächsten Abbildung:

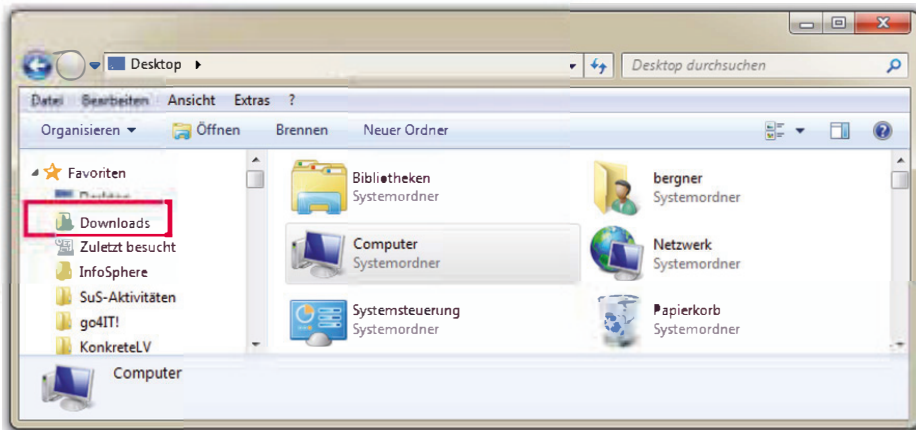


Abb. 2-7 Explorer-Fenster

Links in der Leiste siehst du schon das Wort »Downloads«. In diesen Ordner hast du eben dein Programm für den Calliope mini heruntergeladen. Daher wechsle mit einem Klick in den Ordner »Downloads«.

In diesem Ordner könnten mehrere Dateien liegen, die alle ganz verschiedene Namen haben können. Du musst nun die eine Datei suchen, die die Endung ».hex« hat. Vermutlich heißt diese Datei »mini-Ohne-Titel.hex«, da du deinem Projekt noch keinen Namen gegeben hast.

## Eine hex-Datei

Die Programme für den Calliope mini können verschiedene Namen haben, aber sie haben immer die Endung ».hex«. Die Datei, in der das Programm gespeichert wird, heißt also immer »Name.hex«, wobei der Name des Programms später von dir selbst festgelegt werden kann.

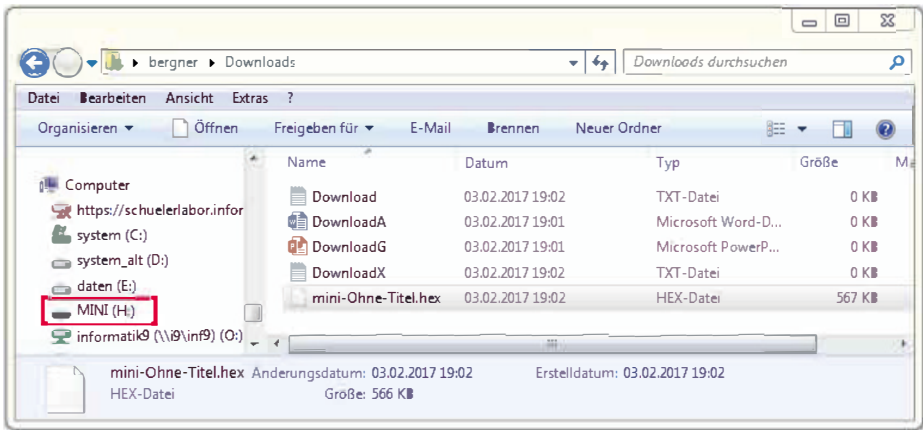


Abb. 2–8 Ordner »Downloads« mit verschiedenen Dateien

Wenn du die richtige Datei gefunden hast, kannst du sie ausschneiden. Klicke dazu mit der *rechten* Maustaste auf die hex-Datei und wähle dann »Ausschneiden« aus.

Links in der Spalte, in der auch der Ordner »Downloads« angezeigt wird, siehst du etwas weiter unten den Ordner für deinen Calliope mini. Klicke also nun auf »MINI«.

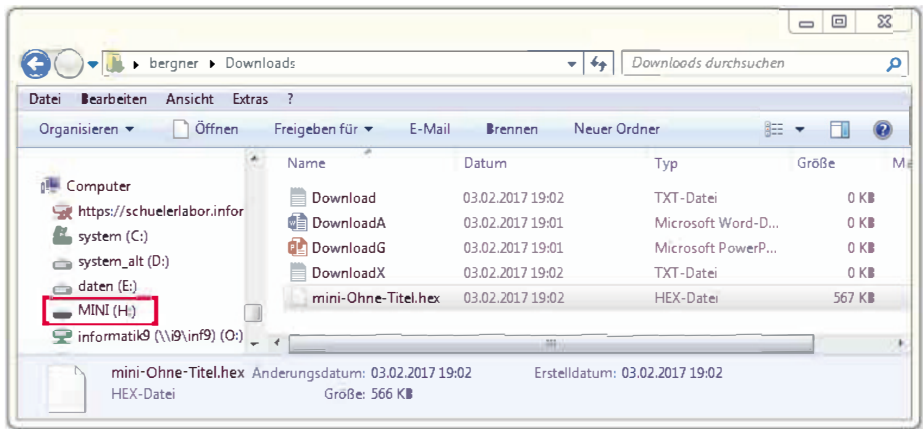


Abb. 2–9 Ordner »MINI« auswählen

In diesem Ordner sind bereits zwei Dateien. Die kannst du einfach dort lassen und dein Programm dazu speichern. Dazu wieder mit der rechten Maustaste unter die beiden vorhandenen Dateien klicken und »Einfügen« auswählen.

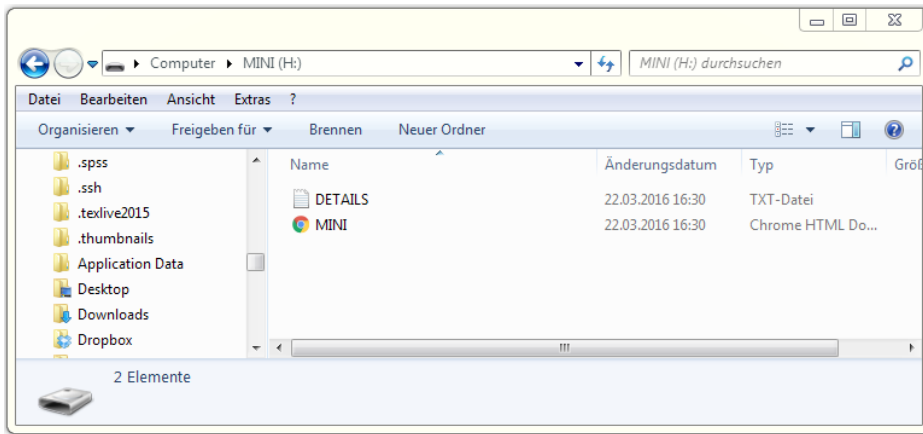


Abb. 2–10 Inhalt des Ordners »MINI«

Dann musst du kurz warten, bis dein Programm auf dem Calliope mini (also in dem Ordner »MINI«) gespeichert wurde. Währenddessen blinkt eine kleine gelbe LED auf dem Calliope mini, die dir anzeigt, dass er gerade arbeitet. Wenn das Programm übertragen wurde und alles geklappt hat, siehst du auf deinem Calliope mini ein Smiley mit Nase blinken.

Herzlichen Glückwunsch, du hast dein erstes Programm geschrieben und auf den Calliope mini übertragen!

Wenn du magst, kannst du auch ganz andere Muster auf dem LED-Display anzeigen lassen. Probier doch einmal auch folgende Dinge aus:

- ▶ Lass das Smiley auf dem Calliope mini einmal traurig gucken.
- ▶ Schaffst du es auch, dass das Smiley abwechselnd fröhlich und traurig guckt?
- ▶ Kannst du auch die Ziffer 8 darstellen?
- ▶ Und sogar den Buchstaben Z?

## 2.3 Erstes eigenes Programm »Deinen Namen schreiben«

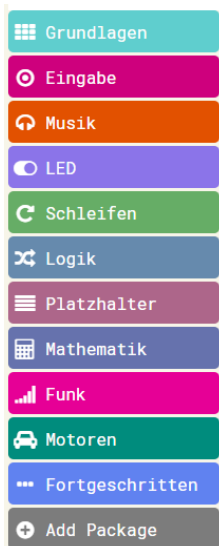
Nachdem du nun bereits Profi darin bist, dein Programm auf den Calliope mini zu übertragen, bist du bereit, ein eigenes neues Programm zu schreiben, denn dein Calliope mini kann noch viel mehr, als nur ein Smiley anzeigen zu lassen.

### Ziel: Eigenen Namen schreiben

Nun soll der Calliope mini nicht nur ein Symbol, eine Ziffer oder einen einzelnen Buchstaben darstellen, sondern deinen eigenen Namen als Laufschrift anzeigen.

### Laufschrift

Eine Laufschrift ist ein Text, der von rechts nach links durchs Bild wandert, wie du es vielleicht von elektronischen Anzeigen an Bushaltestellen, auf den Bildschirmen im Bus oder auch aus einem Schaufenster kennst.



Zum Programmieren musst du nun zurück auf die Webseite mit dem PXT-Editor gehen. Falls dort noch das Fenster mit den Tipps zum Herunterladen deines letzten Programms angezeigt wird, kannst du es einfach mit einem Klick auf »Fertig« schließen. Nun siehst du wieder dein altes Programm. Da du in den nächsten Kapiteln viele Programmierbefehle kennenlernen wirst, hier noch ein Tipp, woher du immer weißt, in welcher Gruppe (aus der Liste in der Mitte des Editors) du einen Befehl oder einen Block findest.

Abb. 2-11 Befehlsgruppen

## Farbe der Programmierbefehle und -blöcke

An der Farbe der Programmierbefehle und -blöcke erkennst du immer direkt, zu welcher Gruppe sie gehören.

An der hellblauen Farbe der drei Befehle im Smiley-Beispielprogramm kannst du erkennen, dass sie alle zur Gruppe »Grundlagen« gehören.

### 2.3.1 Löschen von Befehlen und Programmen

Damit du ein neues Programm schreiben kannst, musst du als Erstes das bestehende Programm löschen.

## Löschen von Befehlen, Blöcken oder einem kompletten Programm

Um das gesamte Programm zu löschen, klickst du auf den obersten Befehl (hier ist das der Befehl »dauerhaft«), hältst die Maustaste gedrückt und ziehst das alte Programm nach links, dorthin, wo die Befehlsgruppen aufgelistet sind. Dort wird dann ein Mülleimer-Symbol angezeigt. In diesen Mülleimer kannst du das alte Programm schieben. Das Gleiche kannst du auch mit einzelnen Befehlen oder Blöcken machen, wenn du dein Programm nur ein wenig verändern möchtest, statt komplett neu anzufangen.



Wichtig ist nun, dass deine Programmierfläche rechts komplett leer ist. Falls da noch andere Programmierbefehle übrig sind, solltest du diese ebenfalls löschen, bevor du mit deinem neuen Programm startest.

Abb. 2–12 Symbol »Mülleimer« zum Löschen

### 2.3.2 Dauerschleife anlegen

Jetzt ist es an der Zeit, dass du dein eigenes Programm schreibst. Öffne dazu die Befehlsgruppe »Grundlagen« mit einem einfachen Klick. Sofort öffnet sich eine Übersicht über alle Programmierbefehle und -blöcke in dieser Gruppe.

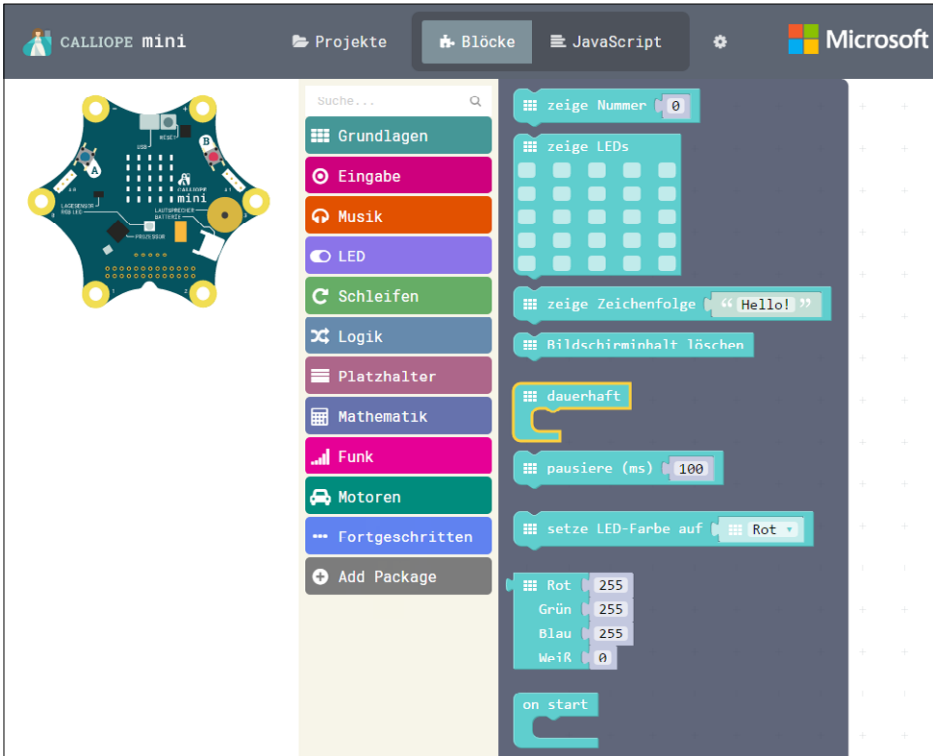


Abb. 2-13 Anzeige der Programmierbefehle und -blöcke in der Gruppe »Grundlagen«

Damit dein Name später auch durchgehend als Laufschrift erscheint, brauchst du zuerst wieder den Programmierbefehl »dauerhaft«. Sobald du den Befehl anklickst, verschwindet das zusätzliche Fenster und du kannst den Befehl auf der Programmieroberfläche ablegen. Dabei ist es egal, wo genau du den Befehl platzierst. Am besten oben links, damit auch die weiteren Befehle noch Platz haben. Wie du im Simulator schon siehst, passiert im Moment noch gar nichts, da du noch nicht programmiert hast, was denn dauerhaft passieren soll.

### 2.3.3 Text als Laufschrift ausgeben

Dazu musst du nun noch einmal die Gruppe »Grundlagen« öffnen und dort den Befehl »zeige Zeichenfolge« anklicken und ebenfalls auf die Programmierfläche ziehen.

## Zeichenfolge

Der Begriff »Zeichenfolge« bedeutet nichts anderes als Text, denn ein Wort besteht ja aus mehreren Zeichen, also Buchstaben, die in einer Folge hintereinander stehen. Zwischen zwei Wörtern befindet sich ein Leerzeichen, daher ist auch ein kompletter Satz eine Zeichenfolge.

Wenn du den Befehl »zeige Zeichenfolge« auf die Programmierfläche gezogen hast, hast du sicher schon bemerkt, dass er nur ausgegraut erscheint. Im Simulator siehst du auch, dass immer noch nichts passiert, also noch kein Text angezeigt wird.



Abb. 2-14 Ausgegrauter Befehl »zeige Zeichenfolge«

## Form der Programmierbefehle

Die Programmierbefehle haben verschiedene Formen. Manche haben oben eine kleine Delle. Das bedeutet, dass sie in einen anderen Befehl eingebaut werden müssen und nicht alleine stehen dürfen.

Der Befehl »zeige Zeichenfolge« hat oben eine solche kleine Delle und gehört damit zu den Befehlen, die nicht alleine funktionieren, sondern in einen anderen Befehl eingebaut werden müssen.

Im Befehl »zeige Zeichenfolge« siehst du einen eingebauten Programmierblock. An dieser Stelle musst du dich darum nicht weiter kümmern, da automatisch bereits ein Textblock mit dem Text »Hello!« eingefügt wurde.

Wenn du ganz genau hinschaust, siehst du beim Befehl »dauerhaft« in der Lücke in der Mitte einen kleinen Hubbel, der genau zu der Delle im Befehl »zeige Zeichenfolge« passt. Damit du ein funktionierendes Programm hast, musst du nun also den Befehl »zeige Zeichenfolge« in den Befehl »dauerhaft« einbauen. Die Befehle werden wie bei einem Puzzle zusammengesetzt. Wenn das gelungen ist, sind beide Befehle hellblau und im Simulator wird der Text »Hello!« angezeigt, was das englische Wort für »Hallo« ist.



Abb. 2-15 Programm zur Ausgabe des Textes »Hello!«

Wenn du nun in das Feld klickst, wo der Text »Hello!« steht, kannst du dort deinen eigenen Namen eintragen.

### 2.3.4 Programm testen und erweitern

Nun ist es wieder an der Zeit, dein Programm auf den Calliope mini zu übertragen. Klicke also auf »Herunterladen« und folge wieder der Anleitung, damit dein Name auch auf deinem echten Calliope mini angezeigt wird.

Das klappt natürlich nicht nur mit deinem Namen, sondern auch mit dem deiner Mutter, deines Vaters, deines Haustiers oder auch mit jedem anderen Wort. Du kannst sogar auch mehrere Wörter eintragen, versuch doch einmal Folgendes:

- ▶ Lass deinen Calliope mini fragen: »Wie geht es dir?«
- ▶ Lass ihn mit einem zweiten »zeige Zeichenfolge«-Befehl selbst antworten: »Mir geht es prima.«
- ▶ Kombiniere die Anzeige von Symbolen (zum Beispiel Smileys) mit der Anzeige von Text.
- ▶ Findest du heraus, wozu der Befehl »zeige Nummer« wohl genutzt werden kann?
- ▶ Experimentiere auch einmal mit dem Befehl »pausiere (ms)«.

#### »Pause«-Befehl

Wichtig beim Befehl »pausiere (ms)« ist, dass die Zeit, wie lange die Pause dauern soll, in Millisekunden (daher die Abkürzung »ms«) angegeben werden muss. Eine Millisekunde ist ein Tausendstel einer Sekunde. Wenn du also möchtest, dass dein Calliope mini eine Pause von 1 Sekunde macht, musst du in den Befehl »1000« eintragen, bei 2 Sekunden »2000« und so weiter.

## 2.4 Calliope mini macht Musik

Super, jetzt kennst du dich mit dem LED-Display ja bereits prima aus und kannst Text und Symbole anzeigen lassen. Aber wäre es nicht klasse, wenn dein Calliope mini dazu auch noch Musik macht?

Du kannst hier entweder dein letztes Programm weiter verwenden oder auch ein neues Programm beginnen. Wenn du ein neues Programm schreiben möchtest, brauchst du zuerst wieder den Befehl »dauerhaft«.

### Fehlerquelle: doppelte Befehle

Falls du versehentlich einmal den Befehl »dauerhaft« doppelt auf deine Programmierfläche gezogen hast, wird der zweite Befehl ausgegraut. Genau dies passiert auch, wenn du Befehle mit einer kleinen Delle oben nicht richtig in einen anderen Befehl einbaust. Alle ausgegrauten Befehle werden nicht ausgeführt, werden also von deinem Calliope mini ignoriert. Bevor du dein Programm auf den Calliope mini überträgst, solltest du immer prüfen, ob es ausgegraute Befehle gibt. Erst wenn alle Befehle farbig dargestellt werden, kann der Calliope mini dein Programm richtig ausführen.



Abb. 2-16 Fehlerquelle: ausgegraute Befehle

### 2.4.1 Töne abspielen

Damit der Calliope mini Musik machen kann, brauchst du einen Befehl aus der Gruppe »Musik«. Öffne die Gruppe und zieh den Befehl »spiele Note [C] für [1 Takt]« in deine Dauerschleife. Wenn der Lautsprecher an deinem Computer eingeschaltet ist, wirst du nun auch schon einen einzelnen Ton hören. Dieser Ton kommt vom Simulator, da dieser zur Ausgabe von Tönen die Lautsprecher deines

Computers benutzt. Falls sich in deiner Dauerschleife nur der Befehl »spiele Note [C] für [1 Takt]« befindet, hörst du den Ton ganz schnell hintereinander. Falls du gleichzeitig noch einen Lauftext programmiert hast, ertönt der Ton immer nur dann, wenn die Laufschrift einmal durchgelaufen ist.

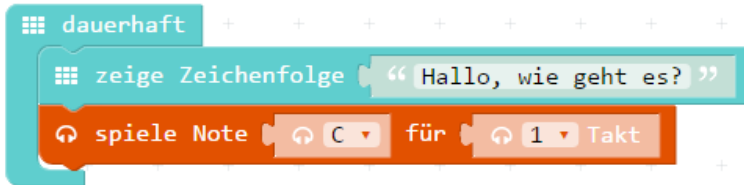


Abb. 2–17 Abfolge von zwei Befehlen

In diesem Beispiel wird zuerst der gesamte Text »Hallo, wie geht es?« angezeigt, bevor einmal der Ton »C« ertönt. Dann wird erneut der Text ausgegeben, bevor wieder der Ton ertönt und so weiter. Die Befehle »zeige Zeichenfolge« und »spiele Note« werden abwechselnd ausgeführt.

### 2.4.2 Lieder komponieren

Du kannst jetzt versuchen, ein eigenes kleines Lied zu komponieren.

- ▶ Probier doch einmal verschiedene Töne aus. Es gibt ganz viele verschiedene Töne, die du über den kleinen Pfeil neben dem »C« auswählen kannst.
- ▶ Genauso kannst du die Länge der Töne verändern, indem du sie kürzer abspielen lässt. Dazu musst du auf den kleinen Pfeil neben der »1« drücken. Wenn du »1/2« auswählst, bedeutet dies, dass der Ton nur halb so lange gespielt wird. Dein Lied wird also schneller.
- ▶ Mit dem Befehl »pausiere (ms)« aus der Gruppe »Grundlagen« kannst du dein Lied auch komplett pausieren. Aus dem Musikunterricht kennst du vielleicht schon ganze, halbe und Viertelnoten.

## 2.5 Programme umbenennen, speichern und weiter bearbeiten

### 2.5.1 Programme benennen und speichern

Wenn dir ein Programm ganz besonders gut gefällt, kannst du es auf deinem Computer speichern. Unten siehst du (rechts neben dem Knopf zum Herunterladen) ein Feld in dem »Ohne Titel« steht.

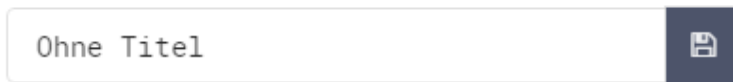


Abb. 2–18 Umbenennen und Speichern des Programms

Dies zeigt dir an, dass dein Projekt bisher noch keinen eigenen Namen hat. Du kannst dir einen passenden Namen überlegen und in das Feld schreiben. Anschließend drückst du auf das »Speichern«-Symbol rechts daneben. Damit hast du dein Programm (genau wie bei einem Klick auf »Herunterladen«) in den »Download«-Ordner deines Computers gespeichert. Von dort kannst du es auf deinen Calliope mini verschieben oder auch an einen anderen Ort auf deinem Computer. Du gehst wie folgt vor:

- ▶ Schreib den gewünschten Namen in das Feld.
- ▶ Drück auf das »Speichern«-Symbol.
- ▶ Öffne den Ordner »Downloads«.
- ▶ Statt das Programm auf das Laufwerk »MINI« zu verschieben, speicherst du das Programm in einem Ordner auf deinem Computer ab, den du später leicht wiederfindest.

Falls du den Computer mit anderen teilst oder dich noch nicht so gut mit dem Speichern von Dateien auskennst, bitte jemanden, dir dabei zu helfen. Du möchtest ja nicht, dass jemand anderes versehentlich deine Programme löscht, also sag besser kurz Bescheid, wo du deine Programme abspeicherst.

## 2.5.2 Programme importieren und weiter bearbeiten

Wenn du nun später an einem deiner Projekte weiterarbeiten möchtest, musst du folgende Schritte gehen:

1. Öffne wieder den PXT-Editor in deinem Browser.
2. Klicke oben auf »Projekte«.
3. Klicke dann auf »Datei importieren ...«.

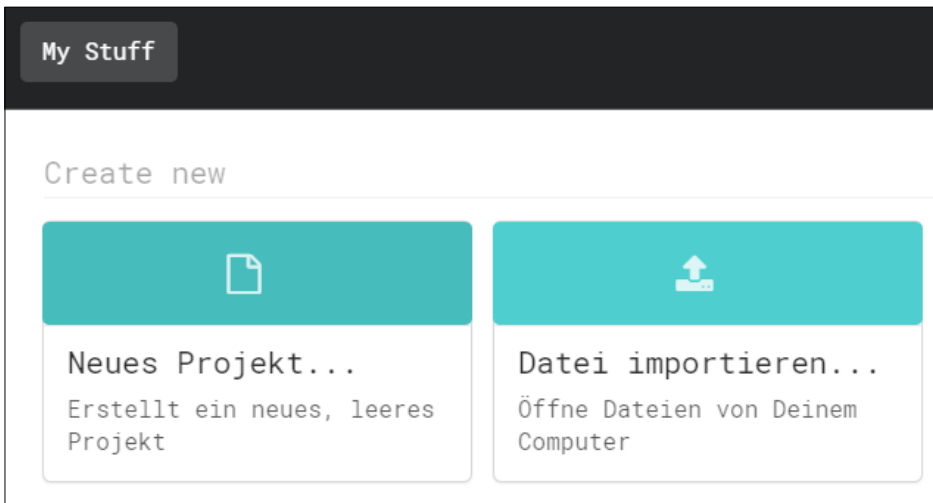


Abb. 2–19 Neues Projekt erstellen oder vorhandene Datei importieren

### Importieren

Das Wort »Importieren« kennst du vielleicht schon von Waren, die aus dem Ausland nach Deutschland importiert werden, damit auch wir Ananas oder andere Früchte essen können, die in Deutschland nicht wachsen. Importieren im PXT-Editor bedeutet ebenfalls, dass du etwas von außerhalb einfügst. In diesem Fall fügst du deine Programmdatei (mit der Endung ».hex«), die du früher einmal abgespeichert hast, in den PXT-Editor ein.

4. Nun kannst du unter »Datei auswählen« die gespeicherte Datei auf deinem Computer auswählen, an der du gerne weiterarbeiten möchtest.

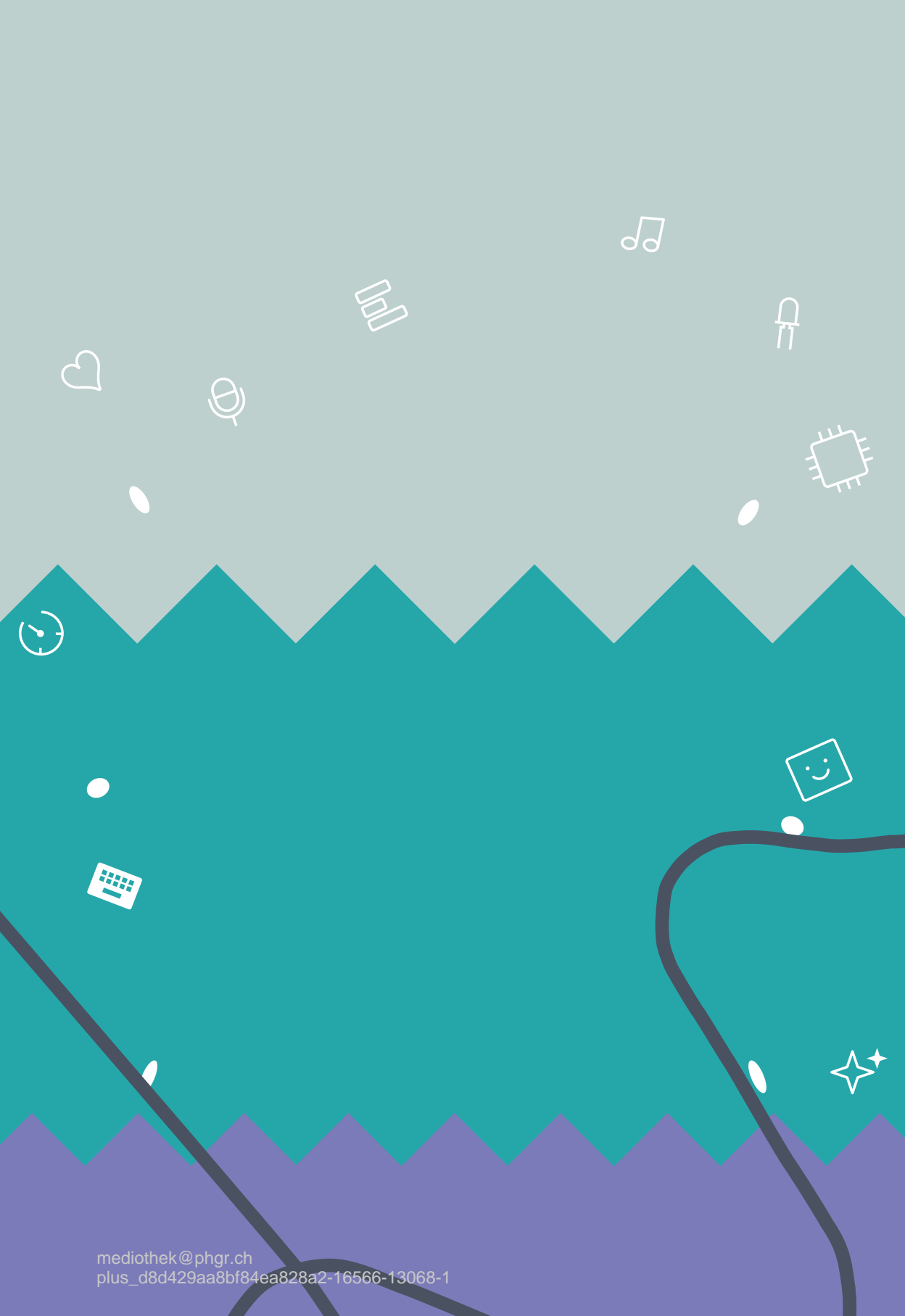
Hierzu musst du in dem geöffneten Fenster wieder den Ordner öffnen, wo du früher einmal dein Projekt abgespeichert hast.



Abb. 2-20 Datei zum Importieren auswählen

5. Anschließend solltest du alle verwendeten Programmierbefehle wieder auf der Programmierfläche sehen. Du kannst sie nach Belieben verändern.

Damit du auch immer das richtige Programm findest, ist es wichtig, dass du deinen Projekten passende Namen gibst.



# 3

## Mit dem Calliope mini interagieren

In diesem Kapitel dreht sich nun alles rund um die beiden Knöpfe, die auf dem Calliope mini mit »A« und »B« beschriftet sind.

Die Einführung in dieses Kapitel wurde von Nadine Bergner geschrieben, das Bastelprojekt »Ein Klavier mit dem Calliope mini« wurde von Mario Pesch erstellt und das »Angelspiel« stammt von Natalia Prost.

## 3.1 Wenn, dann, was?

Du hast es bereits geschafft verschiedene Formen, Zahlen und sogar Texte auf dem LED-Display anzeigen zu lassen, auch hast du den Calliope mini schon ein Lied spielen lassen. Auf dem Calliope mini gibt es aber noch weitere spannende Bauteile. Falls du dich nicht mehr genau erinnerst, schau gerne nochmal in Kapitel 1 nach, welche Bauteile es gibt. In diesem Kapitel dreht sich nun alles rund um die beiden Knöpfe, welche mit »A« und »B« beschriftet sind.

### Knöpfe

Knöpfe sind keine Schalter. Das bedeutet, dass diese nicht wie ein Lichtschalter funktionieren, den du einmal drückst und dann bleibt das Licht an, bis du noch einmal auf den Schalter drückst und das Licht ausschaltest. Bei einem Knopf leuchtet das Licht nur dann, wenn du den Knopf herunterdrückst, und geht sofort aus, wenn du den Knopf loslässt. Wichtig ist also, dass die beiden Knöpfe auf dem Calliope mini nur merken können, ob sie heruntergedrückt werden oder eben nicht.

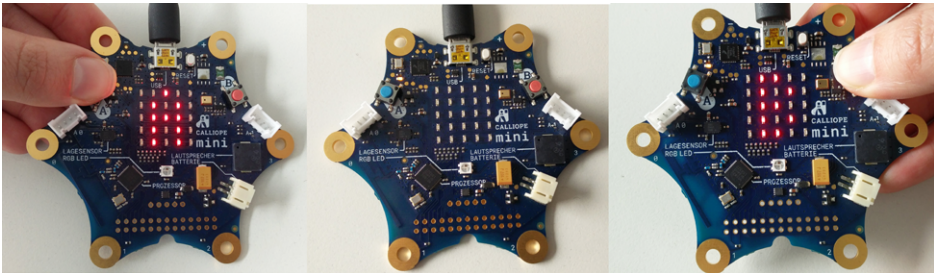


Abb. 3-1 Calliope mini reagiert auf Knopfdruck

Aber der Calliope mini soll nicht nur merken, ob einer der Knöpfe heruntergedrückt wird, sondern dann auch etwas Bestimmtes tun. Wie wäre es zum Beispiel, wenn der Calliope mini beim Herunterdrücken des Knopfes »A« einen Ton abspielt und beim Drücken des Knopfes »B« ein Smiley anzeigt?

Also los geht es! Öffne ein neues Projekt und lösche (wie in Kapitel 2 beschrieben) das bisherige Beispielprogramm. Das Ziel ist jetzt, dass der Calliope mini erst dann etwas tut (also einen Ton abspielt oder einen Smiley anzeigt), wenn ein Knopf gedrückt wird. Daher brauchen wir diesmal nicht die Dauerschleife, sondern einen neuen Befehl, nämlich einen »Wenn-Dann-Befehl«.

## Wenn-Dann-Befehl

»Wenn-Dann-Befehle« sind in der Informatik ein ganz wichtiges Element, wenn man möchte, dass ein technisches Gerät (wie auch der Calliope mini) dann und nur genau dann einen bestimmten Befehl (z. B. Ton abspielen) ausführt, wenn etwas Bestimmtes (z. B. Knopf »A« wird gedrückt) passiert ist. Auf Englisch heißt dies auch »if-else« und ist ein Programmierkonstrukt, welches in praktisch allen (auch den professionellen) Programmiersprachen existiert.

In der Gruppe »Eingabe« findest du den Befehl für die Abfrage, ob der Knopf »A« gedrückt ist.

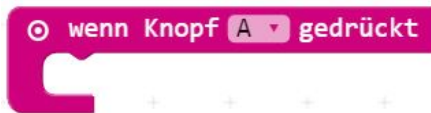


Abb. 3–2 Befehl »Wenn Knopf A gedrückt«

Jetzt wunderst du dich vielleicht, wo denn das »dann« geblieben ist? Dieses wird in diesem Befehl nur nicht hingeschrieben, aber alle weiteren Befehle, die du in diesen Befehl hineinziehst, werden nur dann ausgeführt, wenn der Knopf »A« gedrückt ist. Lasse den Calliope mini nun den Ton »C« abspielen, wenn der Knopf »A« gedrückt wird. Zur Erinnerung: Den passenden Befehl dafür findest du in der Gruppe »Musik«.

Jetzt ist es an der Zeit, dein Programm zu testen. Erinnerst du dich noch an die Schritte, die du machen musst, um dein Programm auf den Calliope mini zu übertragen? Falls du Probleme hast, schaue noch einmal in Kapitel 2 nach, wie es genau geht. Nun sollte dein Calliope mini auf Knopfdruck einen Ton abspielen. Wenn es nicht klappt, kontrolliere, ob du die richtigen Befehle verwendest und ob diese richtig zusammengesteckt sind.

Nun war der Plan aber, dass auch beim Druck auf Knopf »B« etwas passiert. Der Calliope mini sollte ein Symbol, zum Beispiel ein lachendes Smiley, auf dem LED-Display anzeigen. Dazu brauchst du nun den Befehl »wenn Knopf B gedrückt«. Aber der ist ja in der Gruppe »Eingabe« gar nicht zu finden? Kein Problem, denn neben dem »A« siehst du wieder einen kleinen Pfeil. Dieser bedeutet, dass du dort eine Auswahlliste öffnen kannst. Also ziehe dir noch einen zweiten Befehl »wenn Knopf A gedrückt ist« auf deine Programmierfläche und klicke auf den kleinen Pfeil neben dem »A«.

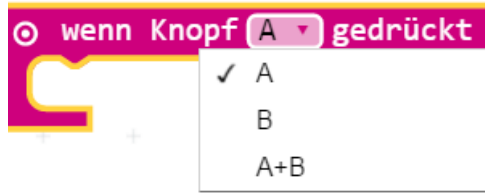


Abb. 3–3 Auswahlliste der Knöpfe

Du siehst direkt, dass du deinen Calliope mini so programmieren kannst, dass dieser eine bestimmte Sache nur dann tut, wenn beide Knöpfe gedrückt werden. Aber dazu später mehr. Hier brauchst du erst einmal nur das »B« auszuwählen. Weiter kannst du nun mit dem Befehl »zeige LEDs« aus der Gruppe »Grundlagen« ein beliebiges Muster anzeigen lassen. Probiere dein Programm einmal auf deinem echten Calliope mini aus.

Funktionieren beide Knöpfe wie gewünscht? Da du nur programmiert hast, dass das Symbol beim Druck auf Knopf »B« angezeigt wird, aber nicht, dass dieses auch wieder gelöscht wird, musst du zum erneuten Test das Programm durch einen Druck auf den Reset-Knopf (der kleine Knopf oben in der Mitte) neustarten. Jetzt kennst du alles, was du brauchst, damit dein Calliope mini auf den Druck der Knöpfe reagiert. Wie wäre es also mit ein paar neuen Ideen?

- ▶ Lass doch den Calliope mini beim Druck auf Knopf »A« ein trauriges und beim Druck auf Knopf »B« ein lachendes Smiley anzeigen. So kannst du den Calliope mini nutzen, um deine momentane Stimmung auszudrücken.
- ▶ Auch kannst du den Calliope mini mehrere Dinge ausführen lassen, wenn ein Knopf gedrückt wird. Zum Beispiel könntest du ein (fröhliches oder auch trauriges) Smiley blinken lassen.
- ▶ Wenn beide Knöpfe gleichzeitig gedrückt werden, könnte dies bedeuten, dass deine Stimmung mittelmäßig ist. Dann könnte das Smiley einen geraden Strich als Mund anzeigen.
- ▶ Natürlich kannst du auch drei verschiedene Töne ausgeben lassen, so dass dein Calliope mini zu einem Instrument wird, welches allerdings nur drei verschiedene Töne abspielen kann.

Was sind deine Ideen, was der Calliope mini beim Druck auf Knopf »A«, Knopf »B« oder beim Drücken beider Knöpfe tun soll? Leg los und setze deine eigenen Ideen um. Viel Spaß dabei!

## 3.2 Ein Klavier mit dem Calliope mini

Du hast deine ersten Schritte mit dem Calliope mini bereits gemeistert. Du hast gelernt, wie der Calliope mini Töne erzeugen und wie du mit ihm interagieren kannst. Nun wollen wir beides kombinieren und den Calliope mini verwenden um etwas fantastisches zu bauen – dein eigenes Klavier! In dem Projekt wirst du lernen mit verschiedenen Alltagsgegenstände Töne zu erzeugen. Neben der Programmierung werden wir auch ein wenig basteln!

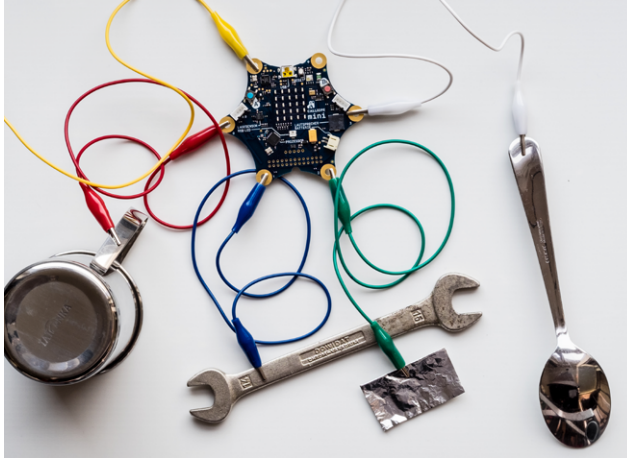


Abb. 3–4 Das Calliope mini-Klavier!

### 3.2.1 Welche Materialien und Werkzeuge werden benötigt?

Wie viele?	Was	Anmerkung
1	Calliope mini	
5	Krokodilklemmen	
	Alufolie	
	Verschiedene leitfähige Gegenstände	Hier kannst du deiner Phantasie freien Lauf lassen

Tab. 3–1 Verwendete Gegenstände für das Calliope mini-Klavier

## 3.2.2 Die Vorbereitung

- ▶ Für das Projekt benötigen wir die Informationen aus dem Abschnitt 3.1. Falls dir noch etwas unklar ist ließ dir das Kapitel erneut durch.
- ▶ Du kannst sehr viele verschiedene Materialien als Tasten für dein Klavier verwenden, wichtig ist nur, dass diese den elektrischen Strom leiten. Eine Auswahl findest du im nachfolgenden Kasten zum Thema »Elektrische Leitfähigkeit«.

### Elektrische Leitfähigkeit

»Elektrische Leitfähigkeit« ist ein Begriff, der aus der Physik kommt, und beschreibt die Fähigkeit eines Gegenstandes, Strom weiterzuleiten. Wichtig ist für dich, dass ein Gegenstand dann leitfähig ist, wenn er Strom übertragen kann. Strom kannst du dir als ganze kleine Teilchen vorstellen, die vom Minuspol zum Pluspol wandern.

Eine kleine Auswahl an leitfähigen Gegenständen:

- ▶ Alufolie
- ▶ Kupferkabel
- ▶ Gabel, Messer, Löffel
- ▶ Äpfel, Bananen und viele weitere Obstsorten
- ▶ Mineralwasser
- ▶ ganz viele verschiedene Metalle

## 3.2.3 Das Basteln

Für das Calliope mini-Klavier solltest du dir die ersten Kapitel aufmerksam durchgelesen und die ersten Schritte erfolgreich gemeistert haben. Wir beginnen zuerst mit dem Bau des Klaviers und werden anschließend die Programmierung durchführen. Für das Klavier verwendest du die goldenen Anschlüsse an den Ecken des Calliope mini. Wenn du dir die Abbildung 3–5 anschaust, siehst du, dass die Ecken mit - (Minuspol), 0, 1, 2, 3 und + (Pluspol) markiert sind.

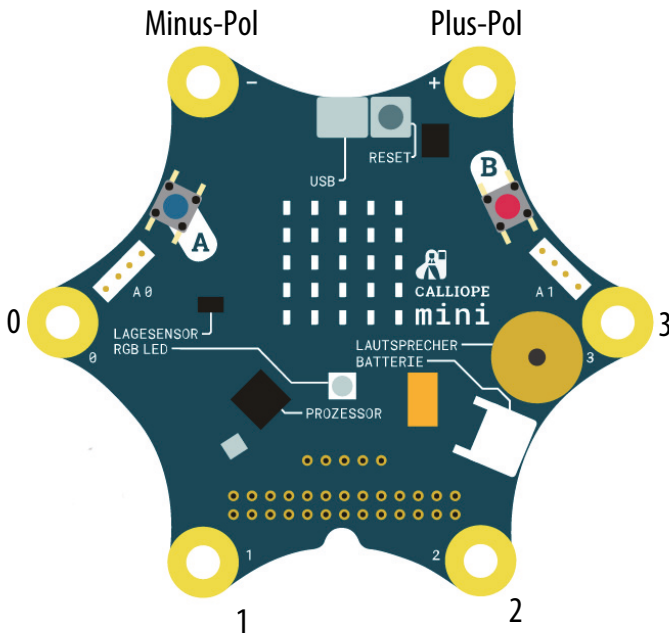


Abb. 3-5 Übersicht über die Anschlüsse des Calliope mini

Der Anschluss Minus ist ein Minuspol. Nimm eine Krokodilklemme und verbinde sie mit dem Anschluss.

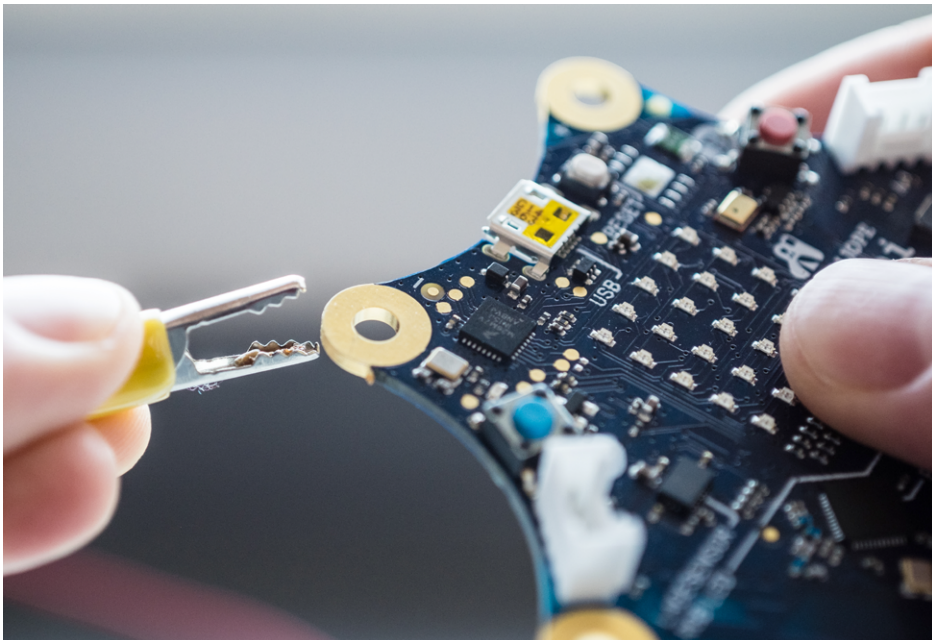


Abb. 3-6 Anschluss einer Krokodilklemme an den Calliope mini

Damit der Calliope mini erkennen kann, dass eine Taste bzw. ein Gegenstand gedrückt wird, muss der Stromkreis geschlossen werden. Für den geschlossenen Stromkreis benötigst du immer einen Minus- und einen Pluspol (das kennst du vielleicht schon von einer Batterie). Die Anschlüsse 0-3 sind auf dem Calliope mini Pluspole. Du kannst nun einen Gegenstand oder Alufolie mit Hilfe der Krokodilklemme mit einem der drei Anschlüsse verbinden.



Abb. 3-7 Verbindung von Anschluss 0 mit einem Gegenstand

Verbinde nun drei weitere leitfähige Gegenstände mit den Anschlüssen (in der Programmieroberfläche »Pin« genannt) 1-3 auf dem Calliope mini.

### 3.2.4 Die Programmierung

Du hast den ersten Schritt erfolgreich gemeistert und schon viele Verbindungen mit Hilfe der Krokodilklemmen und leitfähigen Gegenständen hergestellt. Nun müssen wir den Calliope mini noch programmieren. Immer wenn einer der Gegenstände gedrückt wird und somit der Stromkreis geschlossen ist, soll ein Ton abgespielt werden. Du benötigst den Programmierbefehl »wenn Pin .. gedrückt« aus der Gruppe »Eingabe« und den Programmierbefehl zum Erzeugen von verschiedenen Tönen aus der Gruppe »Musik«. Auf deinem Calliope mini siehst du, dass die goldenen Anschlüsse von 0-3 nummeriert sind. In der Programmieroberfläche heißen die Pins »P0«-»P3«.



Abb. 3–8 Programmierbefehle zum Erzeugen eines Tons bei Drücken des Gegenstandes an Pin P0

Du musst nun für die Anschlüsse P1, P2 und P3 jeweils einen solchen Block in die Programmierfläche ziehen. Die Töne kannst du entsprechend ändern. Bei einem normalen Klavier ist die Abfolge C D E F (G A H).

Zum Schluss kannst du dein Programm speichern und auf deinen Calliope mini kopieren.

### 3.2.5 Klavier spielen!

Geschafft! Du hast dein Klavier aufgebaut und die Programmierung vorgenommen. Halte nun die Krokodilklemme, die du mit dem Minuspol verbunden hast, an dem Metallende fest. Du kannst nun mit der anderen Hand die Gegenstände berühren und Töne erzeugen. Durch das Berühren der Krokodilklemmen und des Gegenstandes wird der Stromkreis geschlossen und der Calliope mini erkennt dies als Drücken des Pins. Alternativ kannst du auch die Gegenstände mit dem Ende der Krokodilklemme berühren und so den Stromkreis schließen. Falls es mal nicht funktionieren sollte, überprüfe deinen Programmiercode oder verwende einen anderen Gegenstand, evtl. ist dein Gegenstand nicht leitfähig.

### 3.2.6 Weitere Ideen

Bestimmt hast du nun deine ersten Lieder gespielt. Du wirst allerdings feststellen, dass du mehr Tasten bzw. Gegenstände benötigst, um eine vollständige Tonleiter zu spielen. Vielleicht hast du noch einen zweiten Calliope mini zur Verfügung, sodass du weitere Töne erzeugen kannst? Neben einem Klavier kannst du auch weitere Musikinstrumente bauen. Stelle dir deine eigene Calliope mini-Band zusammen!

## 3.3 Angelspiel

Bei diesem Spiel wirst du eine Angel an deinen Calliope mini anschließen, um Fische aus dem gefährlichen Stromsee zu fangen. Okay, die Angel ist eher nur ein Stift. Und die Fische sind auch nur Büroklammern. Und der See ist eigentlich ein kleines Aquarium aus Pappe. Aber hey, Strom fließt da wirklich!

Du wirst die Pins des Calliope mini benutzen, um einen Stromkreis zwischen deiner Angel und dem Aquarium zu schließen. Aber Moment, Fische mögen keinen Strom! Also pass besser auf, dass der Stromkreis nicht geschlossen wird – sonst gibt es einen sehr nervigen Ton.

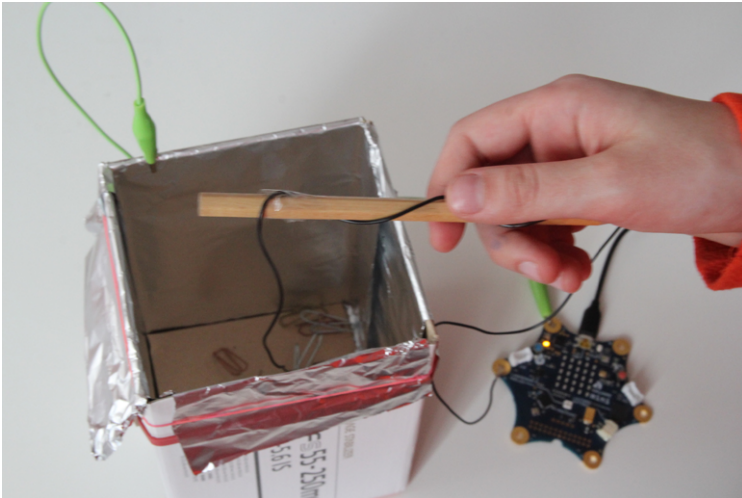


Abb. 3–9 Kannst du alle Büroklammern rausangeln ohne die Wand zu berühren?

### 3.3.1 Benötigte Materialien

Für dieses Spiel wirst du erstmal etwas basteln, bevor du es programmieren wirst. Neben deinem Calliope mini und dem üblichen USB-Kabel brauchst du noch ein paar Materialien, um dein Aquarium und Angelzubehör zusammenzubauen. Frage einen Erwachsenen, wenn dir etwas davon fehlt.

Wie viele?	Was?	Anmerkung
1	Leerer Karton, z. B. Verpackungen	Oder 4 Platten aus Karton, die du im Rechteck verbindest
1 Rolle	Aluminiumfolie	Du brauchst nicht die ganze Rolle, aber ungefähr so viel, dass der Karton innen abgeklebt werden kann.
1	Schere	
1	Kleber oder doppelseitiges Klebeband	
1	Gummiband	Zum Runterklappen der Kartonlaschen, wenn du sie nicht abschneiden willst
1	Stift	als Angel →

Wie viele?	Was?	Anmerkung
60 cm	Dünnes Kabel mit Isolierung oder Krokodilklemmen	als Angelschnur
etwas	Klebeband	
ungefähr 10	Büroklammern aus Metall	
1	Krokodilklemme	

Tab. 3–2 Materialien für das Angelspiel



Abb. 3–10 Benötigte Materialien, dazu: Karton deiner Wahl

## Getränk kartons mit Aluminium

Es gibt manche Getränkekartons, die auf der Innenseite mit einer sehr dünnen Aluminiumschicht beschichtet sind, um die Flüssigkeit im Karton (z. B. Milch) besonders zu schützen. Du kannst auch so einen beschichteten leeren Karton als Aquarium ausprobieren und brauchst dann keine eigene Aluminiumfolie reinkleben.

Aber Achtung: da der Boden dieser Kartons auch mit Aluminium beschichtet ist, musst du den Boden mit einem zugeschnittenen Pappstück abdecken. Dann kannst du deine Büroklammern reinlegen, ohne dass sie schon die Strom leitende Schicht berühren.

### 3.3.2 Aquarium, Angel und Fische bauen

Der Karton wird das Aquarium sein, in dem du angeln wirst. Je schmaler und höher dein Karton ist, desto schwieriger wird das Angeln für dich. Nimm deinen leeren Karton und leg ihn auf die Seite auf die Aluminiumfolie. Dann weißt du, wie viel Aluminium du für diese eine Innenwand des Aquariums brauchst. Umrande den Karton mit einem Stift, wobei du ruhig etwas mehr Rand lassen kannst. Klebe die ausgeschnittene Folie an die Innenseite des Kartons. Wiederhole diesen Schritt, bis du auf jeder Seite des Kartons eine Schicht Aluminiumfolie hast. Dabei ist es wichtig, dass sich alle Folien an den Seiten ein bisschen berühren. Die Aluminiumfolie kann Strom leiten und stellt hier eine Art Stromsee dar.



Abb. 3–11 Das fertige Aquarium

Als nächstes baust du die Angel. Du kannst auch einfache Schnur nehmen. Dann musst du aber auf jeden Fall deinen Angelhaken mit einer Krokodilklemme oder einem anderen leitenden Kabel mit dem Calliope mini verbinden. Nimm das Kabel und miss ungefähr 20 cm an dem einen Ende ab. Das wird deine Schnur sein, die frei schwingen kann, wenn du den Stift in der Hand hältst. Wickel das Kabel nun um den Stift und klebe es mit Klebeband am Stift fest. Achte darauf, dass du am Ende noch ein Stück Kabel übrig hast, das du mit dem Calliope mini verbinden kannst. Wenn dieses Stück zu kurz ist, kannst du es mit einer Krokodilklemme verlängern.

**WENN** du auf dem Bild genau hinschaust, dann siehst du, dass das Kabel an beiden Enden nicht mehr isoliert ist, also keine schwarze Ummantelung mehr hat. Dies ist wichtig, damit du das Metall des Kabels direkt an die Pins am Calliope mini und an den Angelhaken anschließen kannst. Du kannst den Gummimantel selbst entfernen, indem du mit einer kleinen Schere nur ganz leicht zudrückst und das Kabel dabei etwas drehst. Dabei wird der Mantel durchgeschnitten, aber nicht das Kabel selbst, weil du die Schere nicht ganz zumachst. Dann kannst du den abgetrennten Mantel leicht vom Ende abschieben. Dieser Vorgang heißt »Entmantelung«.

**FRAGE** einen Erwachsenen, wenn du dabei Hilfe brauchst.

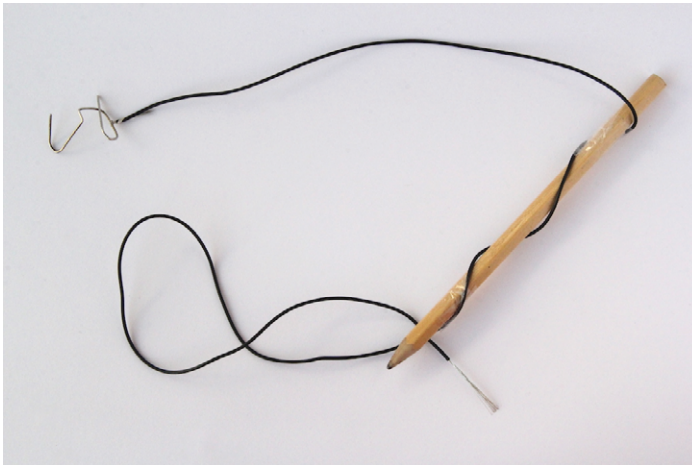


Abb. 3-12 Die fertige Angel

An das vordere Ende des Kabels, das deine Angelschnur sein wird, hängst du eine Büroklammer aus Metall. Verbiege sie etwas, bis sie aussieht wie ein Haken, und befestige das offene Metallkabel daran. Fertig ist dein Angelhaken.



Abb. 3–13 Detailansicht des Angelhakens

Als Fische kannst du ganz einfach auch Büroklammern aus Metall benutzen. Aber wenn diese Büroklammern nur flach auf dem Aquariumboden liegen würden, kannst du sie natürlich nicht mit dem Haken aufsammeln. Also biege am besten die innere Klammer etwas nach außen. Je weniger du verbiegst, desto schwieriger wird es später, mit dem Haken reinzukommen. Probiere es aus und biege manche Büroklammern mehr und manche weniger.



Abb. 3–14 Büroklammer-Fische

## Aluminium-Fische

Du kannst auch (fast) echte Fische aus Aluminiumfolie ausschneiden und in dein Aquarium werfen. Dafür musst du dann aber kleine Ringe oder Schlaufen aus Draht oder Aluminiumfolie basteln, die du an den Fischen festklebst. Dort kannst du deinen Angelhaken dann einhaken und den Fisch aus dem Aquarium rausholen. Achte darauf, dass alle Metallelemente miteinander verbunden sind, damit der Strom fließen kann: vom Calliope mini-Minus-Pol (-), über die Angelschnur, über den Haken, den Fisch die Seiten des Aquariums und zurück zum Calliope mini-Pin.

Lege die Fische in das Aquarium. Dabei dürfen sie die Aluminium-Wände des Aquariums nicht berühren. Wenn der Boden deines Aquariums etwas uneben ist, lege ein Stück passend großen Karton auf den Boden, damit die Fische gerade liegen können. Du kannst natürlich so viele Fische auslegen, wie du möchtest.



Abb. 3-15 Büroklammer-Fische auf dem Boden des Aquariums

### 3.3.3 Mit dem Calliope mini verbinden

Nun verbindest du das Aquarium, die Angel und deinen Calliope mini so miteinander, dass ein geschlossener Stromkreis entstehen kann. Dafür klemmst du ein Ende einer Krokodilklemme an die Aluminiumfolie deines Aquariums und das andere Ende der Klemme an den Minus-Pol (-) an deinem Calliope mini. Das eine Ende deiner Angelschnur, an dem kein Angelhaken hängt, befestigst du am Pin 0 des Minis.

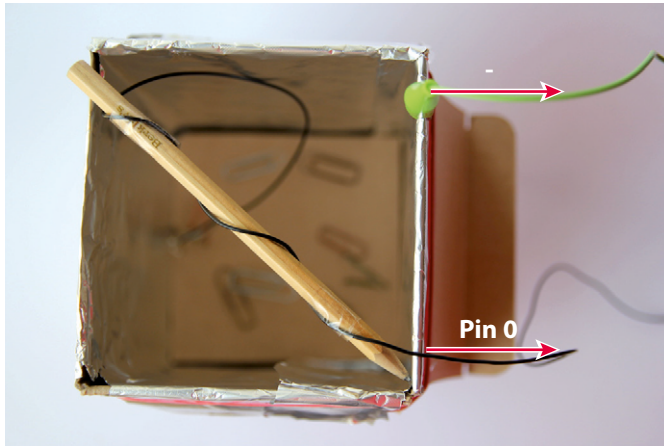


Abb. 3–16 Verbindung zwischen dem Calliope mini und dem Angelspiel (linker Teil)

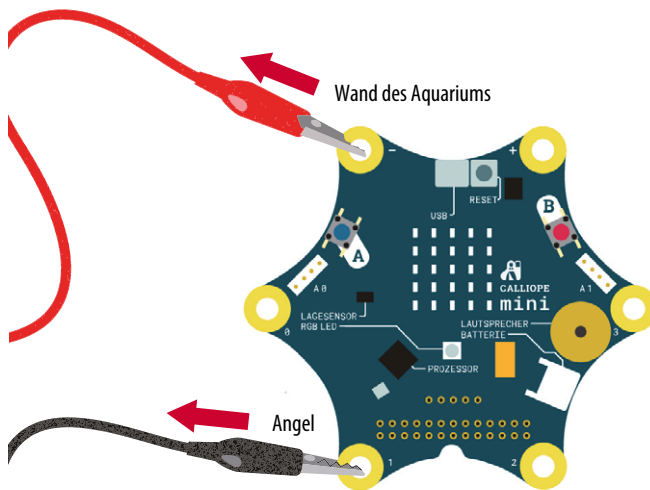


Abb. 3–17 Verbindung zwischen dem Calliope mini und dem Angelspiel (rechter Teil)

Wenn nun das andere Ende des Kabels, wo der Angelhaken hängt, die leitende Aluminiumwand des Aquariums berührt, ist der Stromkreis geschlossen und der Strom fließt. Damit bist du mit dem Aufbau des Spiels fertig und kannst dich im nächsten kurzen Schritt um die Programmierung des Spiels kümmern.

### 3.3.4 Die Programmierung

Öffne nun den PXT-Editor. Für das Angelspiel hast du bereits alle Programmierbefehle kennengelernt. Das wichtigste ist, dass der Calliope mini reagiert, wenn der Stromkreis geschlossen wird, also der Angelhaken oder die anderen Büroklammern eine der Innenwände berühren. Das Programm muss dafür prüfen, ob der Pin 0 gedrückt wurde. Benutze die »wenn Pin 0 gedrückt«-Klammer aus der Befehlsgruppe »Eingabe«. Füge dort einen »spiele Note«-Befehl ein.

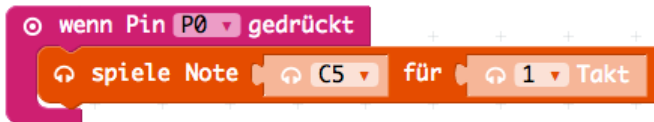


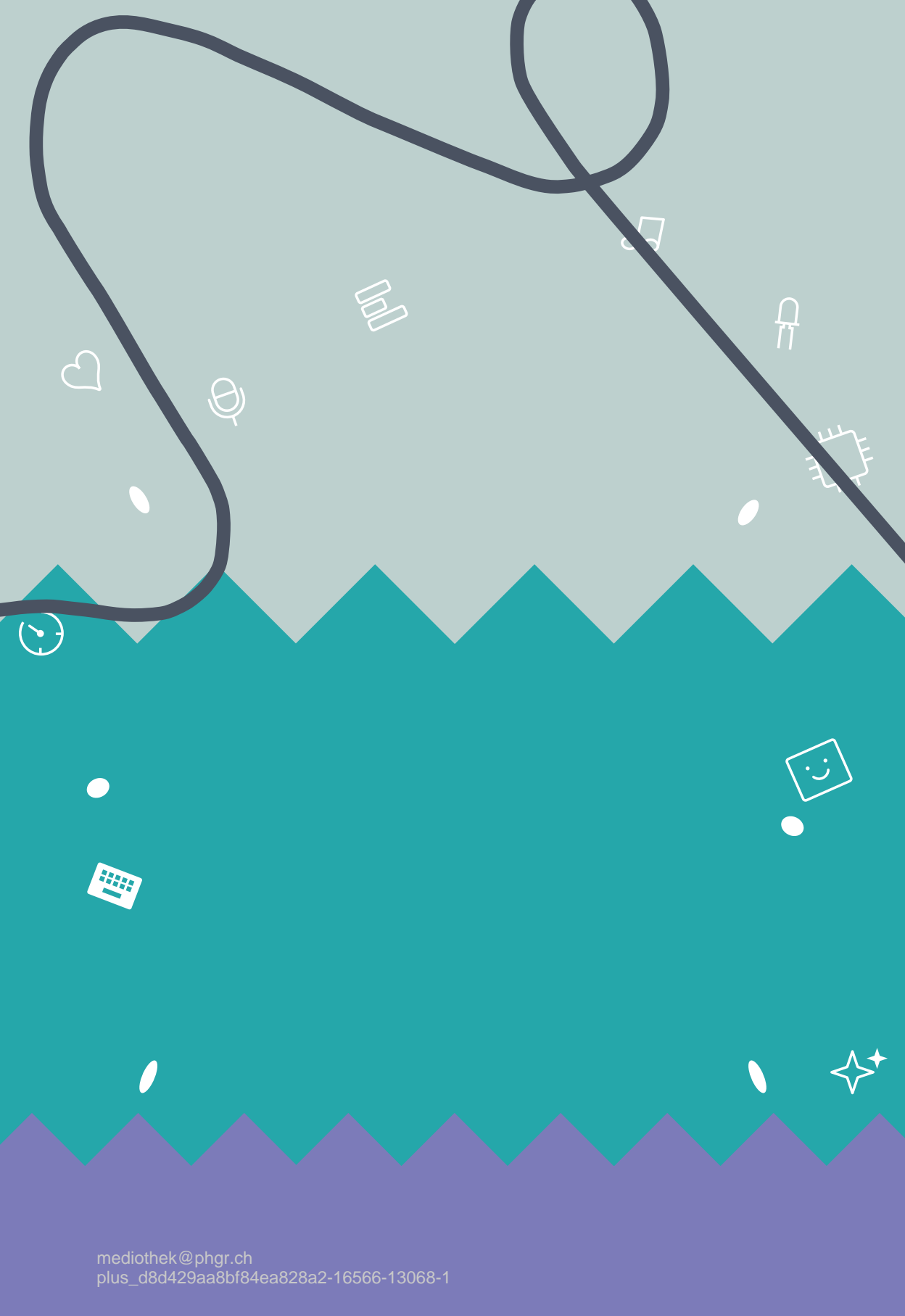
Abb. 3–18 Programm für das Angelspiel

Lade das Programm auf deinen Calliope mini herunter. Was passiert, wenn du die Wand mit dem Angelhaken berührst? Natürlich kannst du auch einen anderen nervigen Ton aussuchen, bestimmt findest du etwas Passendes.

### 3.3.5 Andere Stromkreisspiele

Kannst du dir ein anderes Spiel vorstellen, bei dem es wichtig ist, einen Stromkreis zu schließen – oder eben nicht? Vielleicht kennst du das Spiel »Der heiße Draht«. Dabei gibt es einen gebogenen Draht, der an den Enden irgendwo befestigt ist, und du hältst einen fast geschlossenen Drahhaken in deiner Hand. Ziel ist es, mit dem Haken von Anfang bis Ende durch den Draht zu ziehen, ohne dass sich beide Drähte berühren. Du kannst auch ein Labyrinth aus nichtleitendem dünnem Klebeband (z. B. Washi-Tape) auf einer großen Aluminium-Fläche aufkleben. Versucht das Labyrinth zu durchqueren, ohne mit deinem Draht von der Strecke abzukommen und das Aluminium zu berühren.

Bestimmt fallen dir noch viele andere Spiele ein, bei denen du den Calliope mini als kleine Stromquelle benutzen kannst.



# 4

## Die Sensoren beim Calliope mini

In diesem Kapitel lernst du alle Sensoren des Calliope mini kennen.

Die Einführung in dieses Kapitel wurde von Mario Pesch und Lina Wassong geschrieben. »Die Wasserwaage« stammt von Patrick Franken, das Projekt »Die Calliope-Alarmanlage« von Lina Wassong, »Farbthermometer« von Mario Pesch und »Der Tanzende Teddy« von Julia Kleeberger.

## 4.1 Was sind eigentlich Sensoren?

Sensoren sind so ähnlich wie deine Sinnesorgane, mit denen du riechen, schmecken und fühlen kannst. Nur dass es sich hierbei um kleine elektronische Bauteile handelt. Sensoren können messen, wenn sich etwas in ihrer Umwelt verändert, wie zum Beispiel die Temperatur oder Helligkeit. Außerdem wandeln sie diese so genannten Messwerte in Zahlen von 0 – 255 um. Wie du dir vorstellen kannst, ist 0 der kleinste Wert – z.B. wenn es dunkel ist – und 255 der größte, wenn es ganz hell ist. Mit diesen Messwerten kannst du als Programmierer arbeiten und Programme schreiben, die Temperatur, Geschwindigkeit oder Helligkeit verarbeiten. Auf dem Calliope mini sind direkt mehrere solcher Sensoren:

- ▶ Temperatursensor
- ▶ Beschleunigungs- und Bewegungssensor
- ▶ Kompass
- ▶ Mikrophon

In diesem Kapitel lernst du alle Sensoren des Calliope minis kennen. Du wirst sehen, welche tollen Spiele du damit programmieren kannst.

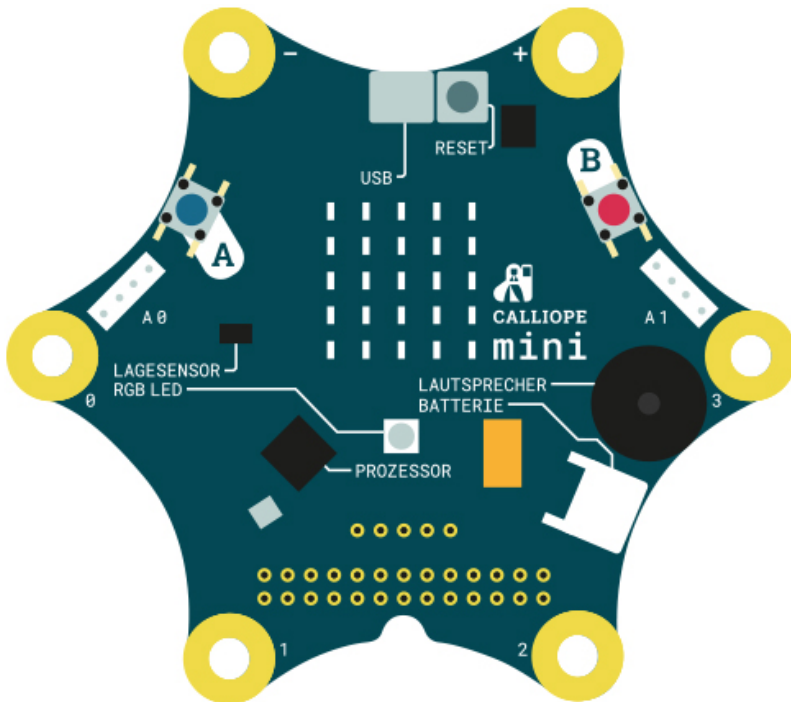


Abb. 4–1 Der Calliope mini

## 4.1.1 Der Temperatursensor

Der Temperatursensor des Calliope mini misst die Temperatur des Prozessors. Du wirst also feststellen können, dass die Temperatur ansteigt, wenn der Calliope mini ein komplexes Programm ausführen muss. Du kannst die Temperatur aber auch ein wenig beeinflussen, indem du einen Finger auf den Prozessor legst und diesen so durch deine Körperwärme erwärmst. Den Programmierbefehl für das Auslesen der Temperatur findest du in der Gruppe »Eingabe«. Dort kannst du den Befehl »Temperatur« verwenden, um die Temperatur in Grad Celsius zu bekommen. Ein kleines Beispiel kannst du in Abbildung 4–2 sehen.

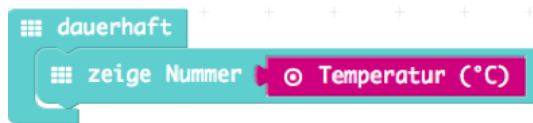


Abb. 4–2 Programmierbefehle zum Anzeigen der Temperatur

## 4.1.2 Der Beschleunigungs- und Bewegungssensor

Der Beschleunigungssensor des Calliope kann dir helfen, verschiedene Projekte umzusetzen. Der Sensor kann die Beschleunigung des Calliope-Boards in verschiedene Richtungen messen. Beschleunigung kennst du vom Fahrrad- oder Autofahren und beschreibt den Prozess des schneller oder auch langsamer werden. Wenn du also kräftig in die Pedale deines Fahrrades trittst, beschleunigst du. Ziehst du nun die Bremsen, wirst du wieder langsamer. Gemessen wird dies dann als negative Beschleunigung.

Der Beschleunigungssensor im Calliope kann die Beschleunigung in drei verschiedene Richtungen messen. Die drei verschiedenen Richtungen kannst du dir am besten vorstellen, wenn du den Calliope mini flach auf den Tisch legst. Die x- und y-Achse kannst du beeinflussen, wenn du den Calliope mini nach rechts oder links bzw. oben oder unten bewegst. Wenn du den Calliope mini vom Tisch anhebst, führst du eine Beschleunigung in Richtung der y-Achse aus. Den Programmierbefehl (Abbildung 4–3) für den Beschleunigungssensor findest du wieder in der Gruppe »Eingabe«. Dort kannst du auch einstellen, in welche Richtung die Beschleunigung gemessen werden soll. Wählst du »Stärke« aus, so wird die gesamte Beschleunigung in alle drei Richtungen gemessen.



Abb. 4–3 Der Programmierblock für den Beschleunigungssensor und seine Auswahlmöglichkeiten

Mithilfe des Beschleunigungssensors können auch Bewegungen erfasst werden. Im Editor findest du dazu den Programmierbefehl »wenn geschüttelt« (siehe Abbildung 4–4) in der Gruppe »Eingabe«. Das Schütteln des Calliope-Boards wird als Beschleunigung in verschiedene Richtungen registriert und weitergeleitet als »schütteln«.

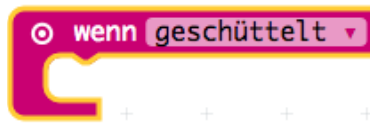


Abb. 4–4 Den Programmierbefehl »wenn geschüttelt« findest du in der Gruppe »Eingabe«

### 4.1.3 Der Helligkeitssensor

Der Helligkeitssensor im Calliope mini ist in das LED-Display integriert. Du kannst damit die Helligkeit messen und erhältst Messwerte zwischen 0 und 255. Wenn der Calliope mini 0 anzeigt, ist es ganz dunkel, bei 255 sehr hell. Im Vergleich zum Temperatursensor erhältst du hier also keinen genauen Messwert, aber du kannst trotzdem tolle Projekte erstellen, die mit Lichtmessung arbeiten. Den Programmierbefehl findest du wieder in der Gruppe »Eingabe«. In Abbildung 4–5 ist ein kurzes Beispiel, das dir zeigt, wie du dir die Helligkeit auf dem LED-Display anzeigen lassen kannst. Probiere doch mal, mit einer Taschenlampe die Helligkeit zu erhöhen oder den Sensor mit etwas abzudecken, um die Helligkeit zu verringern.



Abb. 4–5 Der Programmierblock für den Helligkeitssensor

## 4.2 Die Wasserwaage

In diesem Projekt baust du aus dem Calliope mini eine Wasserwaage. Was ist eine Wasserwaage und wozu braucht man sie, fragst du dich vielleicht. Hier eine kurze Erklärung: Wenn du schon einmal ein Bild aufgehängt hast, merkst du schnell, dass es gar nicht so einfach ist, es gerade aufzuhängen. Schließlich soll es doch nicht schief hängen, sondern schön aussehen. Und dann gibt es immer noch jemanden, der behauptet, es hänge doch schief. Mithilfe einer Wasserwaage kannst du dieses Problem lösen. Sie zeigt dir an, ob das Bild (oder etwas anderes) gerade ausgerichtet ist.

Du hast bestimmt schon einmal gesehen, wie deine Eltern eine Wasserwaage benutzt haben. In der Mitte der Waage siehst du etwas Wasser und eine Art Luftblase. Wenn diese Luftblase mittig innerhalb eines markierten Feldes ist, ist das Bild gerade ausgerichtet.

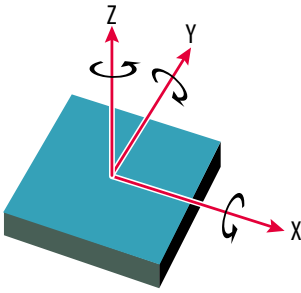


Abb. 4–6 Ein schiefes Bild mithilfe des Calliope mini gerade hängen

Für das Projekt verwendest du natürlich kein Wasser wie in einer traditionellen Wasserwaage, sondern den Beschleunigungssensor des Calliope mini.

### Was ist Beschleunigung?

Du möchtest prüfen, ob der Calliope mini gerade ausgerichtet ist? Richtig! Wenn du in einem Auto sitzt und es losfährt, wirst du leicht in den Sitz gedrückt, also beschleunigt – nichts Neues für dich. Aber auch gerade jetzt beim Lesen dieses Buches bist du einer Beschleunigung ausgesetzt – und das ganz, ohne dass du es bemerkst. Diese Beschleunigung richtet sich nach unten (du kennst sie als Anziehungskraft – Gravitation). Das ist der Grund, weshalb du nicht in der Luft herumschwebst oder warum ein Apfel vom Baum fällt. Ohne diese Beschleunigung wärest du schwerelos, wie ein Astronaut im Weltraum.



Diese dauerhafte Beschleunigung auf der Erde kannst du jetzt auch verwenden, um zu prüfen, wie der Calliope mini ausgerichtet ist. Das Calliope-Board kann dazu die Beschleunigung in drei Richtungen messen, diese Richtungen werden in der Fachsprache Achsen genannt. Wir haben rechts und links, vorne und hinten sowie oben und unten Achsen. Dies entspricht den Achsen X, Y und Z.

Abb. 4-7 Die drei Achsen des Beschleunigungssensors

Wenn der Calliope mini gerade ausgerichtet ist, liefert dir der Sensor für die x- und y-Achse keine Beschleunigung und für die z-Achse  $1g$ , was die Beschleunigung (Gravitation) ist, die dich auf dem Boden hält. Wenn der Calliope mini nun gekippt wird, also nicht mehr gerade ausgerichtet ist, verändert sich die Beschleunigung auf den Achsen. Die Beschleunigung auf der z-Achse nimmt ab, die der x- und y-Achse erhöht sich. Dieses Verhalten kannst du nun verwenden, um die Ausrichtung des Calliope zu erkennen.

## 1g

Mit dem Formelzeichen  $g$  wird der Grad von Beschleunigung beschrieben. Alle stehenden Objekte auf der Erde haben dabei eine Beschleunigung von ungefähr  $1g$  nach unten, also zur Mitte der Erde. Im Auto ist eine Beschleunigung von bis zu  $0,2g$  möglich, auf einer Achterbahn bis zu  $4g$ . Ab  $5g$  wird es für den Menschen sehr gefährlich, da der Körper durch die hohen Kräfte zusammengedrückt wird.

### 4.2.1 Die x-Achse wird gemessen

Mit dieser Information kannst du jetzt auch schon anfangen, dein Programm zu schreiben. Du beginnst mit der x-Achse. Wenn die Beschleunigung  $0$  ist, soll die LED grün leuchten, sonst rot.



Abb. 4-8 Die x-Achse wird gemessen

Eigentlich sollte die LED jetzt grün leuchten, sobald der Calliope mini gerade ausgerichtet ist. Tut er aber leider nicht! Um den Fehler zu finden, kannst du dir die Beschleunigung mithilfe eines neuen Programms auf dem Display ausgeben. Erstell dazu ein neues Programm und füge die folgenden Blöcke hinzu:



Abb. 4-9 Ein kleines Hilfsprogramm soll dir die Fehlersuche erleichtern

Versuch jetzt, durch Kippen des Calliope mini eine 0 auf dem Display zu erreichen. Ziemlich schwierig, oder? Das liegt daran, dass der Sensor »rauscht«. So nennt man es, wenn ein Sensor keinen genauen Wert zurückgibt, sondern um einen Wert herumwackelt.

Du darfst in deinem Programm also nicht nur prüfen, ob der Wert der Zahl 0 entspricht, sondern auch, ob er ein wenig unter oder über 0 liegt. Du musst also einen Bereich festlegen, in dem die LED leuchtet, der als »nah genug an 0« gesehen werden kann. Dieser Bereich sollte von -30 bis 30 gehen. Zum Vergleich: Beschleunigung kann Werte zwischen -200 und 200 annehmen, daher ist der gewählte Bereich für dein Ziel exakt genug.

Verwende den »größer«- und »kleiner«-Vergleich, um diesen Bereich in deinem Programm anzugeben.

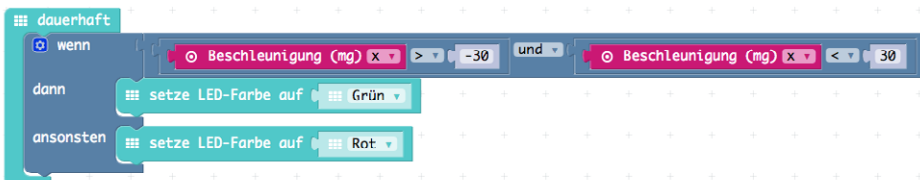


Abb. 4-10 Das sollte schon besser funktionieren

Du prüfst jetzt, ob die Beschleunigung über -30 ist oder unter 30. Wenn beides zutrifft, soll die LED grün leuchten. Übertrage das Programm auf deinen Calliope mini, und mit ein wenig Hin- und Herkippen solltest du es schaffen, dass die LED grün leuchtet. Teste es einmal aus!

### 4.2.2 Die y-Achse wird gemessen

Eine Kleinigkeit fehlt jetzt noch. Aktuell prüfst du nur die Beschleunigung auf der x-Achse. Um die Wasserwaage komplett zu machen, fehlt noch die y-Achse. Falls du vergessen hast, welche Achse wie gelegen ist, schau noch einmal oben in der Abbildung 4–7 nach.

Du willst also, dass die LED nur grün leuchtet, wenn die Beschleunigung auf der x- und y-Achse zwischen -30 und 30 ist. Dazu kannst du eine zweite Verzweigung verwenden, in der du die gleichen Bedingungen für die y-Achse aufstellst. Das Ganze sollte dann wie folgt aussehen:

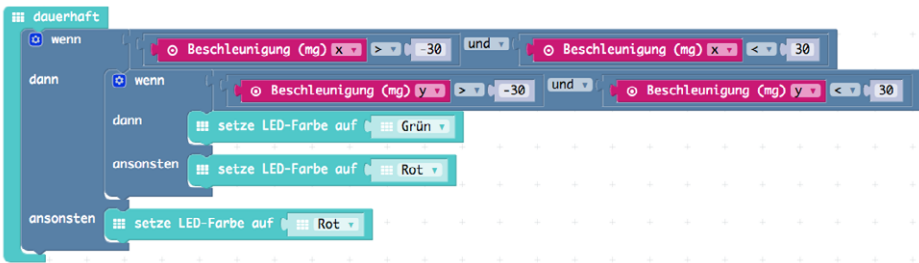


Abb. 4–11 Unser fertiges Programm

Wichtig ist hierbei, dass du die LED auch dann rot leuchten lässt, wenn nur eine Achse alle Bedingungen erfüllt, denn dann ist dein Calliope mini noch nicht gerade ausgerichtet.

Jetzt geht es ans Testen. Testen ist eine ganz wichtige Sache bei der Programmierung. Leg dafür den Calliope mini auf irgendeinen Gegenstand, den du gerade ausrichten möchtest, z.B. einen Bilderrahmen. Wenn du den Calliope mini jetzt ausrichtest, sollte die LED grün leuchten, sobald er gerade ist.

## 4.3 Die Calliope-Alarmanlage

### 4.3.1 Alles, was du über die Calliope-Alarmanlage wissen musst

In den vorherigen Kapiteln hast du bereits erfahren, was ein Sensor ist und welche tollen Spiele sich damit programmieren lassen. Aber hättest du gedacht, dass du mit dem Calliope mini auch eine Alarmanlage bauen kannst? Um dein Zimmer, Baumhaus oder deine Höhle besser abzusichern, ist der Calliope mini genau richtig, denn es befindet sich bereits alles auf dem Board. Mithilfe des Lichtsensors, des Lautsprechers und natürlich mit dem richtigen Programm wirst du immer alarmiert, wenn jemand an deiner Alarmanlage vorbeiläuft. Wie das genau funktioniert, erfährst du in den nächsten Schritten.

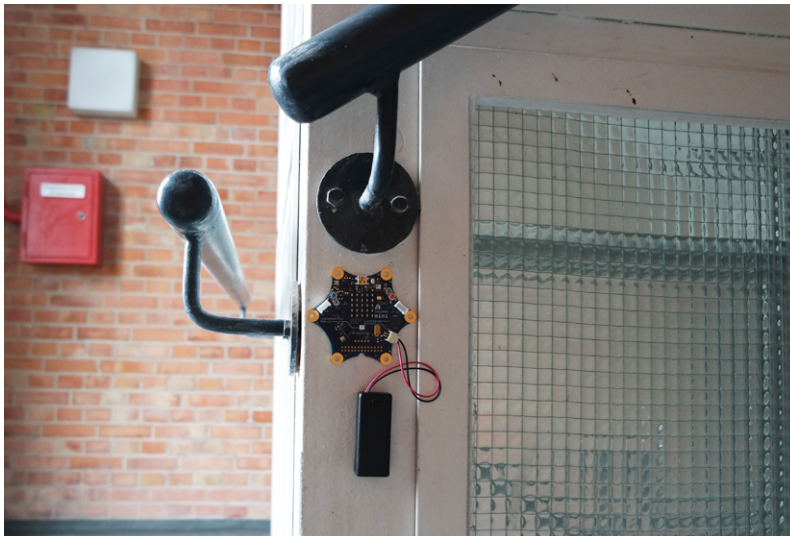


Abb. 4-12 Die installierte Alarmanlage

### 4.3.2 Materialien und Werkzeuge

Wie viele?	Was	Anmerkung
1	Calliope	
1	Battery-Pack	
2	Batterie	
1	Doppelseitiges Klebeband	5 cm

Tab. 4-1 Materialien für die Calliope-Alarmanlage

## Seit wann gibt es eigentlich Alarmanlagen?

Die erste moderne »Gefahrenmeldeanlage« hatte der Tüftler Augustus Russell Pope 1853 in der Nähe von Boston in den USA erfunden. Ein paar Jahre später wurde diese elektromagnetische Alarmanlage sogar zur Überwachung von Gebäuden in New York eingesetzt. In den 1970er-Jahren bekam die Alarmanlage zusätzlich einen Bewegungsmelder und wurde ab den 1980er-Jahren in Europa immer beliebter. Mittlerweile gibt es drahtlose Funkalarmanlagen mit Videoüberwachung und Internetanschluss, die sich per Smartphone steuern lassen. Was wohl Augustus Russell Pope dazu sagen würde?

### 4.3.3 Wie funktioniert die Calliope-Alarmanlage?

Es gibt viele Arten von Alarmanlagen, die ganz unterschiedlich funktionieren und verschieden aufgebaut sind. Aber wie du sicherlich weißt, erfüllen sie immer den gleichen Zweck: Sie werden zur Überwachung eingesetzt.

Auch mit dem Calliope-Board lässt sich mit dem Lichtsensor und Lautsprecher eine Alarmanlage bauen, die Bewegungen feststellt und dann einen Alarm auslöst. Und das funktioniert so: Wenn du die Alarmanlage anstellst, misst der Lichtsensor die Helligkeit. Wenn jemand an der Alarmanlage vorbeiläuft, wirft die Person einen Schatten auf das Calliope-Board, und es wird dunkler. Der Lichtsensor bemerkt den Unterschied und schickt ein Signal an den Lautsprecher, das ihn laut aufpiepsen lässt. So weißt du direkt Bescheid, wenn jemand versucht, heimlich einzusteigen. Das klingt spannend? Dann überleg doch mal, wie die passende Programmierung dazu aussehen könnte.

### 4.3.4 Der Lichtsensor

Wie du vielleicht am Anfang des Kapitels gelesen hast, können Sensoren Umweltveränderungen messen und in eine Zahl umwandeln. Wie der Name bereits sagt, kann der Lichtsensor die Stärke, also die Intensität des Lichts messen. Der Lichtsensor auf dem Calliope mini misst also die Helligkeit und wandelt sie in eine Zahl von 0 bis 255 um.

### 4.3.5 ... und jetzt zum Programmieren

Wie du in den vorherigen Projekten bereits gesehen hast, kannst du deinen Calliope mini mit dem PXT-Editor programmieren. Wenn du dir noch nicht ganz sicher bist, wie das alles funktioniert, dann schau dir noch einmal die Anleitung in Abschnitt 2.2 an. Oder kennst du das Programmierwerkzeug bereits? Wunderbar, dann kann es direkt losgehen.

### 4.3.6 Wie soll sich deine Alarmanlage einschalten?

Ganz zu Beginn des Programms kannst du überlegen, wie deine Alarmanlage eingeschaltet werden soll. Immer wenn dein Calliope mini mit Strom versorgt wird? Oder nur, nachdem du einen bestimmten Knopf gedrückt hast?

Aus den Gruppen »Grundlagen« oder »Eingabe« kannst du dir einen Start-Programmierbefehl aussuchen, der bestimmt, wie die Alarmanlage eingeschaltet wird. In diesem Beispiel ist es der Programmierbefehl beim Start, den du in der Gruppe »Grundlagen« findest. Immer wenn dein Calliope eingeschaltet wird, schaltet sich automatisch die Alarmanlage ein. Zieh den Programmierbefehl wie gewohnt in das Programmierfeld.

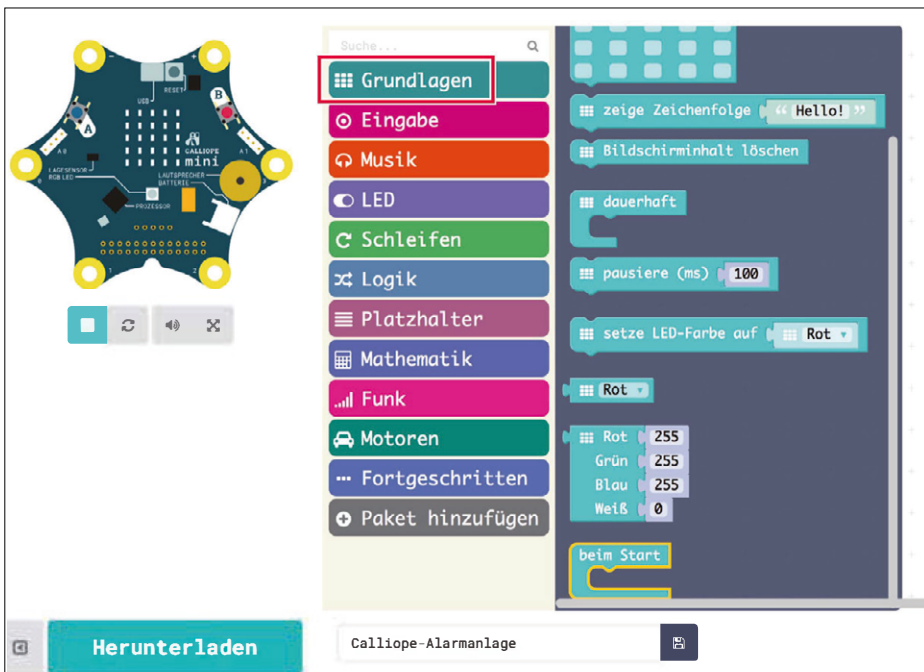


Abb. 4–13 Start des Programms

### 4.3.7 Die Schleife programmieren

Im nächsten Schritt baust du eine Schleife in dein Programm ein. Mit der Schleife überprüfst du wieder und wieder, ob der von dir programmierte Zustand eingetroffen ist oder nicht. Der sogenannte Zustand, den du programmiert, ist einfach die Tatsache, ob die Helligkeit gleich bleibt oder ob ein Schatten auf dein Calliope-Board fällt.

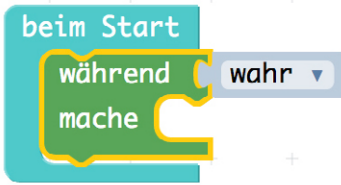


Abb. 4–14 Schleife auswählen

Hierfür öffnest du die Schleifen-Gruppe, in der du den Schleifen-Block »während-wahr-mache« findest.

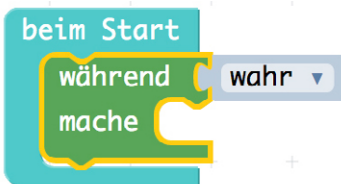


Abb. 4–15 Ausgewählte Schleife

Diesen Block ziehst du dir in dein Programmierfeld und platzierst ihn in deinen Start-Block. Das sieht am Ende so aus:

### 4.3.8 Wann ist es hell? Wann ist es dunkel?

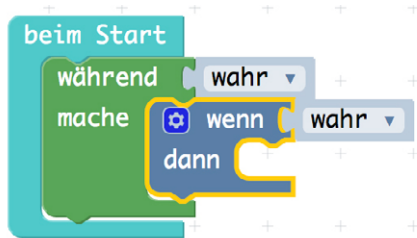
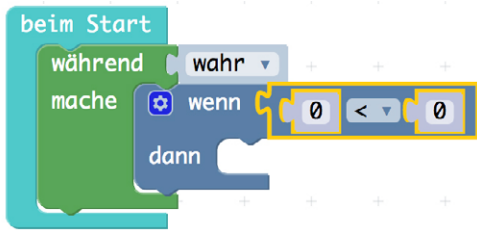


Abb. 4–16 Erster »Logik«-Block

Oben haben wir den Zustand schon angesprochen. Den wollen wir jetzt genauer betrachten. Findest du in der Gruppe »Logik« den »wenn-wahr-dann«-Block? Diesen Block ziehst du im vierten Schritt in deine »während-wahr-mache«-Schleife.

Jetzt musst du das »wenn-dann« der Alarmanlage bestimmen, und das könnte so aussehen: »wenn« der Lichtsensor Dunkelheit misst, »dann« soll der Lautsprecher anfangen zu piepsen. Dein Calliope mini kann zwar sehr viel, allerdings weiß er nicht von alleine, wann es dunkel oder hell ist. In den »Logik«-Programmierbefehl kommt also noch ein weiterer Programmierbefehl hinein, mit dem der Lichtsensor die Helligkeit misst und mit deinem angegebenen Wert (wann es hell



bzw. dunkel ist) vergleicht. Öffne hierzu erneut die »Logik«-Gruppe und zieh dir den »kleiner-als«-Block, der so aussieht »0 < 0«, in die hellblaue »wahr«-Zeile.

Abb. 4-17 »Zweiter« Logik-Block

Ersetze die erste 0 mit dem Block »Lichtstärke«, den du unter »Eingabe« findest. So leicht kannst du mit dem Lichtsensor die Lichtstärke messen! Die zweite 0 wird durch deine Helligkeitsgrenze ersetzt. Deine Helligkeitsgrenze ist eine Zahl zwischen 0 und 255, die bestimmt, wann es dunkel genug ist, um den Alarm auszulösen. 0 ist hierbei totale Finsternis und 255 ganz hell.

Je nachdem, wo deine Alarmanlage angebracht wird, variiert diese Zahl ein wenig. Am besten probierst du einfach aus, welcher Wert am besten funktioniert.

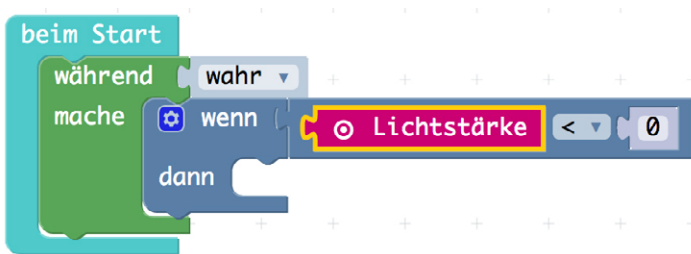


Abb. 4-18 Einsetzen des »Lichtstärke«-Blocks

### 4.3.9 Den Alarmton programmieren

Dein Programm kann bereits die Lichtstärke messen und errechnen, ob der Wert unter oder über deiner Helligkeitsgrenze liegt. Was jetzt noch fehlt, ist der eigentliche Alarm. Zum Glück hat der Calliope mini einen Lautsprecher, der unterschiedliche Töne ausgeben kann. In diesem Schritt programmierst du das Alarmgeräusch.

Öffne die Gruppe »Musik« und zieh den Programmierbefehl »spiele-Note« hinter deine »dann«-Logik. In dem ersten der zwei heller hervorgehobenen Felder kannst du dir deinen Lieblingston aussuchen. In diesem Fall wird beim Auslösen des Alarms der Ton C abgespielt.



Abb. 4–19 Alarmton programmieren

### 4.3.10 Die Lichtstärke im Simulator

Geschafft! Nachdem du den orangefarbenen »Musik«-Programmierbefehl eingefügt hast, bist du fertig mit dem Programm. Jetzt kannst du deine Calliope-Alarmanlage im PXT-Simulator testen. Oben links neben dem Calliope mini befindet sich ein grau-gelber Kreis mit einer Zahl darunter. Diese Zahl ist die Lichtstärke, die dein Lichtsensor im Simulator misst. Per Doppelklick auf den Kreis kannst du eine Lichtstärke zwischen 0 und 255 einstellen. Und hörst du schon den Alarmton?

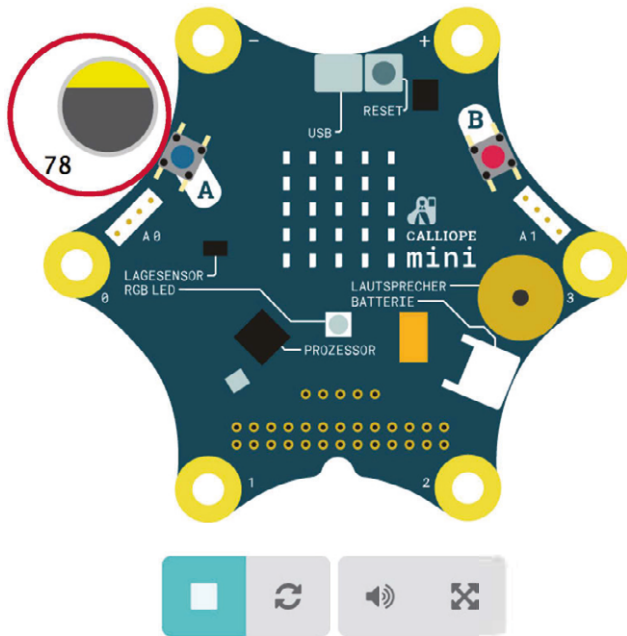


Abb. 4–20 Einstellung der Lichtstärke

## Unsichtbares Licht

Wusstest du, dass es auch unsichtbares Licht gibt? Als unsichtbares Licht werden Infrarotlicht und ultraviolette Strahlung bezeichnet. Das menschliche Auge kann diese Lichtarten nicht sehen, aber sie lassen sich mit Infrarot- bzw. UV-Sensoren messen. Diese beiden Sensoren befinden sich zwar nicht auf dem Board, aber du kannst sie an einen der Daten-Pins des Calliope mini anschließen, um die Intensität der unsichtbaren Strahlen zu messen.

Wenn du mit deinem Alarmanlagen-Programm zufrieden bist, lädst du das Programm wie gewohnt über die Schaltfläche »Herunterladen« auf deinen Computer. Schließ jetzt deinen Calliope über das USB-Kabel an den Computer an und kopier das Programm auf dein Board.

### 4.3.1 Geschafft!

Nachdem du die Alarmanlage erfolgreich programmiert hast, musst du noch ein geeignetes Versteck finden. Mit doppelseitigem Klebeband lassen sich der Calliope mini und die dazugehörige Batteriehalterung gut befestigen. Achte bei der Installation darauf, dass der Calliope und somit der Lichtsensor nicht abdeckt wird.

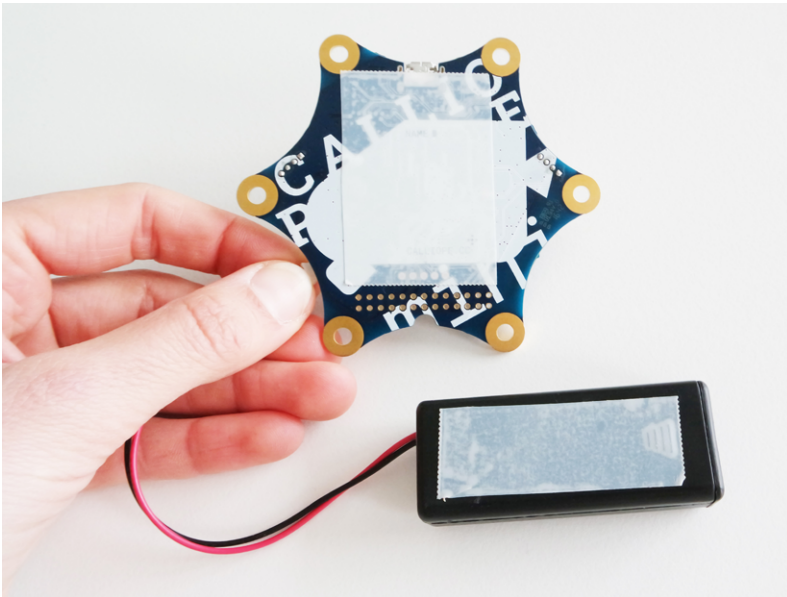


Abb. 4–21 Doppelseitiges Klebeband auf der Rückseite

## Auslösen der Alarmanlage nur bei Erwachsenen

Wie kannst du verhindern, dass die Alarmanlage jedes Mal piepst, wenn du daran vorbeiläufst? Ein kleiner Tipp: Bring das Calliope-Board weiter oben an, so dass du keinen Schatten auf den Lichtsensor wirfst, sehr wohl aber alle Erwachsenen, die daran vorbei laufen, da sie größer sind als du.

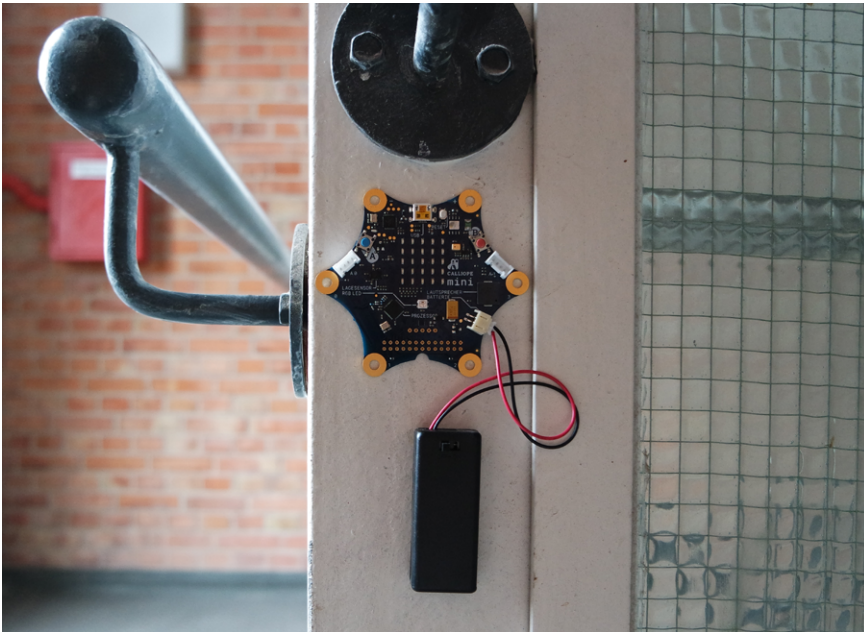


Abb. 4-22 Die versteckte Calliope-Alarmanlage

Um dein Versteck bestmöglich abzusichern, kannst du deine Alarmanlage selbstverständlich weiterentwickeln. Zum Beispiel könntest du den Alarm so programmieren, dass der Calliope mini viel länger piepst, wenn jemand den Alarm auslöst. Oder möchtest du vielleicht, dass der Alarm nur per Knopfdruck wieder abgeschaltet werden kann und nicht automatisch? Wie du siehst, gibt es viele Möglichkeiten, an dem Programm weiterzuarbeiten.

Viel Spaß beim Absichern deiner Räume!

## 4.4 Das Farbthermometer

### 4.4.1 Farbe und Temperatur?

Farbthermometer? Was hat denn eine Farbe mit einem Thermometer zu tun? In diesem Projekt machst du Temperatur durch Farbe sichtbar – zuerst auf dem LED-Display und anschließend mit der RGB-LED. In Abschnitt 4.1.1 hast du schon gelesen, wie der Calliope mini die Temperatur misst und wie du sie auch etwas beeinflussen kannst. Jetzt wirst du lernen, wie du die Farbe der RGB-LED beliebig verändern kannst und wie du den Calliope erste eigene Entscheidungen treffen lassen kannst.

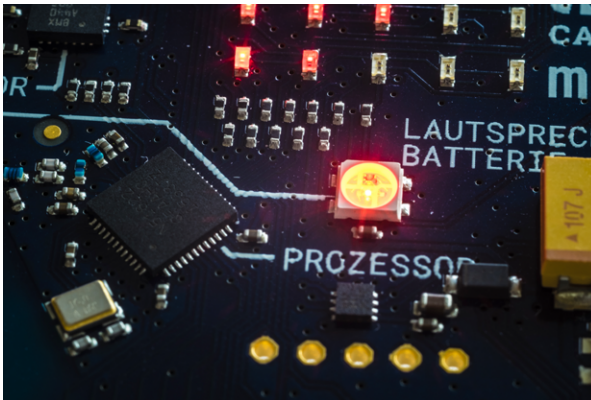


Abb. 4–23 Die Temperatur des Prozessors wird in Farbe umgewandelt!

### 4.4.2 Calliope mini unter der Lupe

Für das Farbthermometer verwendest du die RGB-LED (siehe dazu den Kasten »RGB-LED«), um die Temperatur anzeigen zu lassen. Du wandelst also eine Zahl in ein sichtbares Ergebnis um. In Abschnitt 4.1.1 wurden bereits einige Informationen zum Temperatursensor des Calliope mini gegeben. Du erinnerst dich vielleicht daran, dass die Temperatur, die der Calliope misst, die Prozessortemperatur ist. Wenn du die Temperatur ein wenig verändern möchtest, kannst du also mit dem Finger auf den Prozessor fassen. Keine Angst, da kann nichts passieren! Alternativ kannst du mithilfe des Calliope-Simulators links im PXT-Editor die Temperatur verändern.

## RGB-LED

Der Calliope mini besitzt eine RGB-LED. RGB bedeutet Rot-Grün-Blau. Die RGB-LED kann mithilfe dieser drei Farben alle Farben anzeigen. Du kennst das vielleicht vom Malen mit Wasserfarbe, dass du auch dort verschiedene Farben mischen kannst und eine neue entsteht. Die verschiedenen Farben kannst du mithilfe von Zahlen zwischen 0 und 255 einstellen. Zwei Beispiele:



Die beiden Beispiele zeigen dir, wie du die Farben mischen kannst. Der Editor bietet dir aber auch eine große Auswahl an fertigen Farben an, die du aus der Gruppe »Grundlagen« auswählen kannst.

### 4.4.3 Die Programmierung

Im ersten Schritt beginnst du im Editor damit, die Temperatur des Calliope mini auszulesen. Dazu verwendest du den Programmierbefehl »Temperatur« aus der Gruppe »Eingabe«. Du benötigst allerdings noch eine Ausgabe. Um ein Gefühl dafür zu bekommen, wie hoch die Temperatur ist, solltest du dir zu Anfang die Temperatur auf dem LED-Display ausgeben lassen. Verwende dazu den Programmierbefehl »zeige Nummer« aus der Gruppe der »Grundlagen«.

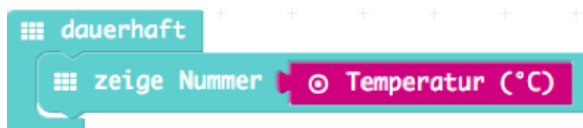


Abb. 4–24 Anzeige der Temperatur auf dem LED-Display

Dein Calliope mini zeigt dir nun in Laufschrift die aktuelle Temperatur an. Versuch nun, durch Auflegen des Fingers die Temperatur zu beeinflussen.

Jetzt wandelst du die Temperatur noch in eine Farbe um. Erinnerst du dich noch an den »Wenn, dann«-Befehl aus Abschnitt 3.1. Mithilfe dieser Bedingung

kannst du nun die RGB-LED abhängig von der Temperatur in verschiedenen Farben leuchten lassen.

In der Gruppe »Logik« findest du den Programmierbefehl »wenn, dann«. Die RGB-LED soll zwei verschiedene Temperaturen anzeigen können. Wenn die Temperatur über 25 Grad ist, soll die LED rot leuchten, wenn die Temperatur unter 25 ist, soll sie blau leuchten. Für diesen Befehl benötigen wir einen sogenannten »Operator«. Eine kurze Erläuterung zu verschiedenen Operatoren findest du im nachfolgenden Kasten »Operatoren«.

## Operatoren

Operatoren werden in vielen Situationen beim Programmieren benötigt. Mithilfe von Operatoren können Bedingungen überprüft oder auch Werte verglichen werden. Einige dieser Operatoren kennst du wahrscheinlich auch schon aus dem Mathematik-Unterricht:

Das »= $\leftarrow$ «-Zeichen: Hiermit kannst du den Calliope einen Programmierbefehl ausführen lassen, wenn zwei Werte gleich groß sind.

Das » $\neq$ «-Zeichen: Hiermit kannst du den Calliope einen Programmierbefehl ausführen lassen, wenn zwei Werte unterschiedlich groß sind.

Das »<«-Zeichen: Mithilfe dieses Zeichens kannst du den Calliope zwei Werte vergleichen lassen. Wenn die Spitze des Symbols auf den kleineren Wert zeigt, führt der Calliope den nächsten Befehl aus.

Das » $\leq$ «-Zeichen: Dieses Zeichen ist eine Erweiterung des »kleiner«-Zeichens ( $\leq$ ) und schließt auch Werte ein, die kleiner und gleich groß sind ein.

Das »>«-Zeichen: Mithilfe dieses Zeichens kannst du den Calliope zwei Werte vergleichen lassen. Die Öffnung des Symbols zeigt diesmal auf den größeren Wert.

Das » $\geq$ «-Zeichen: Dieses Zeichen ist eine Erweiterung des »größer«-Zeichens ( $\geq$ ) und schließt auch Werte ein, die größer oder gleich groß sind.

Alle diese Operatoren findest du in der Gruppe »Logik«.

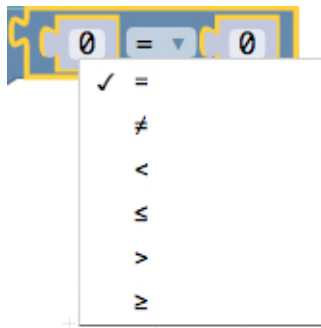


Abb. 4–25 Übersicht über die Operatoren

Mithilfe der Operatoren und des Programmierbefehls »wenn, dann« kannst du deinen Calliope mini nun anhand der Temperatur entscheiden lassen, welche Farbe die RGB-LED ausgibt.



Abb. 4–26 Wenn-dann-Bedingung

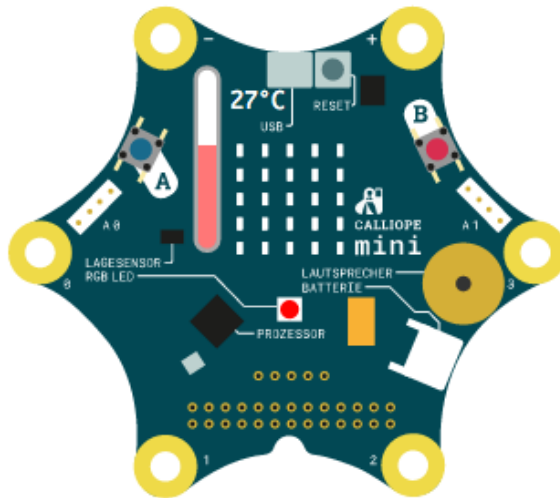


Abb. 4–27 Der Calliope-Simulator zeigt dir eine Temperatur, die du verändern kannst.

Der Simulator auf der linken Seite des Editors zeigt dir die Temperatur an, und auch die RGB-LED leuchtet in der Farbe Rot. Du kannst mit der Maus die Temperatur verändern und einmal beobachten, was mit der Farbe der LED passiert. Du wirst feststellen, dass die LED immer rot bleibt. Du musst deinen Befehl noch erweitern, sodass die LED-Farbe geändert wird, wenn die Temperatur kleiner als 25 Grad Celsius ist. Klick dazu auf das kleine Zahnrad neben dem »wenn«. Um den Befehl zu erweitern, kannst du das »else« unter das »if« ziehen, und du wirst sehen, dass unterhalb des »dann« ein »ansonsten« auftaucht. Hier kannst du nun die LED-Farbe auf Blau setzen.

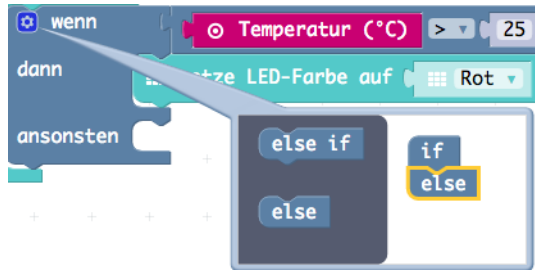


Abb. 4–28 Erweiterung der »wenn-dann«-Bedingung

Du hast es nun geschafft, dass der Calliope mini in Abhängigkeit von der Temperatur zwei verschiedene Farben anzeigt.

#### 4.4.4 Erweiterungen

Du kannst dein Programm natürlich erweitern und noch mehr Temperaturen und Farben einprogrammieren. Dazu wird der Programmierbefehl »else if« verwendet. Hast du schon versucht, deine eigene Lieblingsfarbe zu mischen? Wie das geht, kannst du in Abschnitt 4.4 weiter vorne nachlesen. Später, in Kapitel 8, gibt es auch noch Projekte zum Thema »Mapping«. Hier wirst du noch weitere Möglichkeiten kennenlernen, die Farbe der RGB-LED in Abhängigkeit von der Temperatur zu verändern.

#### 4.4.5 Was du gelernt hast

Und, erinnerst du dich noch an alle neuen Bestandteile des Calliope mini, die du in diesem Kapitel kennengelernt hast? Du kannst nun die Temperatur auf dem LED-Display anzeigen lassen, aber auch die Farbe der RGB-LED in Abhängigkeit von der Temperatur verändern. Und im besten Fall hast du dir sogar deine Lieblingsfarbe für die RGB-LED programmiert.

## 4.5 Tanzender Teddy

Ist dir schon mal in den Sinn gekommen, dass du den Mini dazu verwenden kannst, damit deine Kuscheltiere ihre Gefühle ausdrücken können? In diesem Projekt wirst du einen Teddy bauen, der sich freut, wenn man mit ihm tanzt: Er beginnt selbst eine eigene Melodie zu spielen und sein Herz fängt an zu leuchten.

Dafür verwendest du den in Abschnitt 4.1.2 erklärten Schüttelsensor des Calliope mini: Wenn du den Teddy in die Hand nimmst und hin und her bewegst, also tanzen lässt, spielt er eine Melodie und sein Herz beginnt zu schlagen.

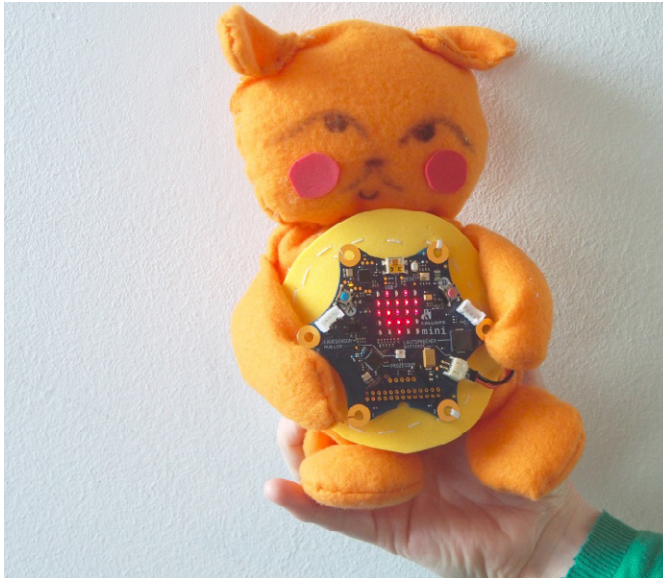


Abb. 4–29 Die Freude des Teddys ist groß: Sein Herz beginnt zu schlagen und eine Melodie erklingt, wenn du ihn bewegst, sodass er tanzt.

### 4.5.1 Die Programmierung

Für die Umsetzung des Projekts sind zwei Arbeitsschritte notwendig: die Programmierung deines Calliope mini und der Aufbau des Teddys.

Ich erkläre dir im ersten Schritt die hierfür notwendige Programmierung.

Um abzufragen, ob dein Calliope mini geschüttelt wurde, verwendest du den Programmierblock »wenn geschüttelt«.



Abb. 4–30 Der Programmierblock »wenn geschüttelt«

Jetzt kannst du dir überlegen, welches Ereignis ausgelöst werden soll, wenn der Teddy zu tanzen beginnt, also wenn er geschüttelt wird. Dabei sind deinem Ideenreichtum keine Grenzen gesetzt. In meinem Beispiel habe ich mich dafür entschieden, dreimal hintereinander eine kleine Tonfolge abzuspielen und das Herz des Teddys schlagen zu lassen.

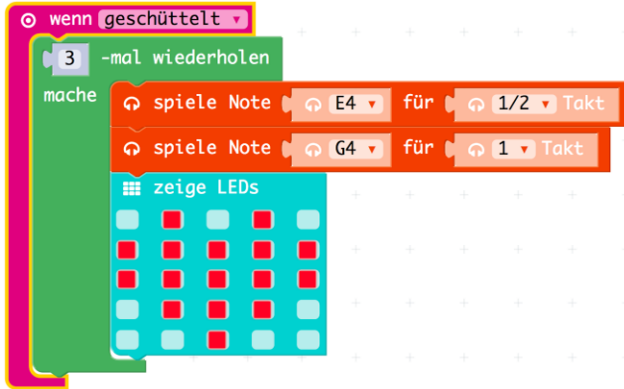


Abb. 4–31 Im Inneren des »wenn geschüttelt«-Programmierblocks legst du fest, was ausgeführt werden soll.

Damit die Display-Anzeige wieder gelöscht wird, wenn der Teddy aufhört zu tanzen, füge ich noch folgenden Programmierblock hinzu:

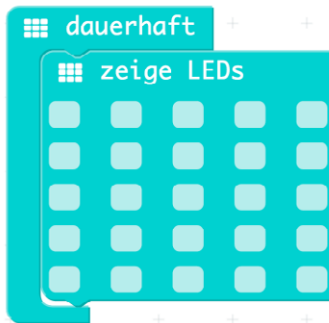


Abb. 4–32 Das Display wird gelöscht.

Um zu testen, wie dein Code funktioniert, kannst du auch den Simulator links im PXT-Editor nutzen. Durch das Drücken des Shake-Buttons wird die Schüttel-Funktion simuliert.

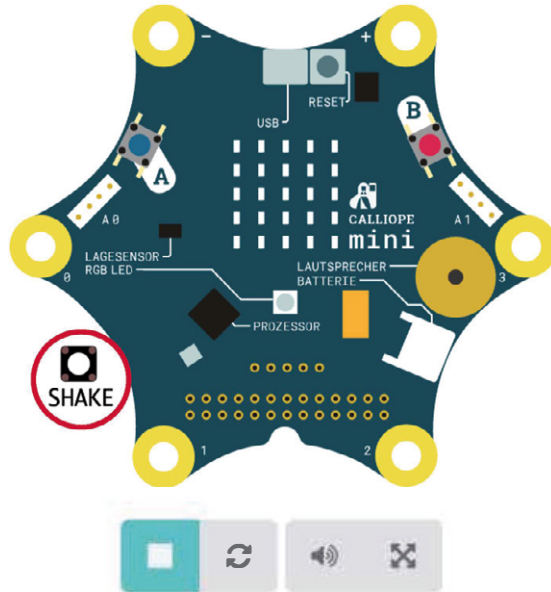


Abb. 4–33 Den Shake-Button drücken

Wenn du mit deinem Programm zufrieden bist, lädst du es auf dein Calliope-Board und trennst es vom USB-Kabel. Damit du deinen Calliope mini dann in den Teddy einbauen kannst und er auch unabhängig vom Computer funktioniert, braucht er eine eigene Stromversorgung. Hierfür benutzt du das Battery-Pack.

Damit hast du die Programmierung für den tanzenden Teddy erfolgreich abgeschlossen.

## 4.5.2 Der Aufbau des Teddys

Und jetzt geht's ans Gestalten des Teddys.

### Alternativen

Wenn du nicht selber nähen willst, kannst du auch schauen, ob sich zwischen deinen eigenen Kuscheltieren ein geeignetes findet, das du für das Projekt einsetzen kannst.

Alternativ kannst du den Teddy auch aus Pappe und Papier anfertigen. Übertrage dafür die Teile des Körpers aus einer Vorlage und den Bauch des Teddys auf festen Fotokarton. Für das Gesicht kannst du dünneres Papier verwenden.

Wenn du selbst einen Teddy bauen willst, benötigst du folgende Materialien:

Was	Anmerkung
Stoff für den Körper des Teddys	in einer Farbe deiner Wahl
Filz oder Moosgummi für den Bauch	in einer Farbe deiner Wahl
ggf. Filz- oder Moosgummireste	in Hellrosa und Rot
Füllwatte	
Garn	
ggf. Textilklebstoff	
Schere	
Nadel	

Tab. 4–2 Materialien zur Anfertigung des Stoffteddys

Zeichne auf Stoff oder Papier einen Teddy und schneide ihn an den Umrissen aus. Achte darauf, dass du auch für die Nähkante Platz einplanst.

Als Erstes legst du den Stoff für den Körper des Teddys so zusammen, dass seine beiden Außenseiten aufeinanderliegen, und überträgst die einzelnen Teile aus der Vorlage darauf und schneidest sie aus.



Abb. 4–34 Aus der Vorlage übernehmen

Jetzt nähst du Kopf, Körper, Beine und Ohren rechts auf rechts zusammen und lässt jeweils eine kleine Öffnung.



Abb. 4–35 Das Annähen

Wenn du fertig bist mit dem Nähen, krepelst du die einzelnen Teile durch die Öffnung um und füllst Kopf, Körper und Beine mit der Füllwatte. Danach nähst du die Öffnungen zu.



Abb. 4–36 Das Befüllen

Dann zeichnest du auf den Stoff für den Bauch des Teddys einen Kreis mit ca. 10 cm Durchmesser und schneidest ihn aus.

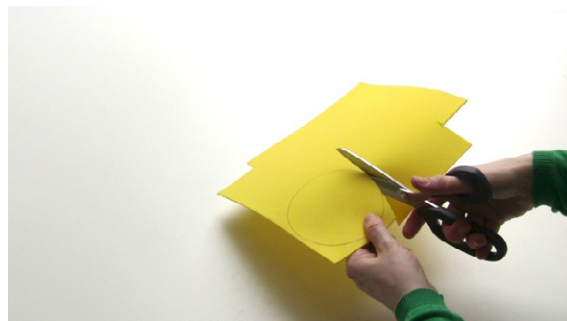


Abb. 4–37 Einen Kreis ausschneiden

Näh den Teddybauch an den Körper an. Lass dabei eine kleine Öffnung, hinter der du das Battery-Pack verstauen kannst.

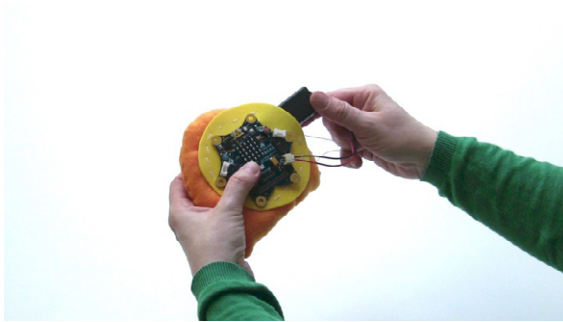


Abb. 4–38 Das Verstauen des Battery-Packs

Befestige jetzt den Calliope mini mit einigen kurzen Stichen durch die Pins an den Sternen-Ecken am Teddybauch.

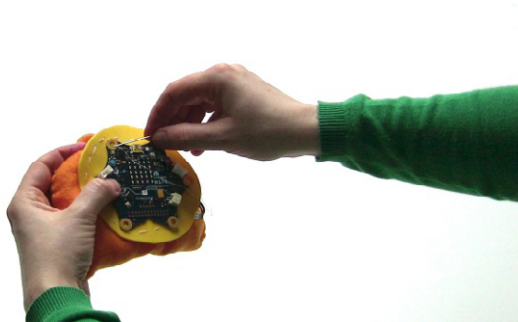
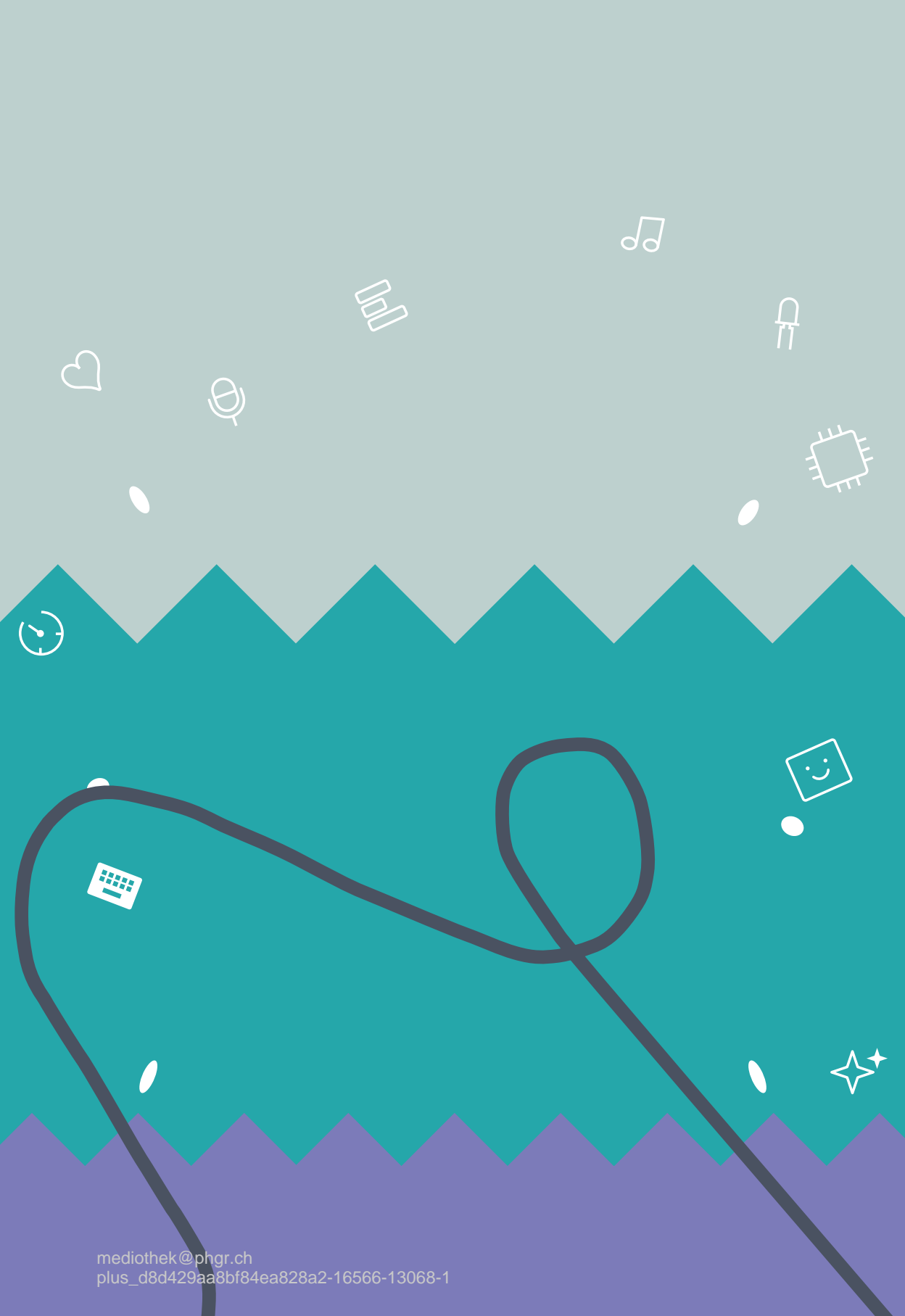


Abb. 4–39 Den Calliope mini befestigen

Jetzt nähst du noch Arme und Beine des Teddys am Körper an und die Ohren am Kopf. Dann malst du dem Teddy ein Gesicht, bestehend aus Augen, Mund und Nase. Wenn du magst, kannst du noch zwei Filzstücke für die Wangen mit Textilkleber befestigen.

Zum Schluss nähst du den Kopf mit kleinen Stichen an den Körper. Fertig. Wenn du deinen Teddy jetzt tanzen lässt, spielt er ein Lied und sein Herz fängt an zu schlagen.



# 5

## Dein rechter, rechter Platz ist leer

In diesem Kapitel geht es um Platzhalter, die man auch Variablen nennt.

Die Einführung zu diesem Kapitel wurde von Natalia Prost geschrieben. Das Projekt »Der Klickzähler« wurde von Natalia Prost erstellt, ebenso wie »Die digitale Sanduhr«. Das Calliope-Projekt »Das Metronom« stammt von Patrick Franken.

## 5.1 Variable »Platzhalter«

Eine der Befehlsgruppen im PXT-Editor heißt »Platzhalter«. Im nächsten Projekt, dem Klickzähler, lernst du, wie du einen neuen Platzhalter erstellen kannst. Hier erfährst du aber erst einmal, was ein Platzhalter überhaupt ist.

### Platzhalter = Variable

Der PXT-Editor benutzt das Wort »Platzhalter« und meint damit etwas, was die meisten Programmierer als »Variable« bezeichnen. Wenn etwas variabel ist, dann verändert es sich ständig (von alleine) oder kann zum Beispiel durch jemanden oder etwas verändert werden. Lass dich also nicht verwirren, wenn du schon einmal programmiert hast und weißt, was eine Variable ist. In der Informatik gibt es verschiedene Arten von Daten, die mit einer Variablen dargestellt werden können. Sie heißen »Datentypen«. Die wichtigsten Datentypen sind Zahlen und Texte verschiedener Längen.

Am einfachsten ist es, wenn du dir Platzhalter wie eine abschließbare Box vorstellst. Jeder Schlüssel öffnet nur genau eine Box, und auf dem Schlüssel steht der Name dieser einen Box. In der Box kann eine Zahl drin sein oder ein Wort oder auch ein ganzer Satz. Es können aber auch viele andere kleine Boxen drin sein.

Der Inhalt einer Box kann geändert werden, das heißt, Platzhalter können auch verändert werden: Sie sind variabel. Es gibt verschiedene Befehle, mit denen Platzhalter verändert werden können. Wir werden in diesem Kapitel nur mit Nummer-Platzhaltern arbeiten, also stell dir vor, dass in diese Boxen nur Zahlen hineinpassen.

Nun überlege dir, wie du Zahlen überhaupt verändern kannst. Wie kannst du mit einer oder mehreren Zahlen rechnen? Richtig: Du kannst zum Beispiel zwei Zahlen zusammenrechnen (+), voneinander abziehen (-), sie miteinander multiplizieren ( $\cdot$  oder  $\times$ ) und teilen ( $:$  oder  $\div$ ) – und natürlich noch vieles andere mehr.

Computerprogramme arbeiten oft so, dass Zahlen nach einer bestimmten Zeit oder nach einem bestimmten Befehl um 1 größer oder um 1 kleiner werden. Dafür braucht das Programm also keine zweite Zahl, sondern es reicht ein einziger Platzhalter, der so verändert wird, dass zum Beispiel +1 dazugerechnet wird. Der Zahlenwert des Platzhalters, also der Inhalt der Box, wird dadurch um 1 größer.

Vielleicht kannst du dir schon vorstellen, dass du mit Hilfe von Platzhaltern viele verschiedene Dinge zählen kannst.

## 5.2 Der Klickzähler

Viele Kinder spielen auf dem Schulhof gerne mit Springseilen, Basketbällen oder Tischtennisschlägern. Es ist spannend herauszufinden, wer die meisten Seilsprünge in einer Minute schafft oder am häufigsten den Ball auf dem Schläger springen lassen kann. Hast du schon einmal versucht, dabei ganz schnell mitzuzählen? Manchmal ist es schwierig, weil die Bewegung so schnell ist, dass du die Zahl gar nicht (im Kopf) zu Ende sprechen kannst. Bestimmt hast du dich dabei auch schon einmal verzählt.

In diesem Projekt lernst du, wie du mit dem Calliope mini ein Programm schreiben kannst, das jeden Klick – oder, besser gesagt, jeden Knopfdruck – auf eine der beiden Tasten ganz schnell zählen kann. Damit wirst du dich nicht mehr verzählen, weil du bei jedem Sprung nur ganz schnell klicken musst – und der Calliope mini zählt für dich.

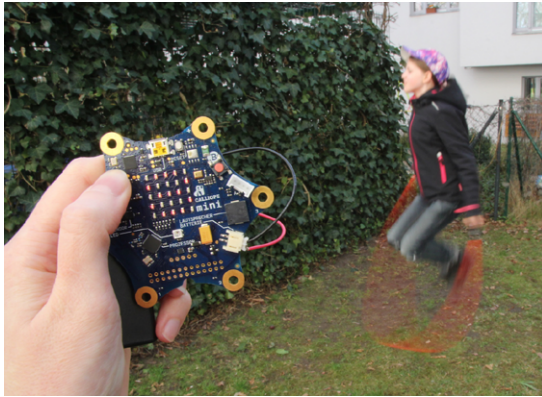


Abb. 5–1 Der Klickzähler zählt jeden Knopfdruck von dir.

Öffne den Editor und beginn ein neues Projekt, das du »Klickzähler« nennen kannst.

### 5.2.1 Platzhalter erstellen

Als Erstes musst du einen neuen Platzhalter für dein Projekt erstellen, damit darin gespeichert werden kann, wie viele Sprünge du oder deine Freunde geschafft haben. Wähle die Befehlsgruppe »Platzhalter« und klicke dort auf »Neuen Platzhalter anlegen«.



Abb. 5–2 Blöcke in der Befehlsgruppe »Platzhalter«

Es erscheint ein neues Fenster, in dem du dem neuen Platzhalter einen eigenen Namen geben kannst. Du kannst deinen Platzhalter nennen, wie du möchtest, aber überlege dir am besten einen guten, passenden Namen, damit du nicht vergisst, wofür du diesen Platzhalter benutzen willst. Wenn du Seilsprünge zählen möchtest, dann nenne deinen Platzhalter »sprünge«.

## Schreibweise

Du kannst den Namen auch mit einem großen Anfangsbuchstaben schreiben, wenn du möchtest. Es ist aber nicht wichtig, damit das Programm funktioniert.

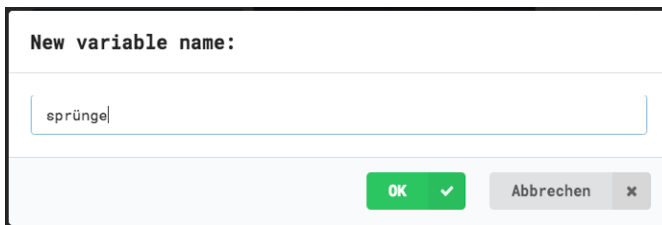


Abb. 5–3 Der neue Platzhalter heißt »sprünge«.

Klicke mit der Maus auf den grünen »OK«-Knopf. Dadurch wird dein Platzhalter gespeichert. Das kannst du daran erkennen, dass ein neuer Block mit dem Namen deines Platzhalters erschienen ist.



Abb. 5-4 Es gibt einen neuen »Platzhalter«-Block mit dem Namen »sprünge«.

Nun hast du einen neuen Platzhalter, und der Calliope mini weiß, dass dieser Platzhalter »sprünge« heißt. Im nächsten Schritt wirst du den Platzhalter, das heißt seinen Zahlenwert, auf der LED-Matrix anzeigen.

## 5.2.2 Platzhalter anzeigen

Zu Beginn dieses Buchs hast du gelernt, wie du deinen Namen mit dem Calliope mini anzeigen kannst. Dein Name ist ein Text, auch »Zeichenkette« genannt. Ein Platzhalter in dem PXT-Editor ist aber kein Text mit Buchstaben, sondern eine Zahl.

Probier zunächst aber aus, was passiert, wenn du den Programmierbefehl »zeige Zeichenfolge« aus der Befehlsgruppe »Grundlagen« benutzt und dort den Platzhalter »sprünge« hineinziehst. Du hast bestimmt gesehen, dass der »sprünge«-Block eine Ausbuchtung an der linken Seite hat. Er passt damit an jede Stelle, wo eine Zahl oder ein Text in einem Befehl eingetragen werden soll.



Abb. 5-5 Benutze den Platzhalter »sprünge« als Zeichenfolge.

Siehst du eine Anzeige im Simulator auf der linken Seite? Nein. Aber warum zeigt der Simulator dir nichts an? Das liegt daran, dass in der »Platzhalter«-Box noch nichts drin ist – und schon gar keine Zeichenfolge aus Buchstaben. Der Platzhalter »sprünge« soll die Anzahl von Sprüngen zählen, er muss also eine Zahl sein, kein Text. Wie groß ist die Zahl ganz am Anfang, wenn du das Programm startest und noch keine Sprünge gezählt hast? Richtig: 0. Das Programm weiß das aber noch nicht, und du musst ihm das beim Start mit einem bestimmten Befehl einprogrammieren.

Wähle dazu in der Befehlsgruppe »Grundlagen« den Befehl »beim Start« aus. Du findest ihn dort ganz unten und musst dafür vielleicht etwas nach unten scrol- len.

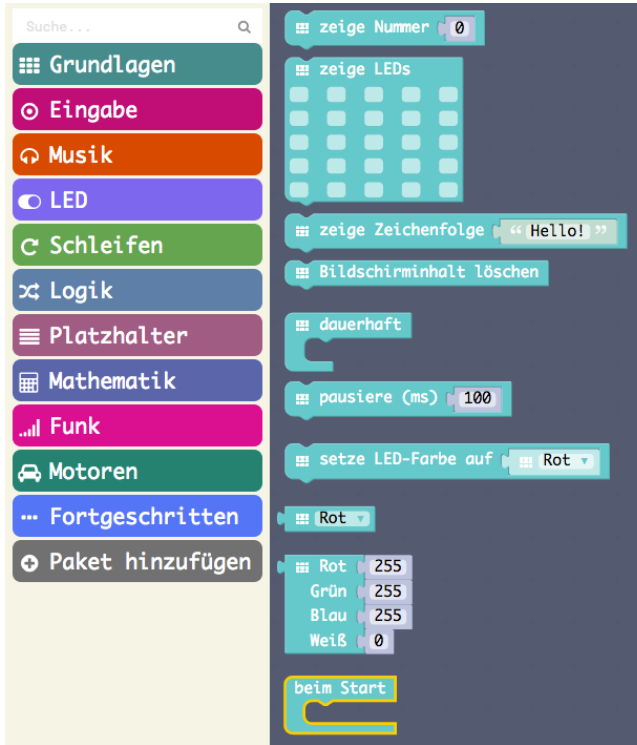


Abb. 5–6 »beim Start«-Befehl auswählen

In dieser »beim Start«-Klammer fügst du nun einen Programmierbefehl ein, der dem Platzhalter »sprünge« den Wert 0 zuweist, also die Zahl 0 in die Box mit dem Namen »sprünge« legt. Wähle dafür in der Befehlsgruppe »Platzhalter« den Befehle »ändere Platzhalter auf 0«.

## Hinweis

Bei der Auswahl des Befehls musst du genau hinschauen. Der Befehl »ändere Platzhalter **auf**« setzt den Platzhalter immer auf den gleichen dort eingetragenen Zahlenwert, während »ändere Platzhalter **um**« den Zahlenwert des Platzhalters um die dort eingetragene Zahl erhöht. Der Zahlenwert des Platzhalters kann ebenso verringert werden, wenn du ein Minus (-) vor der Zahl eingibst.

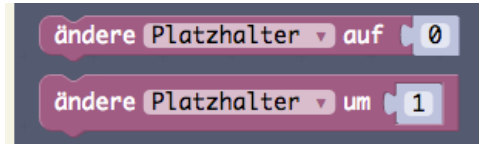


Abb. 5-7 Platzhalter können verändert werden.

Du kannst in dem »ändere«-Befehl mit einem Klick auf das kleine Dreieck neben dem Wort »Platzhalter« deinen eigenen Platzhalter »sprünge« auswählen.

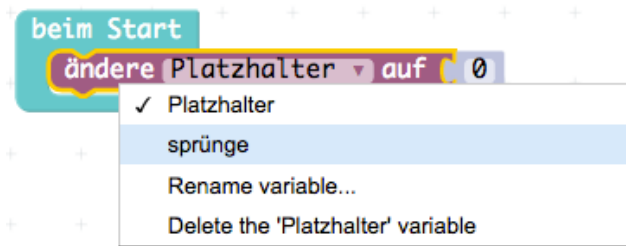


Abb. 5-8 Wähle deinen eigenen Platzhalter »sprünge« aus.

Nachdem du »sprünge« auf 0 geändert hast, siehst du ein blaues Dreieck auf der linken Seite des Programmierbefehls. Das ist eine Fehlermeldung. Der Editor zeigt dir dadurch, dass er deinen Befehl nicht versteht: »sprünge« soll eine Zahl sein, aber in der Dauerschleife hast du noch den Programmierbefehl »zeige Zeichenfolge« ausgewählt. Der Platzhalter »sprünge« ist nun aber vom Typ Zahl und passt dadurch nicht mehr in einen Befehl für Buchstaben.

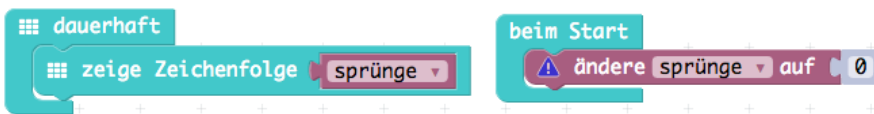


Abb. 5-9 Fehlermeldung, weil die Zahl 0 nicht vom Typ Zeichenfolge ist.

Du brauchst also einen Befehl, der dir statt einer Zeichenfolge eine Zahl (Nummer) anzeigen kann. Du findest ihn ganz oben in der Befehlsgruppe »Grundlagen«.



Abb. 5-10 »zeige Nummer«-Befehl auswählen

Wähle den Befehl aus und zieh ihn in die Dauerschleife. Als Nächstes ziehst du die »sprünge« aus dem »Zeichenfolge«-Befehl heraus und legst ihn auf die 0 im »Nummer«-Befehl. Du wirst sehen, dass die Fehlermeldung nun verschwunden ist. Jetzt passt der Zahleninhalt des Platzhalters »sprünge« zum anzeigenden »Nummer«-Befehl. Lösche den »Zeichenfolge«-Befehl, indem du ihn aus der Dauerschleife herausziehst.



Abb. 5–11 »zeige Nummer«-Befehl einsetzen und »sprünge« anzeigen.

Wenn du bis hierhin alles richtig gemacht hast, dann siehst du im Simulator auf der linken Seite eine 0. Das ist der Wert deines Platzhalters, wenn du das Programm startest und noch nichts gezählt hast. Als Nächstes wirst du deinen Platzhalter verändern.

### 5.2.3 Platzhalter mit Knopfdruck verändern

Du willst, dass dein Calliope mini bei jedem Tastendruck oder Klick mitzählt. Anders gesagt: Wenn du den Knopf »A« drückst, soll dein Platzhalter um 1 erhöht werden. Dafür gibt es in der Befehlsgruppe »Eingabe« den Programmierbefehl »wenn Knopf »A« gedrückt«.

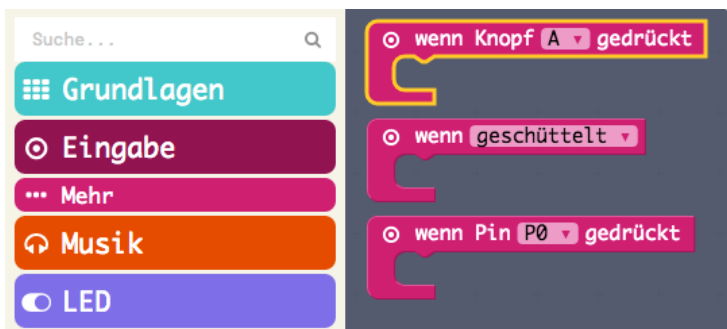


Abb. 5–12 »wenn Knopf »A« gedrückt«-Befehl auswählen

Schau genau hin: Dieser Befehl hat keine Delle am oberen Außenrand. Das bedeutet, dass er nicht in die Dauerschleife eingefügt werden kann, sondern selbst eine eigene neue Klammer darstellt.

Nun brauchst du noch einen Befehl aus der Befehlsgruppe »Platzhalter«. Wähle dort den »ändere Platzhalter um 1«-Befehl aus und ziehe ihn in deine neue

»wenn«-Klammer hinein. Durch einen Klick auf das kleine Dreieck neben dem Wort »Platzhalter« wählst du wieder deinen eigenen Platzhalter »sprünge« aus.

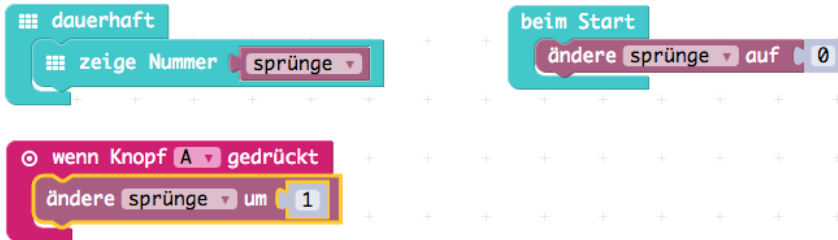


Abb. 5–13 Erhöhe die »sprünge« um 1 nach jedem Druck auf Knopf »A«.

Nun kannst du deinen Code im Simulator auf der linken Seite ausprobieren. Klicke dazu mit der Maus auf den Knopf »A«. Du siehst, wie sich die Zahl auf der Anzeige mit jedem Klick verändert und die neue Zahl angezeigt wird. Das klappt bei den ersten Zahlen 0–9 sehr schnell. Wenn die Zahl größer als 9 ist, reicht der Platz auf der LED-Matrix nicht mehr aus, und die Anzeige wird automatisch zu einer Laufanzeige umgewandelt, wie du es schon von der Anzeige von Texten kennst.

Jetzt ist alles fertig. Oder? Fällt dir ein Problem auf? Ach so, wenn du 21-mal auf den Knopf »A« gedrückt hast, sieht der angezeigte Text so aus: 212121212121 ... Je nachdem, wann du auf die LED-Matrix schaust, wirst du entweder 21 oder 12 lesen (oder 212 oder sogar 2121). Deswegen solltest du die Anzeige noch etwas verbessern.

## 5.2.4 Anzeige beim Zählen verbessern

Um zu sehen, wo die echte Anzahl der »sprünge« anfängt und aufhört, brauchst du irgendeine Form von Abstand oder Trennung zwischen jeder Anzeige. Es gibt verschiedene Möglichkeiten, wie du die einzelnen Zahlenanzeigen voneinander trennen kannst, zum Beispiel diese Blöcke aus der Befehlsgruppe »Grundlagen«:

- ▶ »zeige LEDs«
- ▶ »zeige Zeichenfolge«
- ▶ »Bildschirminhalt löschen«

Bei allen drei Möglichkeiten soll der Calliope mini nach jeder Anzeige der Anzahl kurz etwas anderes zeigen, bevor er dann wieder die Anzahl zeigt. Du brauchst also einen weiteren Befehl in der Dauerschleife. Wähle den großen Programmierbefehl »zeige LEDs« und ziehe ihn in die Klammer der Dauerschleife. Male nun ein kleines Muster, zum Beispiel einen langen Bindestrich »–«, indem du einzelne LEDs durch einen Klick rot markierst.

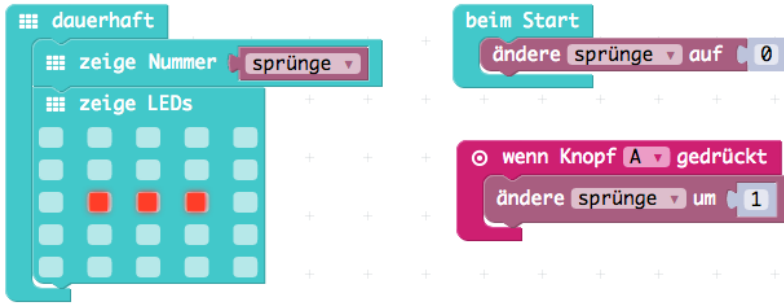


Abb. 5–14 Erstelle eine abwechselnde Anzeige mit einem trennenden Muster.

Probier aus, ob dir die Anzeige im Simulator gefällt. Wenn du noch nicht zufrieden bist, kannst du das Muster ändern oder eine andere der oben genannten Möglichkeiten ausprobieren, um deine Anzeigen voneinander zu trennen.

### Hinweis

Möchtest du schneller sehen, ob dir die Anzeige bei größeren Zahlen gefällt, ohne dass du vorher immer so oft auf den Knopf drücken musst? Du kannst zum Testen den Startwert (»beim Start«) von »sprünge« schon auf 21 oder eine andere Zahl einstellen. Vergiss nicht, diesen Wert wieder auf 0 zu ändern, bevor du das Programm herunterlädst.

## 5.2.5 Von vorne zählen

Wenn ihr euch beim Spielen abwechselt, dann wollt ihr auch nacheinander eure Sprünge oder Punkte zählen. Der Calliope mini soll dann nicht bei der ersten Anzahl weiterzählen, sondern von vorne beginnen mit 0 Sprüngen. Dazu programmierst du es so, dass beim Drücken von Knopf »B« der Platzhalter »sprünge« wieder auf 0 gesetzt wird.

Du brauchst einen neuen »wenn Knopf gedrückt«-Befehl, bei dem du diesmal den Knopf »B« auswählst. Du findest ihn wieder in der Befehlsgruppe »Eingabe«. Über einen Klick auf das kleine Dreieck neben dem »A« kannst du den entscheidenden Knopf auswählen.

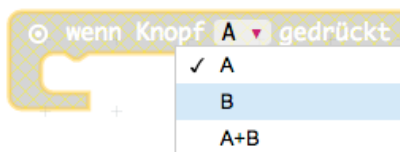


Abb. 5–15 Ändere den Knopf auf »B«

Wähle außerdem in der Befehlsgruppe »Platzhalter« einen neuen »ändere Platzhalter auf 0«-Befehl und füge ihn in die neue Klammer ein. Wähle über das kleine Dreieck erneut »sprünge« aus. Dein Programm reagiert nun sowohl auf den Knopf »A« (»ändere sprünge um 1«) als auch auf den Knopf »B« (»ändere sprünge auf 0«).

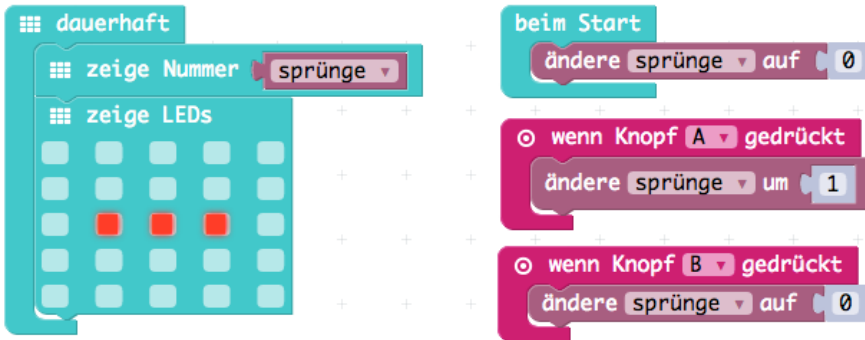


Abb. 5–16 So sieht das fertige Programm für den Klickzähler aus.

Probier dein Programm im Simulator aus. Wenn alles so funktioniert, wie du es möchtest, dann lade das Programm auf deinen Calliope mini herunter. Teste dein Programm und drücke auf die beiden Knöpfe.

## 5.2.6 Noch mehr zählen

Nun hast du einen einfachen Klickzähler programmiert und der Calliope mini reagiert auf jeden Knopfdruck von dir. Überleg dir, wie du dein Programm verändern musst, wenn du die Punkte für zwei Teams, also Team A und Team B, zählen möchtest? Wie viele Platzhalter braucht ihr? Wie erhöht ihr die Platzhalter? Und wie funktionieren die Anzeige und der Neustart?

Du kannst das Programm auch dafür verwenden, um eine Zahl mit jedem Klick kleiner zu machen, bis der Zahlenwert 0 erreicht ist. Welchen Wert muss dein Platzhalter dafür beim Start haben? Am Ende kannst du außerdem einen Ton abspielen. Wenn ihr im Sportunterricht zum Beispiel 50 Hampelmänner machen sollt, wisst ihr dadurch immer, wie viele noch übrig sind, und hört am Schluss eine Melodie zur Belohnung.

Bestimmt fallen dir auch noch viel mehr Dinge ein, die du zählen kannst.

## 5.3 Die digitale Sanduhr

Du möchtest ein Spiel spielen, bei dem eine Runde eine ganz bestimmte Zeit hat? Oder willst du eine bestimmte Zeit lang auf einem Bein balancieren? Dann brauchst du eine Uhr, die die Zeit wie bei einem Countdown herunter zählt und am besten auch noch anzeigt, wie viel Zeit noch übrig ist. Genau das macht eine Sanduhr, und du wirst in diesem Projekt eine eigene digitale Sanduhr programmieren.

In diesem Kapitel lernst du, wie du ein Programm für den Calliope mini schreiben kannst, bei dem die einzelnen LEDs wie (ziemlich große) Sandkörner nacheinander verschwinden.

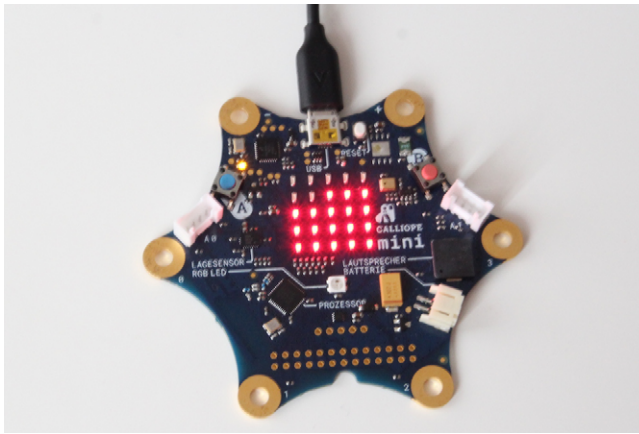
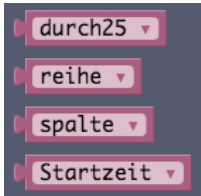


Abb. 5–17 Die digitale Sanduhr läuft ab.

### 5.3.1 Platzhalter erstellen

Das Programm deiner digitalen Sanduhr läuft eine bestimmte Zeit. Dabei möchtest du, dass diese ganze Zeit so aufgeteilt wird, dass die 25 LEDs nacheinander ausgeschaltet werden. Du musst dem Programm dafür sagen, in welcher Reihe und in welcher Spalte du den Punkt ausschalten willst. Du kennst schon Spalten und Reihen in Tabellen, so ähnlich funktioniert das hierbei auch, wie du später noch genau sehen wirst. Für dieses Programm brauchst du insgesamt vier Platzhalter:

- ▶ »Startzeit«: Wie lange soll die Sanduhr dauern?
- ▶ »durch25«: Es gibt 25 LEDs, also musst du die ganze Dauer durch 25 rechnen – so viel Zeit soll bis zum Ausschalten der jeweils nächsten LED vergehen.
- ▶ »spalte«: In welcher Spalte soll die LED ausgeschaltet werden?
- ▶ »reihe«: In welcher Reihe soll die LED ausgeschaltet werden?



Öffne die Befehlsgruppe »Platzhalter« und klicke dort mit der Maus auf »Neuen Platzhalter anlegen«. Benenne den neuen Platzhalter mit »Startzeit« und wiederhole diesen Schritt für die anderen Platzhalter, bis du alle vier Platzhalter hast.

Abb. 5–18 Diese vier Platzhalter brauchst du für die Sanduhr.

### 5.3.2 Sanduhr startbereit machen

Schon bevor das Programm startet, muss es wissen, wie lange es insgesamt laufen soll. Bei einer echten Sanduhr hängt die Dauer vor allem von der Menge der Sandkörner ab. Bei deinem Calliope mini kannst du aber nichts an der Anzahl der LEDs ändern, also musst du es etwas anders machen. Du erreichst es dadurch, dass du den Wert der »Startzeit« ganz am Anfang auf die Zeit setzt, die das Programm dauern soll. Dafür brauchst du die Programmierbefehle »beim Start« und »ändere Platzhalter auf« aus der Befehlsgruppe »Platzhalter«. Ändere den Platzhalter durch das kleine Dreieck auf »Startzeit«. Die Zahl dort muss du in Millisekunden (ms) angeben. Zur Erinnerung: 1 Sekunde hat 1000 Millisekunden. Wenn du also möchtest, dass deine Sanduhr 25 Sekunden dauert, musst du den Befehl »ändere Startzeit auf 25000« ( $25 \times 1000$ ) verwenden. Dann solltest du auch berechnen, wie viel Zeit vergeht, bis die nächste LED ausgeschaltet wird. Dafür brauchst du den »ändere auf«-Befehl und einen » $0 \div 0$ «-Block aus der Gruppe »Mathematik«. Ersetze die erste 0 durch einen »Startzeit«-Block und die zweite 0 durch 25.

Und was glaubst du, sollte zu sehen sein, wenn du die Calliope-Sanduhr anmachst? Nun, da am Anfang alle Sandkörner noch drin sind, solltest du auch alle LEDs rot anzeigen. Dafür brauchst du den »zeige LED«-Befehl aus der Befehlsgruppe »Grundlagen«. Ziehe ihn in deine »beim Start«-Klammer und klicke alle LEDs an, damit sie alle rot leuchten.

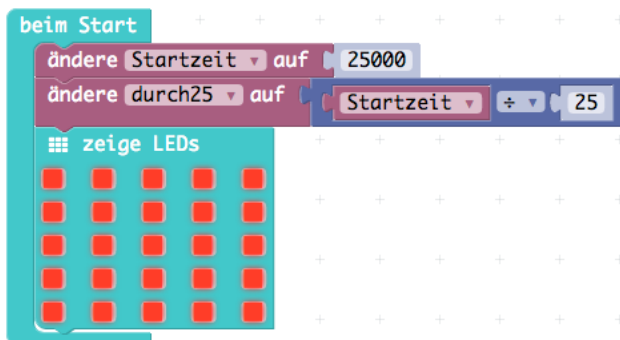


Abb. 5–19 Die digitale Sanduhr ist startklar.

### 5.3.3 Eine einzelne LED ausschalten

Für deine digitale Sanduhr musst du Befehle benutzen, mit denen es möglich ist, jede der 25 roten LEDs einzeln zu steuern, das heißt ein- und auszuschalten. Dafür gibt es die Befehlsgruppe »LED«. Wenn alle 25 LEDs beim Start rot leuchten, brauchst du einen Befehl, der die LEDs nacheinander wieder ausschaltet. Der PXT-Editor bezeichnet die LEDs als Pixel, und der Befehl »schalte Pixel x 0 y 0« meint: Schalte die LED an der Stelle x 0 und y 0 ab.

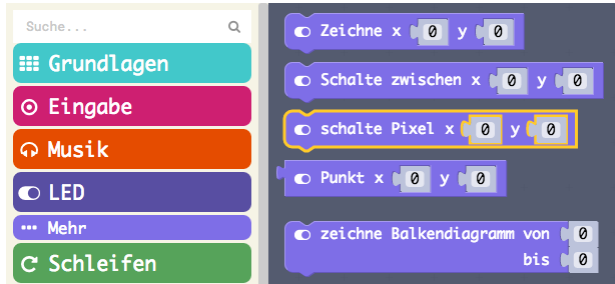


Abb. 5–20 Befehl zum (Ab)schalten einzelner Pixel

Bei den Zahlen hinter x und y handelt es sich um die Angabe der Position der LED. Die Zahl hinter dem x steht für die Spalte, die Zahl hinter dem y steht für die Zeile oder Reihe, in der sich die LED, also das Pixel befindet. Wenn da »x 0« und »y 0« steht, dann ist damit die LED in der oberen linken Ecke gemeint, also die allererste LED, wenn du die LEDs von oben nach unten und von links nach rechts »lesen« würdest. In der Informatik beginnen aber viele Aufzählungen und Listen mit der Zahl 0, also ist mit »x 0« die erste Spalte gemeint und mit »y 4« die unterste, fünfte Reihe. Die LED an der Stelle »x 1 y 2« zeigt dir die folgende Abbildung, damit du es noch einmal besser verstehst.

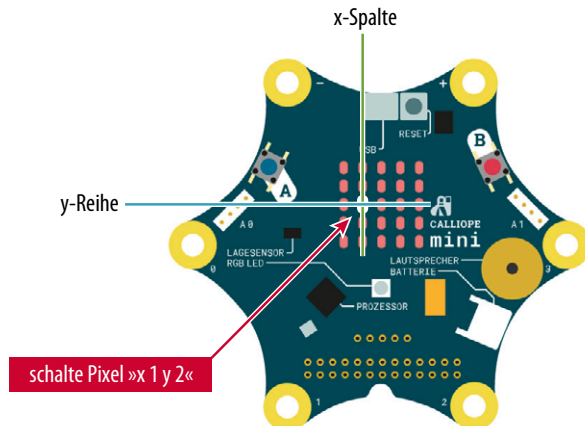


Abb. 5–21 Positionsangaben für die LEDs

Um diesen »schalte Pixel«-Befehl zu aktivieren, musst du ihn in eine passende Klammer einfügen, damit der Befehl ausgeführt wird, wenn du auf die Taste A drückst. Du findest diesen »wenn Knopf »A« gedrückt«-Befehl wie immer in der Befehlsgruppe »Eingabe«. Wähle ihn aus und verbinde die beiden Befehle miteinander. Du kannst nun eine Zahl von 0 bis 4 hinter x und y selbst eintragen und im Simulator ausprobieren, an welcher Stelle diese LED ist, wenn du Knopf »A« klickst.

Nun erinnere dich an die Platzhalter, die du am Anfang dieses Projekts erstellt hast. Einer der Platzhalter heißt »spalte« und ein anderer heißt »reihe«. Anstatt die Zahlen für x und y selbst einzutragen, kannst du auch die »Platzhalter«-Blöcke benutzen. Was glaubst du, welche LED ausgeschaltet wird, wenn du nun im Simulator Knopf »A« klickst? Warum ist es genau diese LED?



Abb. 5-22 Schalte diejenige LED ab, die in dieser Spalte und dieser Reihe ist.

Da du diese beiden Platzhalter noch nicht verändert hast, haben sie den Wert 0, weshalb die allererste LED oben links (Position bei »x 0 y 0«) ausgeschaltet wird.

Du kannst nun die Position der einzelnen LED nur noch verändern, indem du den Wert der Spalte und der Reihe änderst. In diese beiden Platzhalterboxen können immer wieder verschiedene Zahlen eingetragen werden.

## Hinweis

Probiere aus, was passiert, wenn du einen »ändere Platzhalter um 1«-Befehl vor den »schalte Pixel«-Befehl ziehst. Vergiss dabei nicht, durch Klick auf das kleine Dreieck die richtige Variable (»reihe« oder »spalte«) auszuwählen. Drücke im Simulator mehrmals auf Knopf »A«.

## 5.3.4 Ganze Reihe mit Knopfdruck ausschalten

Nun möchtest du, dass die Sandkörner-LEDs nacheinander ausgeschaltet werden – und zwar automatisch nacheinander, sobald du nur ein einziges Mal auf Knopf »A« drückst. Dafür musst du etwas programmieren, bei dem du zuerst mit einer Reihe anfängst, dort nacheinander alle LEDs spaltenweise durchgehst, und dann zur nächsten Reihe gehst und so weiter – so, als würdest du einen Text Wort für Wort lesen. Du wirst gleich sehen, dass du bei solch einer Wiederholung nicht viel rechnen musst, wenn du es einmal verstanden hast. Natürlich kannst du die Reihenfolge ändern, in der die einzelnen LEDs verschwinden, aber dazu erst später mehr.

Es gibt verschiedene Möglichkeiten, wie du diese Wiederholung programmieren kannst. Computerprogramme verwenden dafür sogenannte »Schleifen«. Im PXT-Editor findest du eine Befehlsgruppe, die auch »Schleifen« heißt.



Abb. 5-23 Befehlsgruppe »Schleifen«

Die einfachste Möglichkeit, einen oder mehrere Befehle mehrmals nacheinander zu wiederholen, ist der oberste Programmierbefehl »4-mal wiederholen, mache«. Zieh diesen Befehl in die »wenn Knopf »A« gedrückt«-Klammer und ändere die Zahl auf 5, weil es 5 LEDs in jeder Reihe gibt. Verschiebe den »schalte Pixel«-Befehl aus dem letzten Schritt in die neue grüne Klammer und füge einen »ändere spalte um 1«-Befehl hinzu, wenn du dies im letzten Schritt noch nicht ausprobiert hast.

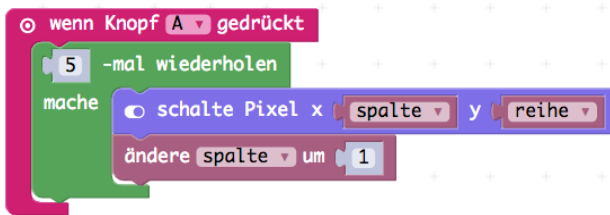


Abb. 5-24 Wiederholung der Befehle für eine ganze Reihe

Was genau bedeutet diese Wiederholung nun? Stell dir vor, du könntest auf einem sehr großen Calliope »maxi« mit deinem Finger jede LED durch Antippen einfach ausschalten. Du beginnst in der 1. Spalte und in der 1. Reihe und tippst die 1. LED an. Sie geht aus. Dann gehst du mit deinem Finger zur nächsten 2. Spalte, also einmal nach rechts. Diese LED tippst du wieder an. Die Reihe wird dabei nicht verändert, du bleibst nur in der 1. Reihe. Du machst weiter: 3. Spalte – antippen – 4. Spalte – antippen – 5. Spalte – antippen – 6. Spalte. Hm, da ist zwar keine LED mehr, aber das macht auch nichts, denn du musst gar nichts mehr antippen, weil die Wiederholungsklammer zu Ende ist.

Probier das Programm nun im Simulator aus. Was passiert? Wenn du alles richtig gemacht hast, dann müsste – nachdem du Knopf »A« gedrückt hast – die oberste Reihe der LEDs ausgeschaltet, also weiß sein. Doch wie lange – oder besser gesagt kurz – hat das gedauert? Das ging schon fast zu schnell, um es überhaupt richtig sehen zu können, oder? Aber du willst ja eigentlich ein Programm, das die LEDs deutlich sichtbar nacheinander ausschaltet. Dafür musst du eine kleine Pause einbauen. Du findest den »pausiere«-Befehl in der Befehlsgruppe »Grundlagen«, und du solltest ihn gleich oben in deiner »Wiederhole«-Klammer einfügen. In diesen Befehl musst du wieder eine Zahl eintragen, die anzeigt, wie viele Millisekunden (ms) die Pause dauern soll. Jetzt brauchst du den »durch25«-Platzhalter. Füge ihn aus der Befehlsgruppe »Platzhalter« als Dauer für den »pausiere«-Befehl ein.



Abb. 5–25 Füge eine Pause hinzu.

Wenn du diesen Code mit der Pause nun im Simulator ausprobierst, siehst du, dass es nun viel länger dauert, bis die letzte LED in der obersten Reihe ausgeschaltet wird. Wie kannst du nun einstellen, dass die LEDs schneller oder langsamer nacheinander ausgeschaltet werden? Du kannst die Dauer der Pause ändern, wenn du den Wert des »Startzeit«-Platzhalters in der »beim Start«-Klammer veränderst. Je größer die Startzeit, desto größer ist auch das Ergebnis von Startzeit geteilt durch 25.

### 5.3.5 Jede Reihe nacheinander ausschalten

Für eine richtige digitale Sanduhr reicht es aber noch nicht, wenn nur die oberste Reihe ausgeschaltet wird. Du möchtest danach in der 2. Reihe weitermachen und wieder von links alle LEDs ausschalten, dann mit der 3. Reihe und so weiter. Du willst die Befehle aus dem letzten Schritt für jede Reihe wiederholen. Kannst du dir schon vorstellen, was du dafür brauchst? Richtig: eine neue »Wiederholen«-Klammer.

Diese neue Klammer musst du so einfügen, dass die alte Klammer von der neuen Klammer eingeschlossen wird. Dafür musst du die Blöcke im Editor etwas herumschieben, aber dann sollte es so aussehen wie auf dem nächsten Bild. Du hast nun eine äußere Klammer (die neue) und eine innere Klammer (die alte).

## Hinweis

Programmierer nennen solche Codes mit mehreren Schleifen-Klammern ineinander auch »verschachtelte Schleifen«. Die Befehle, die in der inneren Klammer stehen, werden so oft ausgeführt, wie die Anzahl der Wiederholungen miteinander multipliziert wird. In diesem Projekt werden die drei Befehle (»pausiere«, »schalte«, »ändere spalte«) 5 x 5 Mal, also 25 Mal ausgeführt. Sehr gut, immerhin sind es ja auch 25 LEDs.



Abb. 5–26 Innere und äußere »Wiederholen«-Klammern

Ist dir der »ändere reihe um 1«-Befehl unten in der Klammer aufgefallen? Stell dir wieder einen großen Calliope maxi vor: Du brauchst diesen Programmierbefehl, damit du mit deinem Finger auf die nächste Reihe zeigen kannst. Und zwar darfst du erst dann in die nächste Reihe gehen, wenn du vorher fünfmal eine LED ange-tippt und eine Spalte weiter nach rechts gegangen bist. Dann wurde die innere Schleife 5-mal ausgeführt und das Programm macht mit den Befehlen der äußeren Schleife weiter.

## Hinweis

Achte beim Einfügen dieses Befehls darauf, dass du ihn genau zwischen dem Ende der inneren Klammer und vor dem Ende der äußeren Klammer einfügst.

Doch was passiert, wenn du das Programm im Simulator mit Knopf »A« startest? Werden mehr LEDs ausgeschaltet als die in der ersten Reihe? Hm, wahrscheinlich nicht. Woran liegt das? Wenn du mit deinem Finger am Ende der 1. Reihe angekommen bist, darfst du mit deinem Finger in die 2. Reihe gehen. Und in welche

Spalte? Richtig: in die 6. Dann in die 7. und in die 8. Spalte und so weiter. Doch auf der LED-Matrix gibt es nur 5 Spalten, also was solltest du eigentlich machen? Du solltest mit deinem Finger in der 2. Reihe wieder die 1. LED ganz links antippen. Das heißt, dass du zurück zur 1. Spalte musst. Welchen Befehl brauchst du dafür und an welcher Stelle musst du den Befehl einfügen? Die Auflösung findest du, wenn du dir das nächste Bild genauer anschaust, doch überlege zuerst selbst, wie du es zurück zur 1. Spalte schaffen könntest.

## Hinweis

Vergiss nicht, dass Computerprogramme oft bei 0 anfangen zu zählen!

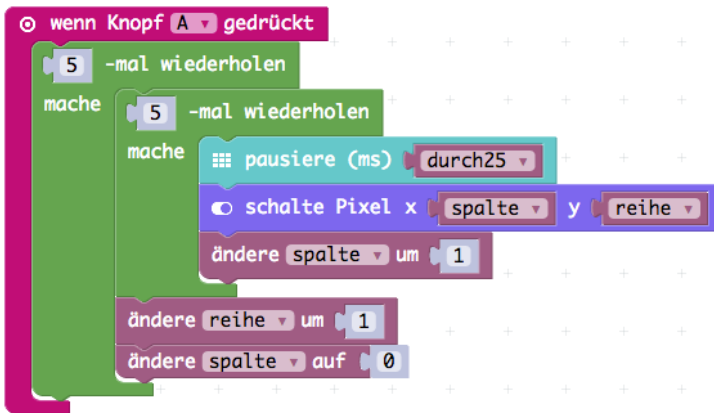


Abb. 5-27 Zurückspringen zur ersten Spalte

## Hinweis

Dein Programm funktioniert auch, wenn du den »spalte«-Platzhalter vor dem Beginn der inneren »Wiederholen«-Klammer auf 0 änderst.

## 5.3.6 Musik abspielen

Wie du schon weißt, kann der Calliope mini auch Musik abspielen. Wenn du möchtest, dass bei deiner Sanduhr nach der abgelaufenen Zeit nicht nur alle LEDs aus sind (keine Sandkörner mehr), sondern auch noch deutlich zu hören ist, dass die Zeit abgelaufen ist, dann kannst du ein paar »Noten«-Blöcke einfügen. Du

findest sie in der Befehlsgruppe »Musik« und kannst zum Beispiel die Tonfolge »G E C« abspielen, nachdem die äußere »Wiederholen«-Klammer beendet ist.

```
ändere reihe um 1
ändere spalte auf 0
spiele Note G für 1 Takt
spiele Note E für 1 Takt
spiele Note C für 1 Takt
```

Abb. 5–28 Melodie am Ende des Programms

### 5.3.7 Sanduhr starten

Deine digitale Sanduhr ist jetzt fertig. Das nächste Bild zeigt dir den gesamten Code deines Programms, wenn du alle Schritte bisher richtig gemacht hast. Beim Starten deiner Sanduhr im Simulator verschwinden die LEDs nun alle nacheinander von links nach rechts und von oben nach unten.

```
beim Start
ändere Startzeit auf 25000
ändere durch25 auf Startzeit / 25
zeige LEDs

wenn Knopf A gedrückt
5 -mal wiederholen
mache
5 -mal wiederholen
mache
  pausiere (ms) durch25
  schalte Pixel x spalte y reihe
  ändere spalte um 1
  ändere reihe um 1
  ändere spalte auf 0
spiele Note G für 1 Takt
spiele Note E für 1 Takt
spiele Note C für 1 Takt
```

Abb. 5–29 So sieht dein fertiges Programm aus.

Dieses Programm kannst du nun speichern und auf deinen Calliope mini herunterladen und dort benutzen. Wenn du möchtest, kannst du deine Sanduhr im nächsten Schritt noch etwas verändern. Lies weiter, wenn du dich schon gefragt

hast, ob und wie es möglich ist, die Reihenfolge zu ändern, in der die einzelnen LEDs ausgeschaltet werden.

### 5.3.8 Reihenfolge verändern

Die Reihenfolge hängt davon ab, ob du in der inneren »Wiederholen«-Klammer die »reihe« oder »spalte« erhöhst und was du in der äußeren Klammer erhöhst und auf 0 änderst. Probier es aus und wähle in allen drei »ändere«-Blöcken das Gegenteil von dem aus, was gerade verändert wird. Also klicke auf das kleine Dreieck neben »spalte« und wähle »reihe« aus und umgekehrt. Klicke dann auf Knopf »A« im Simulator.

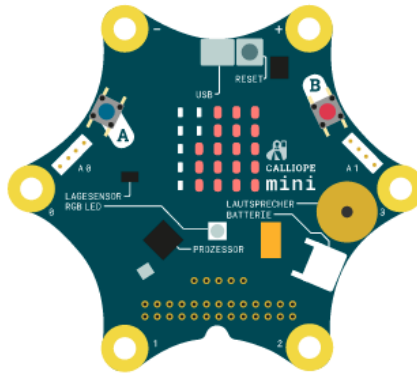


Abb. 5–30 Veränderte Reihenfolge beim Ausschalten der LEDs

Zuerst verschwindet nun die ganze linke Spalte, dann die zweite Spalte – und zwar immer von oben nach unten jede LED nacheinander. Du kannst dir natürlich die Reihenfolge für deine eigene Sanduhr selbst aussuchen.

Etwas schwieriger ist es aber, wenn du möchtest, dass die Sandkörner-LEDs rechts unten als erste ausgehen. Wenn du wieder an deinen Finger und das große Calliope maxi denkst, wirst du ja erst in der letzten Spalte und in der letzten Reihe anfangen, um dort die LED ganz rechts unten zuerst anzutippen. Weil du inzwischen weißt, dass Programmierer mit dem Zählen bei 0 anfangen und dass die ganz linke LED-Spalte bei  $x = 0$  liegt, weißt du natürlich auch, dass die ganz rechte LED-Spalte bei  $x = 4$  liegt. Das Gleiche gilt für die LED-Reihe und  $y$ : Die unterste Reihe liegt bei  $y = 4$ . Da deine Platzhalter »spalte« und »reihe« bei 0 anfangen, kannst du sie von der 4 subtrahieren (-) und erhältst beim ersten Durchlauf der »Wiederholen«-Klammer ( $4 - 0$ ) als Ergebnis 0. Dafür brauchst du den » $0 - 0$ «-Befehl aus der Befehlsgruppe »Mathematik«, den du so verwendest, wie er auf der nächsten Abbildung dargestellt ist.

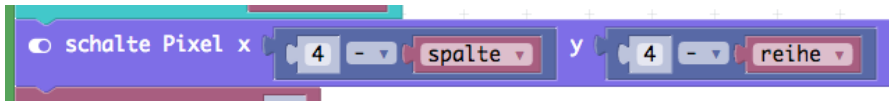


Abb. 5–31 So wird als erstes die LED unten rechts abgeschaltet.

Probier es im Simulator aus. Gefällt dir diese Reihenfolge besser? Oder möchtest du wieder erst die untere Reihe komplett ausschalten? Na, zum Glück weißt du auch noch, wie das geht!

Du kannst das Programm nun auf deinen Calliope mini herunterladen und dort starten. Wenn die Sanduhr abgelaufen ist, kannst du mit dem »Reset«-Knopf immer wieder zum Anfang zurück und die Sanduhr mit Knopf »A« neu starten. Aber wenn du möchtest, kannst du deinen Code auch noch so verändern, dass die Sanduhr bei jedem Knopfdruck auf B wieder von vorne beginnt oder zumindest startklar ist, bis sie beim Drücken auf Knopf »A« wieder losgeht. Was musst du dafür noch ergänzen in deinem Code? Denk daran, dass du »reihe« und »spalte« dann auch erst auf 0 setzen musst, wenn du auf den Knopf »B« drückst.

## 5.4 Das Metronom

Auf den Takt kommt es an! Das ist das Motto heute, denn du baust mit dem Calliope ein Metronom. Das kennst du vielleicht aus dem Musikunterricht. Falls nicht, macht es auch nichts aus, wir erklären es dir kurz. Ein Metronom hilft Musikern, den richtigen Takt zu halten. Dazu gibt es in einem gleichmäßigen Tempo einen Ton von sich. Dieses Tempo wird vom Musiker vorher eingestellt und variiert je nach Musikstück. Du kennst sicherlich viele Lieder mit unterschiedlichem Tempo, manche sind langsam, andere sehr schnell.



Abb. 5–32 Der Calliope mini kommt als Metronom zum Einsatz.

Genau so ein Gerät kannst du mit dem Calliope mini nachbauen. Die Idee ist, dass du über zweimaliges Drücken von Knopf »A« den Takt vorgeben kannst. Dazu wird der Abstand zwischen diesen beiden Klicks gemessen und dann in einem Platzhalter abgespeichert. Folgenden Ablauf soll das Programm haben:

1. Knopf »A« wird gedrückt, du speicherst die aktuelle Laufzeit in einem Platzhalter ab.
2. Knopf »A« wird ein zweites Mal gedrückt. Du ziehst von der aktuellen Laufzeit die vorher im Platzhalter gespeicherte Laufzeit ab und speicherst das Ergebnis in einem neuen Platzhalter.
3. Der Calliope mini soll nun in diesem Tempo einen Ton abspielen.

## Was ist Laufzeit?

Das ist ein Begriff, den du in der Informatik noch sehr oft hören wirst. Die Laufzeit ist eine Zahl, die ununterbrochen mit jeder Millisekunde um 1 hochgezählt wird. Wenn du den Calliope mini einschaltetest, ist die Laufzeit ganz kurz auf 0 und eine Sekunde später auf 1000, also 1 Sekunde in Millisekunden. Das Tolle an der Laufzeit ist, dass du damit rechnen kannst. Wenn du also den Abstand zwischen zwei Ereignissen messen willst, kannst du dazu die Laufzeit verwenden. Das kannst du auch für dein Metronom verwenden.

Beim PXT-Editor findest du die Laufzeit unter »Eingabe«, »Mehr«.

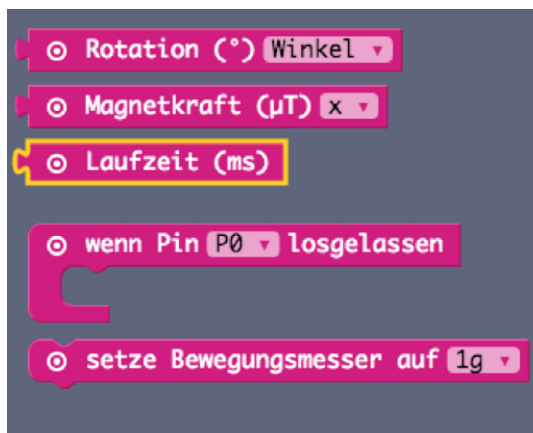


Abb. 5-33 Laufzeit im PXT-Editor

Los geht es mit dem Programmieren! Du beginnst damit, den Knopf »A« zu programmieren. Er soll beim ersten Drücken die aktuelle Laufzeit abspeichern, dazu verwendest du einen neuen Platzhalter »Start«.



Abb. 5–34 Der Platzhalter »Start« wird eingefügt

Diesen Block kannst du nun schon einmal austesten. Lass dir dazu den Platzhalter »Start« auf den LEDs ausgeben. Dafür kannst du den Block »zeige Nummer« verwenden. Dieser Block soll dauerhaft ausgeführt werden, da du immer den aktuellen Wert des Platzhalters ausgegeben haben möchtest.

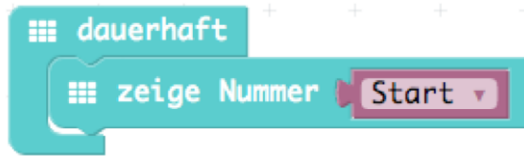


Abb. 5–35 Der Platzhalter »Start« soll auf den LEDs dargestellt werden.

Wenn du das Calliope-Board nun startest, sollte auf den LEDs erst einmal die Zahl 0 zu sehen sein. Das liegt daran, dass der Platzhalter »Start« noch nicht gesetzt wurde. Wenn du nun den Knopf »A« drückst, sollte auf den LEDs die aktuelle Laufzeit angezeigt werden, da du sie immer im Platzhalter »Start« abspeicherst. Teste es einmal aus: Bei jedem Druck von Knopf »A« sollte sich die Zahl erhöhen, da die Laufzeit nie rückwärts laufen kann.

Nun soll irgendwie der zweite Druck auf Knopf »A« erkannt werden. Aus dem vorherigen Experiment weißt du, dass der Platzhalter »Start« den Wert 0 hat, wenn der Knopf noch nicht gedrückt wurde. Das kannst du verwenden, um zu erkennen, ob Knopf »A« schon gedrückt wurde oder eben nicht. Verwende dazu eine Verzweigung, die du schon zu Beginn von Kapitel 3 kennengelernt hast. In der Verzweigung prüfst du, ob der Platzhalter »Start« nicht 0 ist. Sobald der Knopf »A« zum ersten Mal gedrückt wird, speicherst du im Platzhalter »Start« die Laufzeit. Das ist reichlich kompliziert, daher solltest du erst einmal testen, ob dies so funktioniert. Versuche beim zweiten Druck auf den Knopf »A«, die LED rot leuchten zu lassen. Dann weißt du, dass das zweite Drücken richtig erkannt wurde.

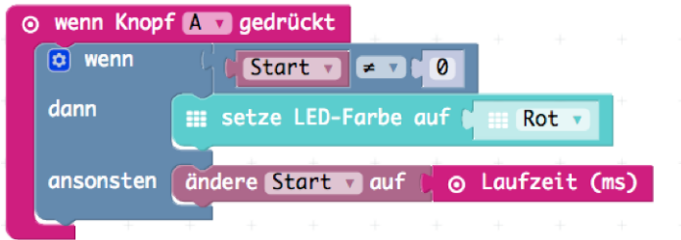


Abb. 5–36 Eine Verzweigung kommt hinzu

Kopiere das Programm auf den Calliope mini und teste es. Beim ersten Druck auf den Knopf »A« sollte noch nichts Sichtbares passieren, da du nur die Laufzeit in »Start« abspeicherst. Beim zweiten Druck allerdings sollte die LED anfangen, rot zu leuchten.

Nun weißt du, dass die Erkennung des zweiten Knopfdrucks funktioniert, und kannst das Programm weiterentwickeln. Der nächste Schritt besteht darin, den Abstand zwischen dem ersten und zweiten Druck zu erkennen. Dazu kannst du – wie vorher schon erklärt – die aktuelle und die in »Start« abgespeicherte Laufzeit verwenden. Die Minus-Rechnung dieser beiden Werte ergibt dabei den Abstand zwischen ihnen in Millisekunden. Den Abstand solltest du in einem neuen Platzhalter »Abstand« abspeichern, sodass du ihn später verwenden kannst.

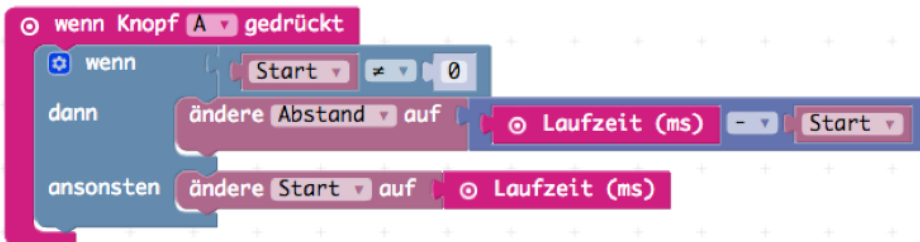


Abb. 5–37 Der Abstand wird im Platzhalter »Abstand« gespeichert

Jetzt kannst du prüfen, ob im Platzhalter »Abstand« auch tatsächlich der Abstand zwischen zweimaligem Drücken auf Knopf »A« gespeichert wird. Verwende dazu wieder den Block »zeige Nummer«, nur wird dieses Mal nicht der Wert von »Start«, sondern von »Abstand« angezeigt.

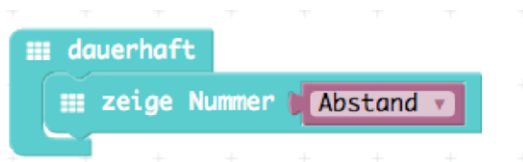


Abb. 5–38 Der Platzhalter »Abstand« soll getestet werden

Kopiere das Programm auf den Calliope mini. Drücke zweimal hintereinander Knopf »A«, und es sollte eine Zahl auf den LEDs angezeigt werden. Diese Zahl ist der Abstand zwischen dem ersten und dem zweiten Druck auf Knopf »A« in Millisekunden. Probiere es ein paar Mal in verschiedenen Geschwindigkeiten aus. Je nach Geschwindigkeit sollte die Zahl entsprechend niedrig oder hoch sein. Um die Zeit zu überprüfen, kannst du dir als Hilfe auch eine Uhr mit Sekundenzeiger nehmen. Drücke Knopf »A«, wenn der Sekundenzeiger auf 0 ist, und drücke ihn ein zweites Mal, wenn er auf 30 Sekunden ist. Auf den LEDs sollte jetzt ungefähr 30000 angezeigt werden, das entspricht den 30 Sekunden, die du gewartet hast.

Nun soll mit Hilfe der in »Abstand« gespeicherten Zeit ein Ton abgespielt werden. Du beginnst am besten mit dem Abspielen eines Tons und lässt das Programm dann eine Sekunde warten. Diese Wartezeit wird später durch die vorher im Platzhalter »Abstand« gespeicherte Zeit ersetzen.

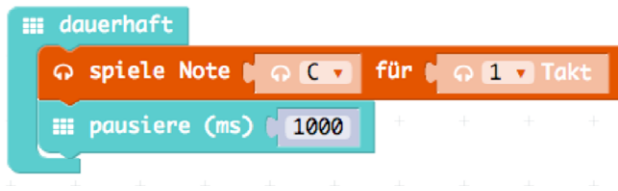


Abb. 5–39 Ein erster Versuch

Sobald das Programm auf dem Calliope mini läuft, sollte mit dem Abstand von einer Sekunde die Note C für einen Takt abgespielt werden.

Das Programm benötigt nun noch ein paar Verbesserungen, bis es das macht, was du mit dem Metronom vorhast. Als Erstes soll es, statt 1 Sekunde zu warten, den von dir vorher ermittelten Abstand warten. Außerdem ist der eine Takt noch zu lang. Ändere dies und gib dem Programm eine zweite Chance.



Abb. 5–40 Der nächste Versuch

Jetzt ist etwas ganz merkwürdig, denn nach dem Einschalten des Calliope mini mit diesem Programm spielt es ohne Unterbrechung sehr schnell hintereinander den Ton ab. Woran liegt das? Beim Start deines Programms ist der Platzhalter »Abstand« noch nicht definiert. Für den »pausiere«-Block bedeutet dies, einfach gar keine Pause einzulegen. So wird der Ton immer wieder abgespielt. Du musst also auch hier noch mit Hilfe einer Verzweigung prüfen, ob »Abstand« schon gesetzt ist. Nur wenn das der Fall ist, soll der Ton abgespielt werden.

Erweitere das Programm so, dass der Ton nur abgespielt wird, wenn »Abstand« nicht 0 ist.

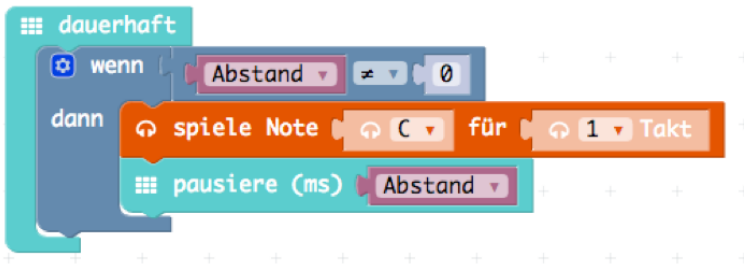
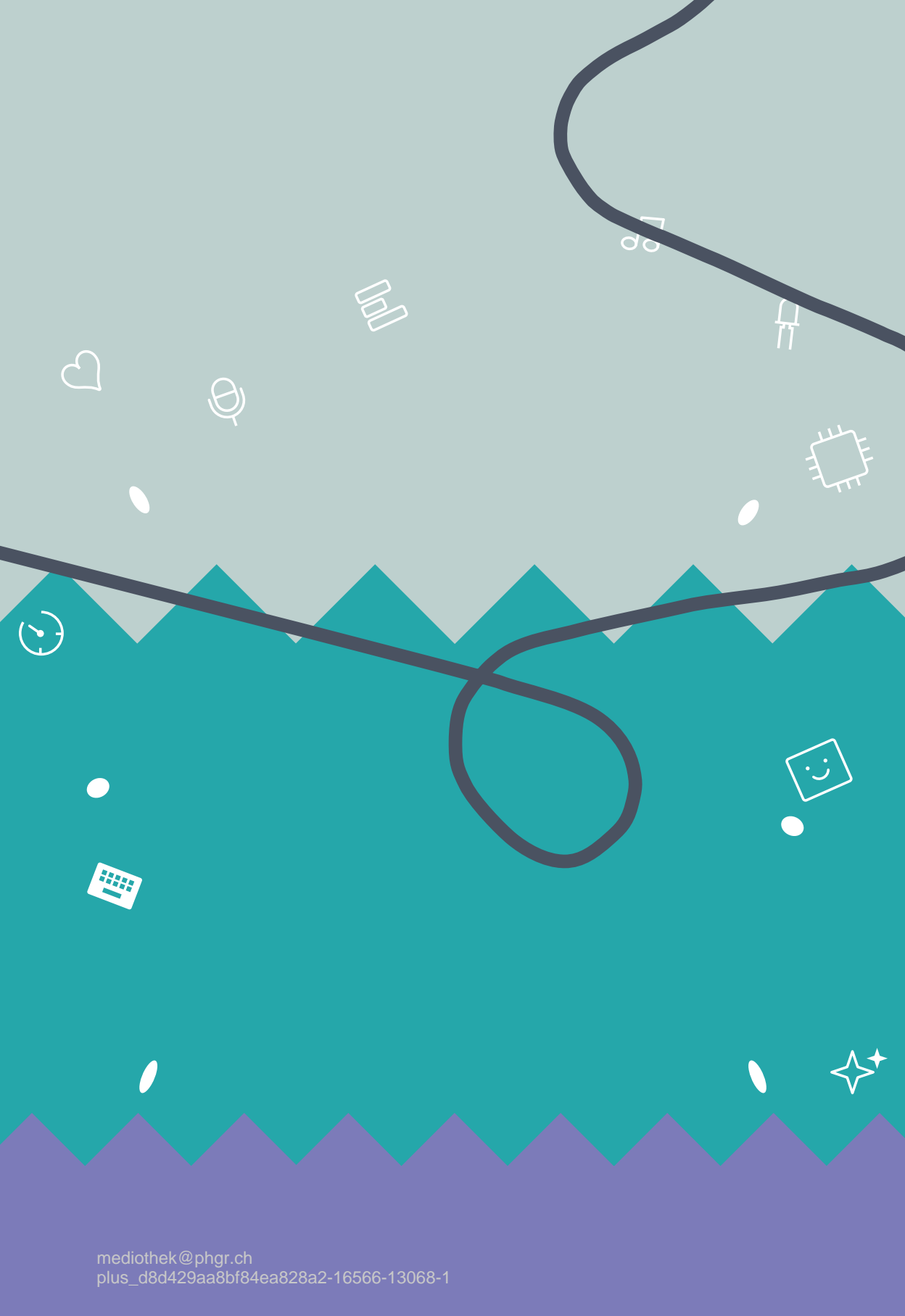


Abb. 5–41 Nun wird der Ton nur abgespielt, wenn »Abstand« gesetzt ist

Durch die Überprüfung, ob »Abstand« gesetzt ist, sollte erst einmal gar kein Ton aus dem Calliope mini kommen. Drücke jetzt zweimal hintereinander Knopf »A«, und das Calliope-Board sollte in diesem Takt den Ton abspielen.

Am besten testest du nun dein Programm mit ein wenig Musik aus. Versuche dazu, den Takt der Musik zu finden und dann Knopf »A« passend zu drücken. Wenn alles gut geht, sollte dein Calliope mini mit dem Takt den Ton abspielen.



# 6

## Würfeln ohne Würfel

In diesem Kapitel geht es um Zufall und darum, wie du ihn deinem Calliope mini bebringst.

Die einführende Worte zu diesem Kapitel wurden von Natalia Prost geschrieben. »Würfeln mit Schwung« wurde von Mario Lukas erstellt. »Symbolschnapper« stammt von Natalia Prost, »Das Reaktionsspiel« von Nadine Bergner und »Farben-Bingo« von Lina Wassong. »Das Wortschatzspiel« wurde von Mario Pesch geschrieben.

## 6.1 Zufallszahlen

Im PXT-Editor findest du zwei Arten von Zufallswerten, die sich beide in der Befehlsgruppe »Mathematik« befinden.

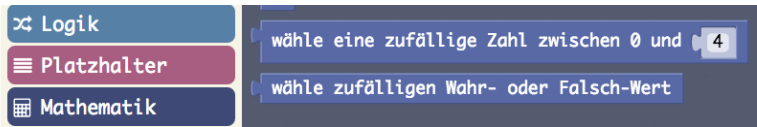


Abb. 6-1 Zufall-Blöcke in der Befehlsgruppe »Mathematik«

Den zufälligen Wahr- oder Falsch-Wert kannst du dir als einen Münzwurf vorstellen. Bevor du die Münze wirfst, fragst du dich, ob die Zahl oben liegen wird. Wenn die Seite mit der Zahl nach dem Wurf wirklich oben liegt, ist die Antwort auf deine Frage »wahr«, wenn aber die Seite mit der Zahl nach unten aufliegt, dann ist die Antwort »falsch, die Zahl liegt nicht oben«.

Besonders wichtig ist außerdem, dass du verstehst, wie der obere Zufallszahlen-Block funktioniert und wie du ihn für deine Projekte benutzen kannst. Zufallszahlen sind wie die Zahlen auf einem Würfel. Es ist reiner Zufall, welche Zahl von 1 bis 6 du würfelst. Es gibt aber auch Würfel mit einer anderen Seitenzahl als 6. Natürlich sind es dann keine echten Würfel mehr, wie du es in der Schule gelernt hast. Aber diese »Spezialwürfel« mit 8, 10, 12 oder 20 Seiten werden bei manchen Brettspielen genauso benutzt wie ein normaler Würfel. Nur dass diese Spezialwürfel eben eine zufällige Zahl zwischen 1 und 8 oder zwischen 1 und 20 anzeigen, je nachdem, wie viele Seiten dieser Würfel hat.



Abb. 6-2 Spezielle »Würfel« mit mehr als 6 Seiten und Zahlen

Computerprogramme beginnen mit dem Zählen oft bei der 0. So ist es auch wieder bei den Zufallszahlen. Auf einem Würfel wäre also auch eine 0 als mögliche Zahl. Wenn du in dem Zufallszahlen-Block eine Zahl zwischen 0 und 6 einstellst, dann würfelt das Programm mit einem 7-seitigen Spezialwürfel, der die Zahlen 0, 1, 2, 3, 4, 5 und 6 hat. Aber wahrscheinlich wolltest du wirklich einen Würfel

mit nur 6 Seiten und den Zahlen von 1 bis 6, oder? Dann musst du einen kleinen Trick im PXT-Editor machen. Und zwar lässt du das Programm nur eine Zahl zwischen 0 und 5 auswählen. Dann ist es ein Würfel mit 6 Seiten und den Zahlen 0, 1, 2, 3, 4 und 5. Kannst du dir schon vorstellen, wie du trotzdem auf die Zahlen 1, 2, 3, 4, 5 und 6 kommst? Es ist ganz einfach: Rechne zu der ersten Zufallszahl + 1 dazu. Dann kann dein Ergebnis nicht kleiner als 1 ( $0+1$ ) und höchstens 6 ( $5+1$ ) sein. Die im nächsten Bild verwendeten Blöcke befinden sich beide in der Befehlsgruppe »Mathematik«.

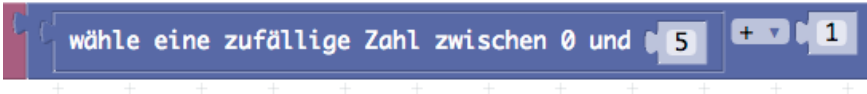


Abb. 6-3 Eine Zufallszahl von 1 bis 6 erzeugen – wie auf einem echten Würfel

Nachdem du nun die Besonderheiten von Zufallszahlen in Computerprogrammen kennst, kannst du mit dem ersten Spiel dazu anfangen.

## 6.2 Würfeln mit Schwung ...

Bisher hast du viel über Zahlen und Zufall gelernt. Jetzt bist du in der Lage, deine Erfahrungen über Zufallszahlen mit den Sensoren deines Calliope mini zu kombinieren. Das folgende Projekt zeigt dir, wie du einen Würfel programmieren kannst, der durch Schütteln deines Calliope-Boards die Zahlen anzeigt.



Abb. 6-4 Der Calliope mini als Würfel

## 6.2.1 Dein Calliope wird zu einem einfachen Würfel

Nachdem du im PXT-Editor ein »Neues Projekt« erstellt hast, kannst du mit der Programmierung beginnen. Im ersten Schritt wählst du aus der Rubrik »Eingabe« den Programmierbefehl »wenn geschüttelt« und schiebst ihn auf die Programmierfläche.

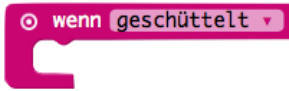


Abb. 6–5 Wenn geschüttelt ...

Dein Calliope mini merkt jetzt, wenn du es schüttelst. Allerdings musst du das Programm so erweitern, dass dein Board weiß, was es tun soll, wenn es geschüttelt wird. Hierzu fügst du dem Programm einen neuen Platzhalter hinzu. Du kannst den neuen Platzhalter mit dem Namen »Zufallszahl« versehen.

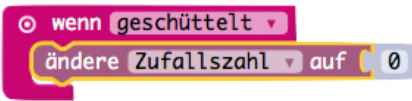


Abb. 6–6 Wenn geschüttelt, dann Zufallszahl ...

In diesem Zustand zeigt dein Calliope mini, wenn er geschüttelt wird, immer die Zahl 0 an.

Da du durch das Schütteln deines Calliope aber eine Zahl würfeln möchtest, musst du dein Programm noch ein wenig abändern. Dies tust du, indem du dein Programm mit dem Programmierbefehl »wähle zufällige Zahl zwischen« erweiterst. Du findest diesen Programmierbefehl in der Gruppe »Mathematik«.

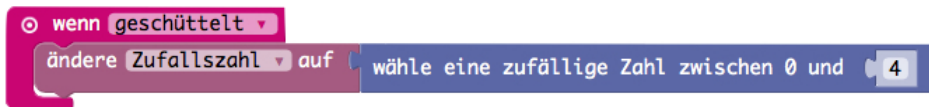


Abb. 6–7 Es soll eine zufällige Zahl zwischen 0 und 4 gebildet werden

Durch dieses Programm ist dein Calliope mini in der Lage, beim Schütteln eine Zahl zwischen 0 und 4 zu würfeln. Leider bekommst du das noch nicht mit, da der Calliope mini noch nichts anzeigt. Dafür musst du noch einen weiteren Befehl hinzufügen. Schieb hierzu ganz einfach den Programmierbefehl »zeige Nummer« auf die Programmierfläche. Achte darauf, dass du den Platzhalter mit dem von dir gewählten Namen »Zufallszahl« auswählst.



Abb. 6–8 Die Zufallszahl soll angezeigt werden

Du kannst dein Programm nun im Simulator im PXT-Editor testen, indem du entweder den Knopf zum Schütteln des Calliope mini drückst oder indem du die Maus über den Calliope-Simulator bewegst.

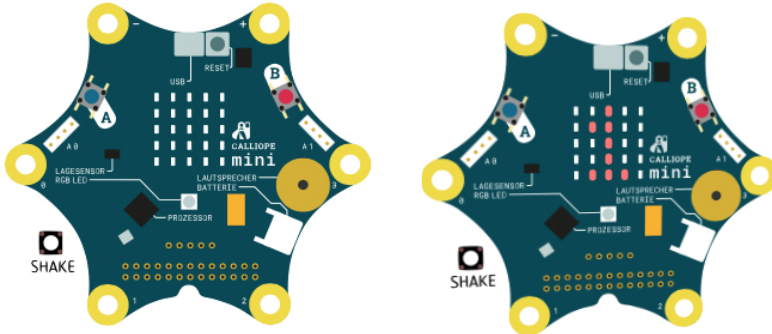


Abb. 6–9 Das Spiel im PXT-Simulator

Beim Herumspielen mit deinem Schüttel-Würfel wirst du schnell zwei ungewöhnliche Dinge feststellen. Zum einen kann dein Würfel bisher nur die Zahlen 0 bis 4 würfeln und zum anderen würfelt dein Calliope die Zahl 0, die auf einem richtigen Würfel nicht vorkommt.

Du hast sicher schon richtig erkannt, dass du dein Programm nur etwas verändern musst, um diese Probleme zu lösen. Am besten beginnst du mit der Lösung des ersten Problems. Hierzu ist kein neuer Programmbefehl notwendig. Du musst nur den Programmbefehl »wähle eine zufällige Zahl« etwas anpassen, indem du aus der 4 eine 6 machst.

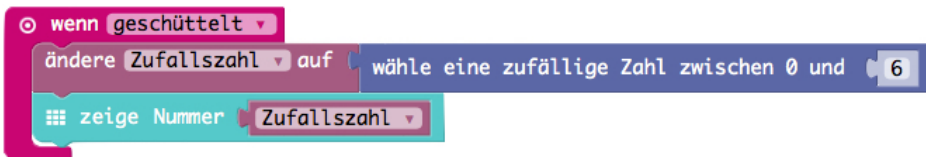


Abb. 6–10 Du passt die zufälligen Zahlen an

Du kannst nun dein Programm wieder im Simulator testen. Nach einigen Würfelversuchen wird dir auffallen, dass der Calliope im Simulator nun tatsächlich bis 6 würfelt. Jedoch wird hin und wieder immer noch die Zahl 0 angezeigt. Leider startet der Programmbefehl »wähle eine zufällige Zahl« immer mit der Zahl 0, somit wird die Zahl 0 auch immer wieder durch Zufall gewählt. Dieses Problem kannst du mit einem Trick umgehen. Hast du vielleicht selbst eine Idee, wie dieser Trick funktionieren könnte?

Dein Calliope mini merkt sich die gefundene Zufallszahl im Platzhalter mit dem Namen »Zufallszahl«. Erst danach gibt das Board die Zufallszahl mit dem Programmbefehl »zeige Nummer« aus. Stell dir vor, dein Calliope mini findet nun durch Zufall die Zahl 0. Wenn du nun die kleine Berechnung  $0 + 1$  durchführen

lässt, ändert sich die Zahl 0 im Platzhalter auf 1. Diese kleine Rechnung kannst du für jede vom Calliope mini gefundene Zufallszahl durchführen. Durch eine kleine Ergänzung deines Programms durch den Programmbefehl »ändere Zufallszahl um 1« hast du diese kleine Berechnung eingebaut.

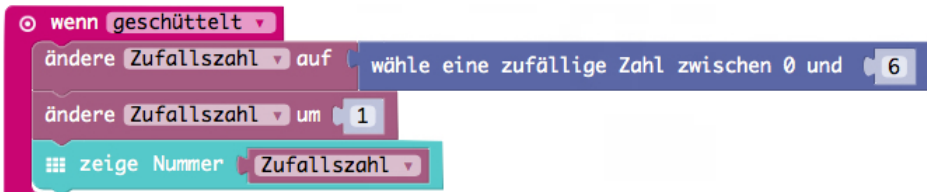


Abb. 6–11 Das Programm ergänzen um eine kleine Berechnung

Du weißt, dass die Berechnung für die Zahl 0 funktioniert. Was ist jedoch mit den Zahlen 1, 2, 3, 4, 5 und 6? Welches Ergebnis berechnet der Programmbefehl »ändere Zufallszahl um 1«, wenn der Calliope mini sich eine dieser Zahlen im Platzhalter gemerkt hat? Die folgende Tabelle hilft dir dabei, die Ergebnisse für die restlichen Zahlen zu überprüfen.

Zufallszahl	Rechnung	Ergebnis
0	+1	1
1	+1	2
2	+1	3
3	+1	4
4	+1	5
5	+1	6
6	+1	7

Tab. 6–1 Die Ergebnisse von »ändere Zufallszahl um 1« in einer Tabelle dargestellt

In der ersten Zeile steht, was du bereits weißt, nämlich dass die Berechnung für die Zufallszahl 0 funktioniert. Das Ergebnis ist die Zahl 1, und somit eine Zahl, die auf einem Würfel mit sechs Seiten vorkommt. Wenn du dir nun die letzte Zeile der Tabelle anschaust, siehst du, dass es ein Problem gibt, wenn die Zufallszahl 6 beträgt. In diesem Fall ergibt die Berechnung mit dem Programmbefehl »ändere Zufallszahl um 1« die Zahl 7. Aber die Zahl 7 kommt auf einem Würfel mit 6 Seiten nicht vor. Bei genauerem Betrachten der Ergebnisse in der Tabelle fällt dir sicher auf, dass die Zahlen 1 bis 6 im Ergebnis vorkommen. Nur die Zahl 7 stört hier. Aus diesem Grund lässt du die Berechnung aus der letzten Zeile »6+1« einfach weg. Hierzu änderst du dein Programm so ab, dass es nur noch Zufallszahlen von 0 bis 5 erzeugt.

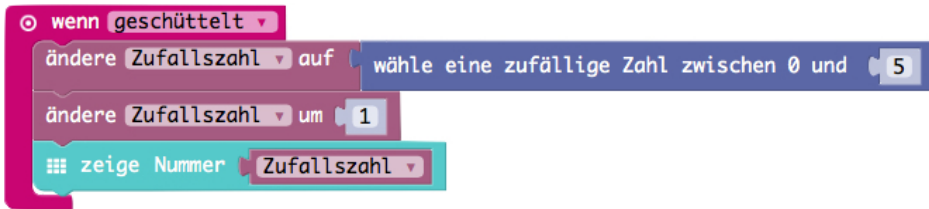


Abb. 6–12 Und noch eine kleine Änderung

Einige Versuche mit dem Simulator zeigen dir, dass dein Würfel nun die richtigen Zahlen, wie sie auf einem Würfel mit 6 Seiten vorkommen, anzeigt.

Dein Programm ist nun eigentlich fertig, aber findest du nicht auch, dass die Anzeige einfacher Zahlen auf deinem Calliope nicht wirklich wie ein echter Würfel aussieht? Schöner wäre es, wenn dir eine Augenzahl wie auf einem echten Würfel durch Punkte angezeigt würde.

## 6.2.2 Calliope-Schüttelwürfel mit Augenzahlen

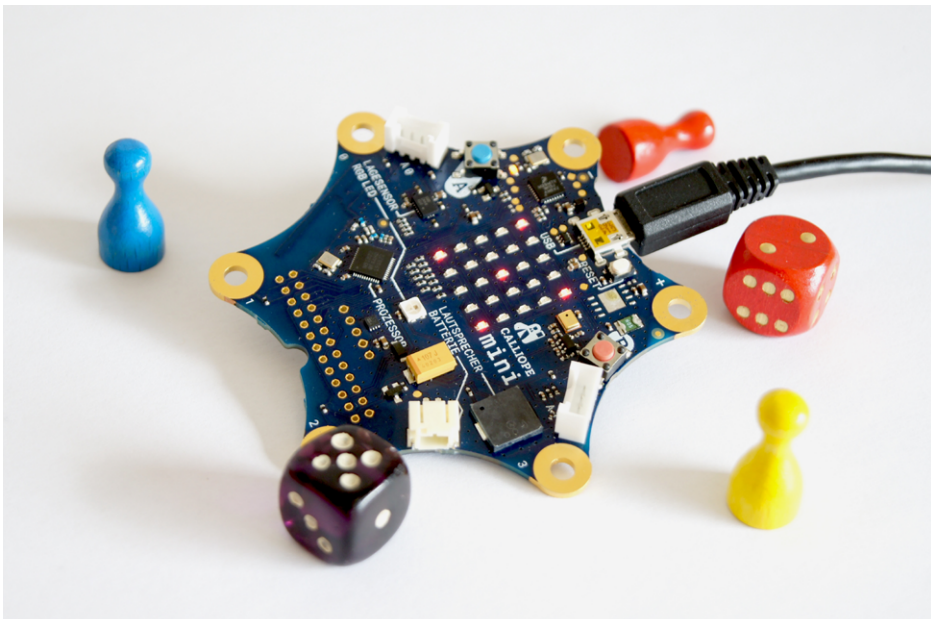


Abb. 6–13 Der Calliope mini stellt die Würfelzahl auch dar

Jetzt geht es darum, dein Würfelprogramm auch wie einen richtigen Würfel aussehen zu lassen. Hierzu verwendest du Programmierbefehle, die du bereits in Abschnitt 4.1.2 kennengelernt hast.

Zunächst beginnst du mit der Anzeige der Augenzahl mit Hilfe der LED-Anzeige. In Abschnitt 2.2 hast du gelernt wie du deinen Calliope mini zum Lachen bringst. Ähnlich funktioniert die Anzeige der Augenzahl. Bevor du beginnst, kannst du den Programmbefehl »zeige Nummer Zufallszahl« entfernen, denn den benötigst du nun nicht mehr.

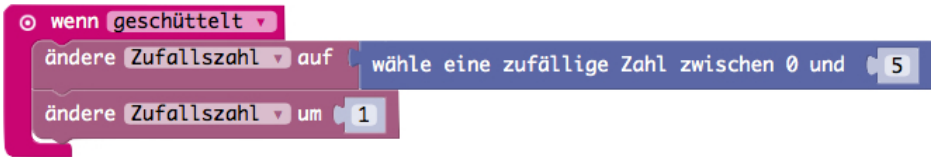


Abb. 6–14 Der Programmbefehl »zeige Nummer Zufallszahl« wurde entfernt

Es bleibt nur das Würfelprogramm ohne eine Ausgabe übrig. Füge nun den Programmbefehl »zeige LEDs« im Editor hinzu und markiere die LED in der Mitte. Diesen Programmbefehl benutzt dein Calliope mini später zum Anzeigen der Augenzahl 1.

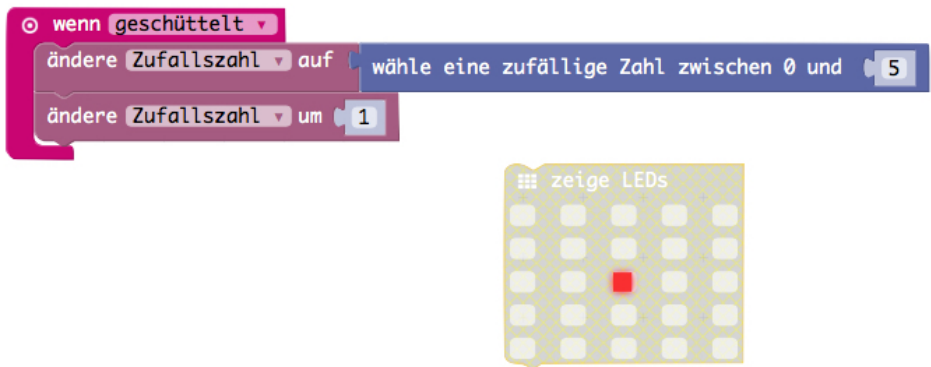


Abb. 6–15 Füge »zeige LEDs« hinzu

Diesen Schritt wiederholst so lange, bis du für jede Seite deines Würfels einen Programmbefehl erstellt hast.

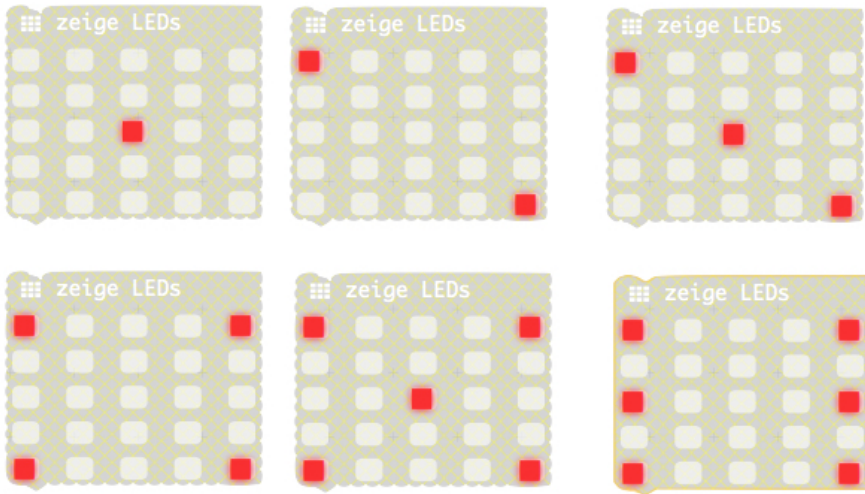


Abb. 6–16 Du stellst die Würfelaugen auf dem Calliope-Display nach

Du bist noch nicht ganz fertig, da du deinem Calliope mini auch beibringen willst, wann er die richtige Augenzahl anzeigen soll. Die eben erzeugten Programmbeefehle kannst du zunächst zur Seite schieben, da du jetzt etwas Platz benötigst, um einige »wenn dann«-Programmierbefehle in dein Programm einzubauen. Durch die »wenn dann«-Programmierbefehle entscheidet dein Calliope-Board, welche Augenzahl zu der gefundenen Zufallszahl passt.

Du beginnst mit der Augenzahl 1. Schiebe also einen »wenn dann«-Programmierbefehl auf die Programmierfläche. Dieser Befehl soll überprüfen, ob die Zufallszahl genau 1 ist. Wenn dies der Fall ist, soll auf der LED-Anzeige die passende Ausgabe angezeigt werden. Dein Programm sollte nun wie folgt aussehen:

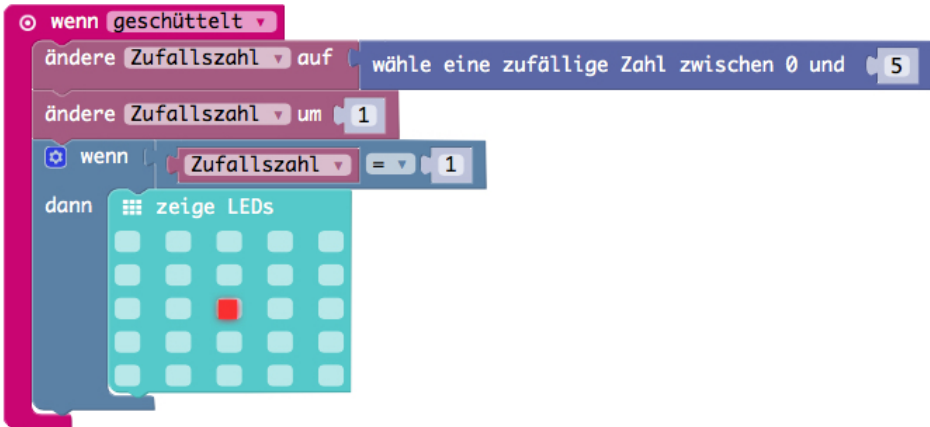
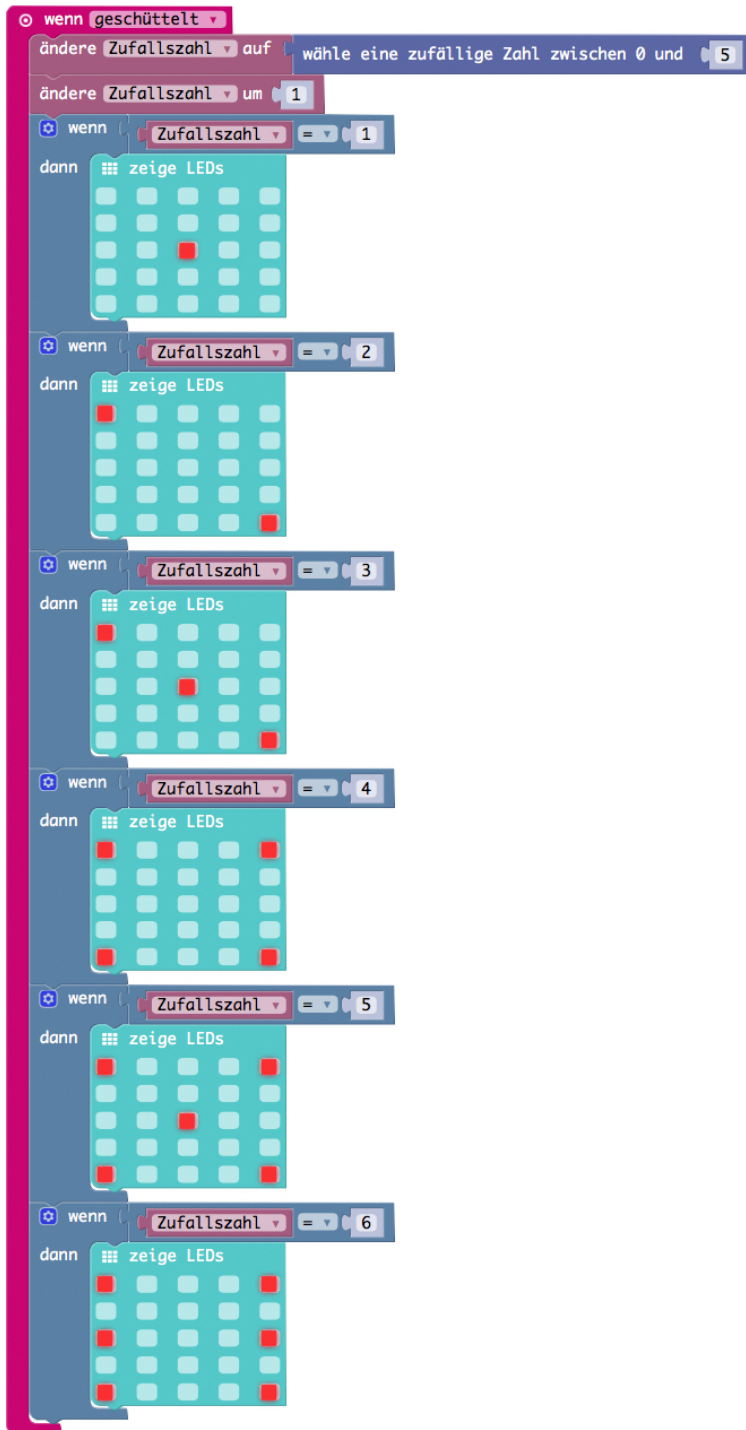


Abb. 6–17 Mit einem »wenn-dann«-Befehl kombinierst du Zufallszahl mit ihrer Darstellung.

Du kannst das Programm nun im Simulator testen, wirst aber feststellen, dass du einige Würfelversuche benötigst, bis die Augenzahl 1 angezeigt wird. Bei deinen restlichen Versuchen passiert nichts. Das liegt daran, dass du deinem Calliope mini bisher nur gesagt hast, was es machen soll, wenn die Zahl 1 gewürfelt wird. Deshalb musst du für die Augenzahlen 2, 3, 4, 5 und 6 auch solche »wenn dann«-Regeln in dein Programm einbauen. Da du mittlerweile schon ein echter Profi-Programmierer bist, sollte dir dieser Vorgang recht leicht fallen. Das nächste Bild zeigt dir, wie das fertige Programm aussieht.



Glückwunsch! Du hast nun einen Calliope-Würfel, den du zum Beispiel beim Spielen von Brettspielen benutzen kannst.

## 6.2.3 Was ist noch möglich?

- ▶ Wie wäre es, wenn dein Calliope mini nach dem Schütteln durch Piepsen des Lautsprechers sagt, dass er gerade eine Zufallszahl sucht.
- ▶ Du könntest aber auch versuchen, deinen Würfel so zu erweitern, dass er die Augenzahl durch Piepsen des Lautsprechers ausgibt.
- ▶ Vielleicht möchtest du aber auch einen Würfel haben, der mehr als 6 Seiten besitzt. Stell dir vor, du hättest einen Würfel mit 20 Seiten ...
- ▶ Manche Brettspiele benötigen einen Würfel, der Farben statt Zahlen anzeigt. Vielleicht kannst du dein Programm so verändern, dass dein Calliope Farben statt Zahlen anzeigt.

Die hier genannten Beispiele sind nur einige Ideen, wie du aus deinem einfachen Calliope-Würfel einen echten Hightech-Würfel machen kannst. Sicher hast du aber noch bessere Ideen!

## 6.3 Symbolschnapper

Bei diesem Spiel geht es darum, wer als Erstes das angezeigte Symbol schnappt, das auf dem Calliope mini dargestellt wird. Der Calliope mini liegt auf dem Tisch, wo ihn alle Mitspieler gut sehen können. Ihr könnt dieses Spiel zu zweit spielen oder mit mehr Mitspielern, aber alle müssen den Calliope mini gut sehen können. Jeder Spieler ist der Reihe nach dran und darf den Calliope mini schütteln. Wenn der Calliope mini geschüttelt wird, zeigt er eines von verschiedenen Symbolen. Welche Symbole das sind, habt ihr euch vorher überlegt und dem Calliope beigebracht.

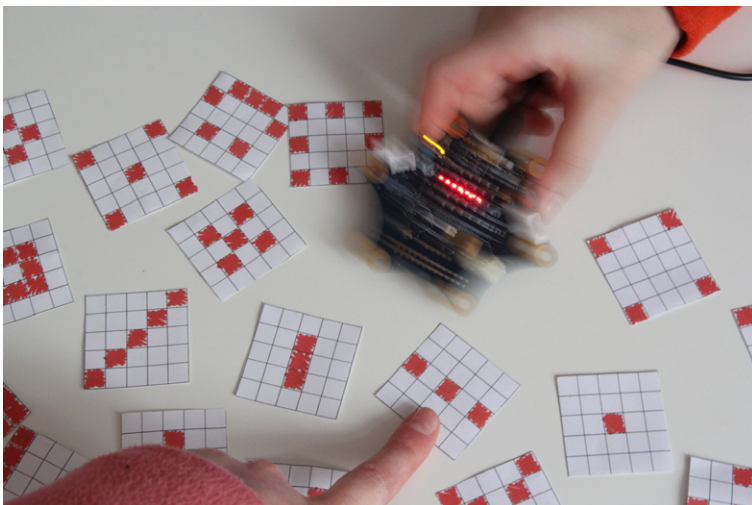


Abb. 6–18 Schüttele und schnapp dir das angezeigte Symbol.

Bevor du mit der Programmierung beginnst, musst du als Erstes die Symbolplättchen vorbereiten, die du für das Spiel verwenden willst.

### 6.3.1 Symbolplättchen ausmalen

Druck die Symbolvorlage aus, die du hier herunterladen kannst: <http://www.papiercomputer.de/informatik/calliope/symbolspiel>. Du siehst dort 20 Felder, die jeweils 25 Kästchen haben. Jedes Kästchen steht dabei für eine der 25 LEDs auf deinem Calliope mini. In diese Felder kannst du beliebige Symbole und Muster eintragen, indem du einzelne Kästchen rot (oder in irgendeiner anderen Farbe) ausmalst. Die ausgemalten Kästchen soll der Calliope später als leuchtende LED anzeigen.

#### Hinweis

Du kannst auch einfaches kariertes Papier benutzen, 5 x 5 Kästchen große Felder umranden und die einzelnen Kästchen ausmalen.

Schneide die Symbolplättchen noch nicht aus.

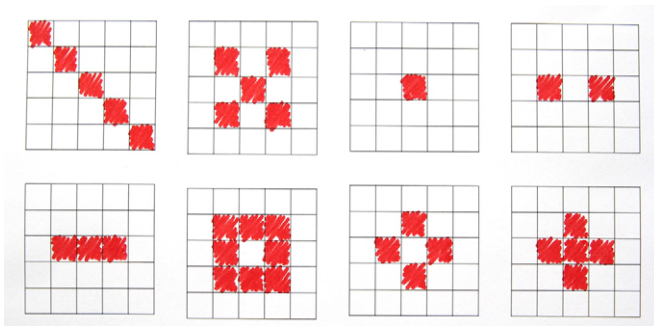


Abb. 6–19 Ausgefüllte Symbolvorlage mit Beispielsymbolen

Wenn es dir zu lange dauert, jedes Kästchen (fast) komplett auszufüllen, kannst du stattdessen auch nur ein Kreuz oder einen dicken Punkt in das Kästchen machen. Du und deine Mitspieler müssen aber gut erkennen können, welche der Kästchen markiert sind, damit sie das LED-Muster vom Calliope mini wiederfinden können.

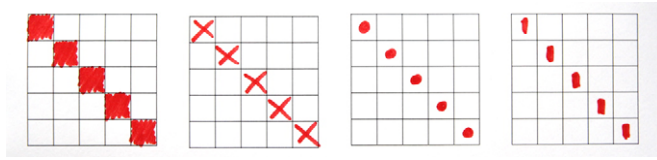


Abb. 6–20 Verschiedene Arten, die Symbolvorlage auszumalen

Vergiss nicht, dass die Plättchen nach dem Ausschneiden auch ganz verdreht auf dem Tisch liegen und die Muster dadurch anders aussehen können.

### Hinweis

Wenn du trotzdem willst, dass die Symbole alle als unterschiedlich erkannt werden, musst du vor dem Ausschneiden die Unterseite der Symbole besonders markieren, zum Beispiel mit einer dicken roten Linie.

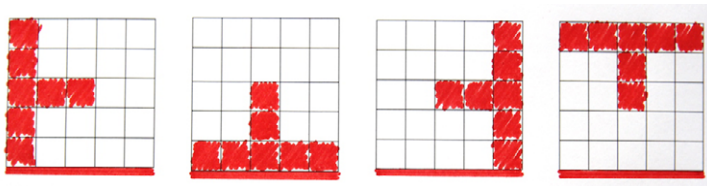


Abb. 6–21 Gleiches Muster mit markierter Unterseite

Nach dieser Vorbereitung kannst du nun den PXT-Editor öffnen und mit dem Programmieren beginnen. Erstelle ein neues Projekt und nenne es »Symbol-schnapper«.

## 6.3.2 Beim Schütteln ein bestimmtes Symbol anzeigen

Im vorherigen Kapitel hast du gelernt, wie du Platzhalter im PXT-Editor erstellst. Zur Erinnerung: Klicke in der Befehlsgruppe »Platzhalter« auf »Neuen Platzhalter anlegen«. Dann erscheint ein Fenster, in dem du dem neuen Platzhalter einen Namen geben kannst. Nenn diesen Platzhalter »Symbolzahl«, weil das Programm im nächsten Schritt das passende Symbol zu dieser Zahl anzeigen soll.

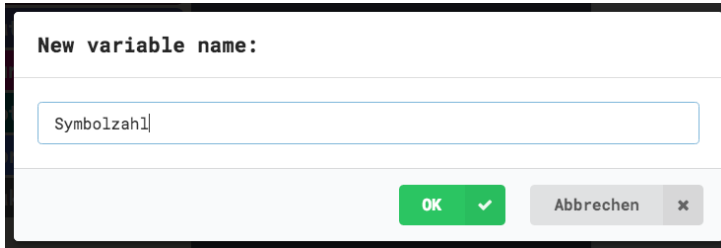


Abb. 6–22 Nenn den neuen Platzhalter »Symbolzahl«.

Hol dir nun einen »wenn dann«-Befehl aus der Befehlsgruppe »Logik« und eine »dauerhaft«-Klammer aus den »Grundlagen«. Das Programm soll prüfen, welchen Wert die Symbolzahl hat, und **wenn** die Symbolzahl 1 ist, **dann** soll es das erste Symbol auf dem LED-Display anzeigen. Dafür brauchst du noch einen Befehl aus der Befehlsgruppe »Logik«: »0 = 0«. Ersetze das »wahr« hinter dem »wenn« mit diesem »0 = 0«-Block und ändere eine der Zahlen in eine 1. An die Stelle der anderen Zahl ziehst du einen Block mit dem Platzhalter »Symbolzahl«.

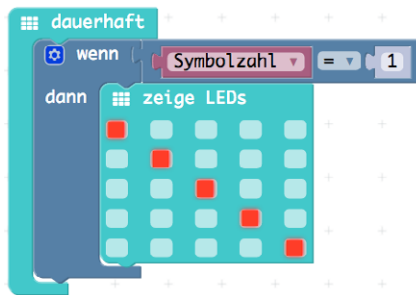


Abb. 6–23 Auswertung der Symbolzahl 1

Schau dir jetzt kurz an, was im Simulator auf der linken Seite passiert. Siehst du etwas? Warum nicht? Was glaubst du, wie groß die Symbolzahl jetzt ist? Richtig, da du keinen Befehl verwendet hast, um die Symbolzahl zu verändern, ist sie 0. Aber das Symbol wird nur gezeigt, wenn die Symbolzahl 1 ist. Du musst die Symbolzahl verändern, um etwas zu sehen. Wie du in der Spielbeschreibung gelesen hast, soll die Anzeige dann verändert werden, wenn der Calliope mini geschüttelt wird.

Als Nächstes brauchst du also einen Programmierbefehl, mit dem der Calliope mini auf Schütteln reagiert. Du findest den Befehl dafür in der Befehlsgruppe »Eingabe«. Danach wählst du den »ändere Platzhalter auf«-Befehl aus der Befehlsgruppe »Platzhalter« und ziehst ihn in die pinkfarbene Klammer. Klicke auf das kleine Dreieck und wähle deinen eigenen neuen Platzhalter »Symbolzahl« aus. Nun kannst du einstellen, dass die Symbolzahl bei jedem Schütteln auf 1 geändert wird. Probier dies im Simulator aus und bewege dort die Maus hin und her oder

drück einfach auf den neu erschienenen »SHAKE«-Knopf. Du wirst sehen, dass nun leuchtende LEDs in Form des von dir gemalten Symbols erscheinen.



Abb. 6–24 Das erste Symbol wird angezeigt.

Aber natürlich hast du mehr als nur dieses eine Symbol. Im nächsten Schritt wirst du darum alle anderen Symbole, die du dir vorhin ausgedacht hast, als LED-Muster im PXT-Editor einzufügen und einer Zahl zuordnen.

### 6.3.3 Alle LED-Symbole malen

Du könntest dir nun 19-mal immer wieder einen neuen »wenn dann«-Befehl aus »Logik« holen, dazu einen »0 = 0«-Block und einen »zeige LEDs«-Befehl aus den »Grundlagen«. Aber schon beim Lesen dauert dir das bestimmt viel zu lange, oder? Eine einfache und außerdem viel schnellere Möglichkeit besteht darin, dass du deinen schon vorhandenen »wenn dann«-Befehl verdoppelst. Dafür musst du mit der rechten Maustaste auf den »wenn dann«-Befehl klicken. Dann erscheint eine kleine Auswahlliste. Wähle dort den obersten Eintrag »Duplizieren« aus.

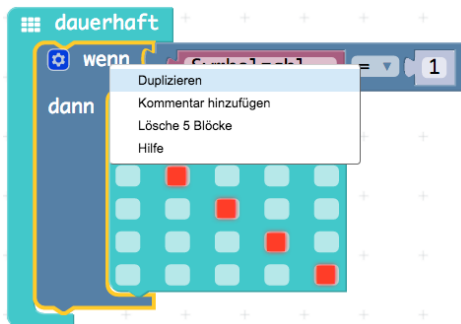


Abb. 6–25 Befehle duplizieren

Du bekommst dadurch einen neuen »wenn dann«-Befehl, der die gleichen Programmierbefehle und-blöcke beinhaltet. Klicke mit der linken Maustaste auf das

Wort »wenn« und zieh die ganze Kombination nach unten bis zu der Stelle zwischen dem ersten »wenn dann«-Befehl und dem Ende der Dauerschleife. Ändere die Zahl hinter »=« auf 2 und klicke so viele LEDs an oder aus, bis es aussieht wie das Symbol im zweiten Feld deiner ausgemalten Symbolvorlage.



Abb. 6–26 Symbolzahl 1 und Symbolzahl 2 zeigen verschiedene LEDs.

Wiederhole diesen »Duplizieren-Schritt« so oft, bis du genauso viele »wenn dann«-Befehle hast, wie du Symbolplättchen ausgemalt hast. Wenn du jedes Feld in der Symbolvorlage mit einem Symbol ausgemalt hast, dann brauchst du 20 »wenn dann«-Befehle. Der letzte »wenn dann«-Befehl sollte überprüfen, ob die Symbolzahl = 20 ist. Jede Zahl von 1 bis 20 sollte nur in einem »=«-Block vorkommen.

Geh zurück zu deiner »wenn geschüttelt«-Klammer und ändere die Symbolzahl dort auf irgendeine andere Zahl zwischen 1 und 20. Probiere wieder das Schütteln im Simulator aus. Dir wird das passende Symbol angezeigt, wie du es im »wenn dann«-Befehl zu dieser Zahl angegeben hast.

## Hinweis

Wenn deine LED-Anzeige ganz schnell zwischen zwei verschiedenen Symbolen hin und her wechselt, hast du einen Fehler gemacht und die gleiche Zahl in mehreren »wenn dann«-Befehlen verwendet. Dies kann leicht passieren, wenn du vergisst, die Zahl zu ändern, nachdem du die Befehle verdoppelt hast.

Probier irgendeine andere Zahl aus und schüttele noch einmal. Wenn du alle Symbole von der Symbolvorlage übertragen hast, kannst du die einzelnen Symbolplättchen ausschneiden.

### 6.3.4 Zufälliges Symbol anzeigen

Du willst natürlich etwas anderes, als die Zahl jedes Mal im Editor zu verändern. Stattdessen soll bei jedem Schütteln des echten Calliope mini irgendein zufälliges Symbol angezeigt werden.

Dafür gibt es im PXT-Editor einen Zufallszahl-Block. Du findest ihn in der Befehlsgruppe »Mathematik«. Der Block hat eine Ausbuchtung auf der linken Seite, du kannst ihn also überall dort benutzen, wo eine Zahl gebraucht wird.

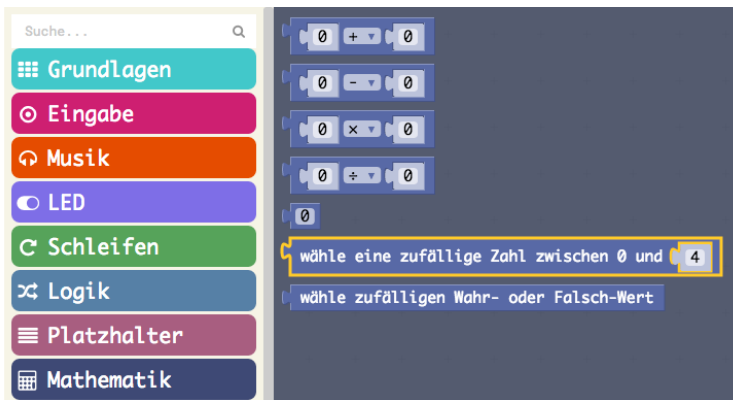


Abb. 6–27 Zufallszahl-Block in der Befehlsgruppe »Mathematik«

Setz den »Zufallszahlen«-Block an die Stelle mit der Zahl, die du soeben selbst verändert hast. Auf der Symbolvorlage hast du Platz für 20 Symbole, also wirst du wahrscheinlich zuerst versuchen, eine Zufallszahl von 0 bis 20 zu bekommen, wenn geschüttelt wird. Aber weil die Zufallszahlen im PXT-Editor immer bei 0 anfangen, würfelst du in Wahrheit einen Würfel mit 21 Seiten mit den Zahlen 0, 1, 2 bis 19 und 20. Da du aber keinen Symbolblock zur Symbolzahl 0 hast, musst du die Zufallszahl um 1 erhöhen. Du findest den »+ 0«-Block ebenfalls in der Befehlsgruppe »Mathematik«. Ersetze dort eine 0 durch den Zufallsblock und die andere 0 durch 1. Nun brauchst du aber nicht  $20+1$  Symbolplättchen, sondern  $19+1=20$  Symbolplättchen.

Damit du den Wechsel von einem Symbol zu einem anderen Symbol besser sehen kannst, kannst du bei jedem Schütteln den Bildschirm kurz löschen. Benutze dafür den Befehl »Bildschirminhalt löschen« aus der Befehlsgruppe »Grundlagen«.



Abb. 6–28 Zufällige Symbolzahl beim Schütteln

Da die Symbolzahl in der Dauerschleife ständig überprüft wird und das passende Symbol dazu anzeigt, führt eine Veränderung der Symbolzahl auch zur Änderung der LED-Anzeige. Probier dein Programm im Simulator aus. Du kannst das Spiel schon jetzt auf das echte Calliope mini überspielen und alleine oder mit deinen Freunden versuchen, das angezeigte Symbol als Erstes vom Tisch zu schnappen. Dabei kann es passieren, dass ein Symbol, das schon weggeschnappt wurde, noch einmal auf der LED-Anzeige erscheint, weil zufällig die gleiche Zufallszahl wieder (und wieder) ausgewählt wird vom Programm. Das ist nicht schlimm, weil ihr dann einfach noch einmal schütteln könnt.

### 6.3.5 Punkte zählen

Bisher ist es so, dass du die angezeigte Symbolkarte schnappen und vor dir ablegen kannst, um am Ende zu wissen, wie viele Symbole du am schnellsten gefunden hast. Nun kannst du versuchen, das Programm so zu verändern, dass es die Punkte von dir und deinem Mitspieler zählen kann. Du wirst sicherlich keine Probleme mehr haben, einen neuen Platzhalter für die Punkte anzulegen und ihn bei jedem Knopfdruck auf Knopf »A« zu erhöhen. Zeige auch etwas auf den LEDs an, damit ihr seht, dass die Punkte erhöht wurden.

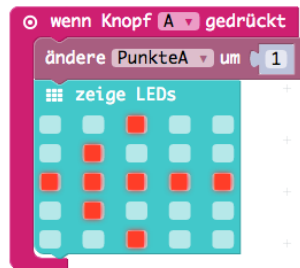


Abb. 6–29 Punkte von Spieler A erhöhen

Mach das Gleiche für deinen Mitspieler B. Wenn du beide Spielerpunkte im Programm hast, könnt ihr die Symbolplättchen auf dem Tisch liegen lassen. Der Spieler, der am schnellsten war, darf dann einfach auf seinen Knopf »A« oder »B« drücken und erhält einen Punkt dazu. Dann ist es auch nicht schlimm, wenn das gleiche Symbol mehrmals angezeigt wird.

Überlege dir anschließend selbst, wann euer Spiel oder eine Runde beendet sein soll. Wenn ein Spieler 8 Punkte hat? Dann müsstest du »dauerhaft« überprüfen, ob die Punkte größer als (>) 7 sind. Oder wenn ihr auf beide Knöpfe gleichzeitig drückt und wer dann die meisten Punkte hat? Dir fällt bestimmt noch etwas anderes ein, probier es aus.

Abb. 6–30 Beispiel für Spielende bei Knopfdruck auf »A« und »B« gleichzeitig

Wenn du das Spiel schon heruntergeladen und ausprobiert hast, hast du vielleicht bemerkt, dass die LED-Anzeige beim Schütteln mehr als einmal verändert wird. Das liegt daran, dass die Befehle in der »wenn geschüttelt«-Klammer mehrmals ausgeführt werden. Um das zu verhindern, kannst du auch einen kurzen »pausiere«-Befehl einfügen, damit das Programm 1 Sekunde wartet, bevor es auf das nächste Schütteln reagieren kann. Wenn dein Code ungefähr so aussieht wie auf dem nächsten Bild, kannst du das Spiel auf den Calliope mini herunterladen und alleine oder mit Freunden spielen.

Abb. 6–31 Das fertige Programm für das Spiel »Symbolschnapper« (Ausschnitt).

## 6.3.6 Noch mehr Zufall

Wie wäre es mit der Anzeige eines ganz zufälligen Musters? Im Projekt Sanduhr (siehe Abschnitt 5.3) hast du gesehen, wie du die LEDs schrittweise durchgehen kannst, um einzelne LEDs auszuschalten. Dafür brauchst du in diesem Projekt auch die neuen Platzhalter »reihe« und »sprünge«. Beginne mit einer leeren LED-Anzeige und geh durch jede einzelne LED durch, aber ohne eine Pause zu machen. Bei jeder LED, also in der inneren »Wiederholen«-Klammer wählst du einen zufälligen Wahr- oder Falsch-Wert oder eine zufällige Zahl zwischen 0 und 1. Entweder die LED ist zufällig an (wahr oder 1) oder die LED ist zufällig aus (falsch oder 0). Der Calliope mini wertet den Platzhalter »LEDleuchtet« aus und entscheidet, ob er die LED an dieser Stelle zeichnen, also LED anmachen, oder (ab)schalten soll. Dadurch wird das angezeigte LED-Muster sehr zufällig.

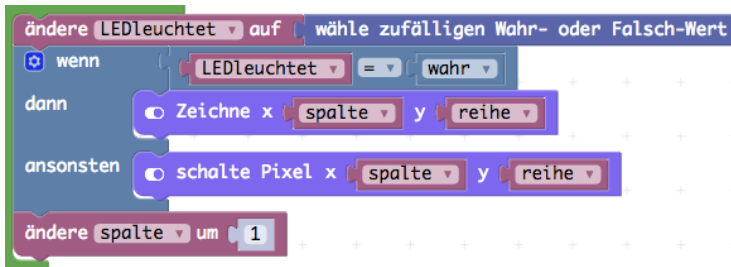


Abb. 6–32 LEDs zufällig an- oder ausmachen

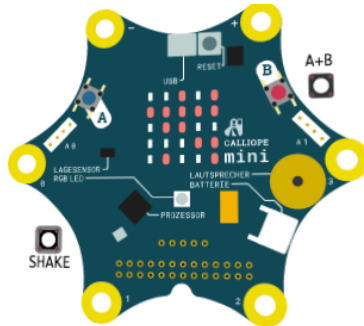


Abb. 6–33 Zufälliges Muster wird angezeigt.

Wie musst du die grünen »Wiederholen«-Klammern bauen und wie und wo musst du die Platzhalter »reihe« und »spalte« verändern (+1 oder auf 0), damit jedes Mal beim Schütteln ganz neue Muster angezeigt werden? Schau dir die Sanduhr noch einmal an, wenn du Tipps brauchst.

Mit dieser Variante könntet ihr nun ein Spiel spielen, bei dem ihr zum Beispiel am schnellsten zählen müsst, wie viele LEDs auf dem Calliope mini leuchten. Wer als Erster die richtige Zahl zwischen 0 (alle Lampen aus) und 25 (alle an) ruft, darf auf seinen Knopf drücken und erhält einen Punkt.

Wenn dir das zu einfach war, hier noch eine Zusatzaufgabe: Lass den Calliope mini selbst ausrechnen und anzeigen, wie viele LEDs angeschaltet sind. Dafür braucht ihr einen neuen Platzhalter, den ihr nach jedem »zeichne«-Befehl um 1 erhöht. Ihr könnt euch das Ergebnis dann zum Beispiel durch Drücken auf »A+B« anzeigen lassen und seht sofort, ob eure selbst gezählte Antwort richtig war. Danach drehst du das Display nach unten um, und der Platzhalter wird wieder auf 0 gesetzt.

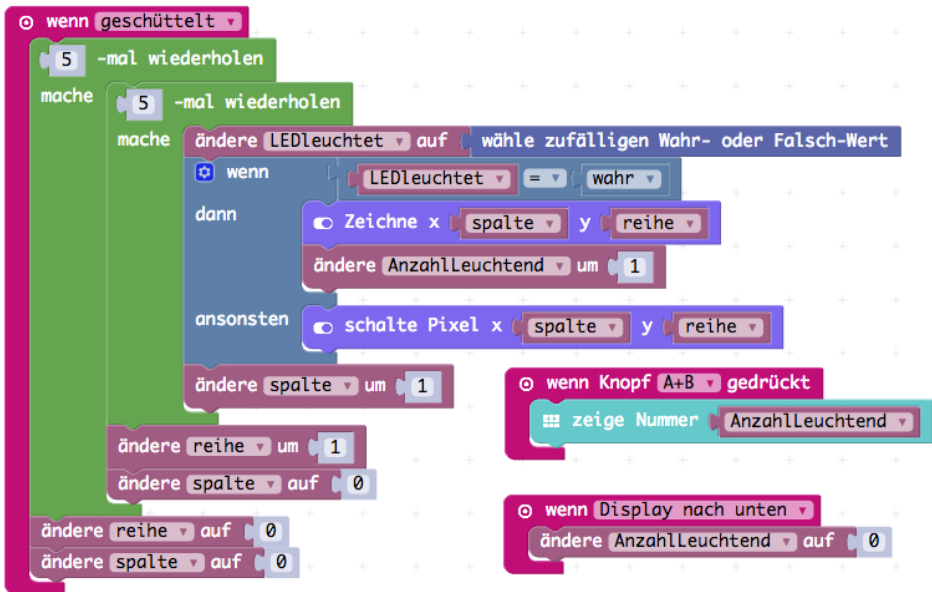


Abb. 6–34 Variante mit zufälligem Muster, das schnell gezählt werden soll.

## 6.4 Das Reaktionsspiel

### 6.4.1 Wer ist die oder der Schnellste?

Lass uns doch einmal deine Reaktionsgeschwindigkeit und die deiner Freunde testen! Das geht natürlich am besten mit einem Reaktionsspiel! In diesem Abschnitt kannst du ein solches Reaktionsspiel erstellen. Du darfst es gerne nach eigenen Wünschen verändern und anpassen. Du könntest zum Beispiel andere Symbole auf dem LED-Display benutzen oder andere Wartezeiten verwenden. Aber nun geht es erst einmal los!

Die Grundidee des Spiels ist, dass auf dem LED-Display plötzlich ein Symbol erscheint und dann beide Spieler möglichst schnell auf den passenden Knopf drücken, also Spieler A auf Knopf »A« und Spieler B entsprechend auf Knopf »B«. Ihr haltet den Calliope mini also bei diesem Spiel zwischen euch.

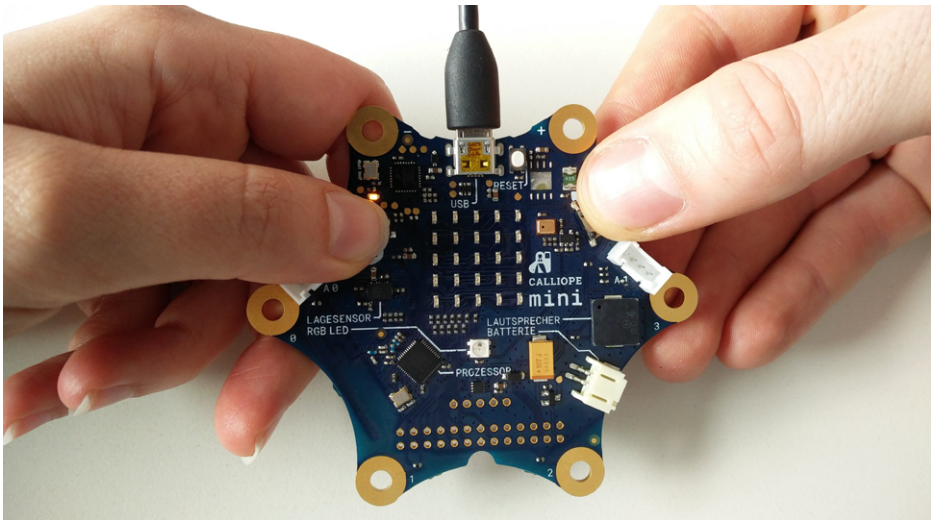


Abb. 6–35 Position der Spieler beim Reaktionsspiel

## 6.4.2 Symbol nach zufälliger Zeit anzeigen

Ganz entscheidend für dieses Spiel ist, dass das Symbol immer nach einer zufälligen Zeit erscheint, damit sich kein Spieler darauf vorbereiten kann. Wie ihr ein Symbol (zum Beispiel ein Smiley oder ein Herz) auf dem Calliope mini anzeigt, wisst ihr ja bereits. Das ist also nun die erste Aufgabe. Lasst ein Symbol dauerhaft auf dem LED-Display anzeigen.

Damit das Symbol aber nicht direkt beim Start des Programms angezeigt wird, braucht dein Programm vorher noch eine Pause. Mit dem Befehl »pausiere (ms)« kannst du festlegen, wann das Smiley erscheint. Jetzt kommt der Zufall ins Spiel!

Damit das Spiel spannend bleibt, soll das Smiley immer nach 1 bis 5 Sekunden erscheinen. Da man in den Befehl »pausiere (ms)« eine Zeit in Millisekunden eingeben muss, soll die Zufallszahl also zwischen 1000 und 5000 liegen. Jetzt ist ein bisschen Mathematik gefragt.

Suche zuerst den Baustein für Zufallszahlen. Kleiner Tipp: Hier geht es um »Mathematik«. Dieser Befehl erstellt allerdings eine Zufallszahl zwischen 0 und 4 (oder einer anderen oberen Grenze). Damit die Zufallszahl also zwischen 1000 und 5000 liegt, musst du zuerst eine 1 hinzuaddieren.

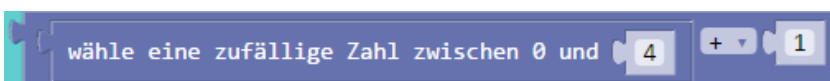


Abb. 6–36 Befehl für eine Zufallszahl zwischen 1 und 5

Mit diesem Befehl berechnet der Calliope mini also eine Zufallszahl zwischen 1 und 5. Du musst sie in einem zweiten Schritt nur noch mit 1000 multiplizieren, dann hast du eine Zufallszahl zwischen 1000 und 5000.

Sehr gut, der erste Schritt ist geschafft, das Smiley erscheint zufällig nach 1 bis 5 Sekunden.

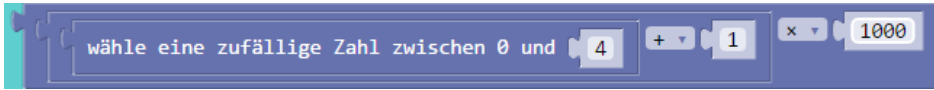


Abb. 6–37 Befehl für eine Zufallszahl zwischen 1000 und 5000

Bisher passiert dies jedoch nur beim allerersten Start des Programms. Du kannst dies testen, indem du das Programm auf den Calliope mini überträgst und zum Neustarten den Reset-Knopf benutzt. Später soll das Smiley wieder verschwinden, wenn einer der beiden Spieler korrekt den Knopf gedrückt hat.

### 6.4.3 Platzhalter speichert, ob »Smiley an«

Jetzt geht es darum, zu messen, welcher Spieler zuerst den Knopf gedrückt hat. Dazu kümmerst du dich am besten zuerst um Spieler A mit Knopf »A«. Das Gleiche für den zweiten Spieler nachzubauen, ist dann später gar nicht mehr so schwer.

Los geht es also mit dem Befehl »wenn Knopf A gedrückt« (Tipp: Hier geht es um »Eingaben«). Was soll denn dann passieren? Na ja, wenn das Smiley leuchtet, soll der Spieler einen Punkt bekommen. Dazu muss der Calliope mini aber wissen, ob das Smiley leuchtet, sonst bekommt nachher auch ein Spieler Punkte, der viel zu früh gedrückt hat. Da der Calliope mini selbst nicht sehen kann, was auf dem LED-Display angezeigt wird, brauchst du hierfür einen Platzhalter. Falls du nicht (mehr) weißt, wozu man einen Platzhalter braucht und wie man ihn benutzt, dann schau dir die Projekte in Kapitel 5 an.

Damit du später nicht durcheinanderkommst, solltest du den Platzhalter eindeutig benennen. Wie wäre es zum Beispiel mit »SmileyAn«? Da ein solcher Platzhalter nur Zahlen speichern kann, musst du hier ein wenig tricksen. Nimm doch einfach an, dass wenn in dem Platzhalter eine 0 gespeichert ist, dies bedeutet, dass der Smiley nicht leuchtet, und wenn im Platzhalter eine 1 gespeichert ist, das Smiley leuchtet.

Nun musst du dein Programm in der Dauerschleife so erweitern, dass der Platzhalter »SmileyAn« auf 1 gesetzt wird, wenn das Smiley leuchtet. Wenn das Smiley wieder ausgeschaltet wird, muss auch der Platzhalter wieder auf 0 gesetzt werden, aber darum kannst du dich später noch kümmern. Füge also den Befehl »ändere SmileyAn auf 1« in deiner Dauerschleife direkt hinter die Anzeige des Smileys ein.

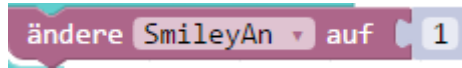


Abb. 6–38 Platzhalter »SmileyAn« muss eine 1 speichern, wenn das Smiley angezeigt wird.

## 6.4.4 Auf Knopfdruck reagieren

In der Abfrage »wenn Knopf A gedrückt« musst du nun zusätzlich abfragen, ob das Smiley leuchtet. Das machst du mit dem »wenn-dann«-Befehl, den du zu Beginn von Kapitel 4 kennengelernt hast. Bei der Bedingung musst du den Wert des Platzhalters »SmileyAn« mit einer »1« vergleichen, denn wenn eine »1« im Platzhalter gespeichert ist, bedeutet dies ja, dass das Smiley gerade leuchtet.

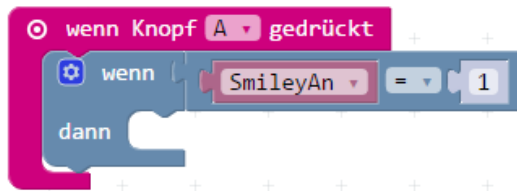


Abb. 6–39 Bedingung, wenn Taster gedrückt und Smiley leuchtet

Wenn Spieler A also seinen Knopf »A« drückt, während das Smiley leuchtet, dann hat er richtig reagiert. Für einen Zwischentest kannst du einen Ton ausgeben lassen, damit du siehst, ob auch alles klappt. Später wird in diesem Moment der schnellere Spieler einen Punkt erhalten. Zum Testen kannst du schon drücken, bevor das Smiley erscheint, und dann, während das Smiley leuchtet. Der Ton darf nur dann ertönen, wenn das Smiley leuchtet und du gleichzeitig auf Knopf »A« drückst. Damit du mehrfach testen kannst, einfach mit dem »Reset«-Knopf das Spiel neustarten.

## 6.4.5 Das Smiley muss verschwinden

Allerdings gibt es nun ein Problem ... Das Smiley leuchtet immer weiter, und auch der Ton ertönt immer weiter. So könnte Spieler A ja ganz viele Punkte sammeln. Also muss das Smiley wieder verschwinden.

Um das Smiley verschwinden zu lassen, kannst du den Befehl »Bildschirminhalt löschen« benutzen. Du musst ihn direkt unter den Befehl zur Tonausgabe einfügen.

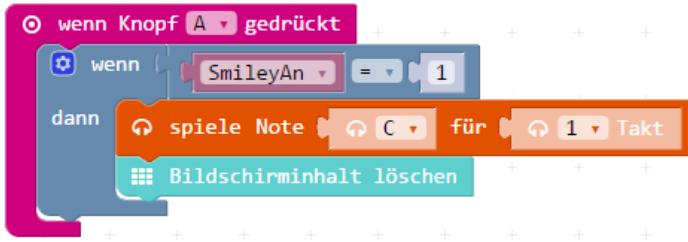


Abb. 6–40 Löscht das Smiley, wenn Spieler A richtig reagiert hat.

Wenn du dein Spiel nun testest, solltest du einen Fehler feststellen. Das Smiley ist weg, aber Spieler A sammelt trotzdem weiter Punkte. Das liegt daran, dass das Calliope mini immer noch denkt, das Smiley würde leuchten, da der Platzhalter »SmileyAn« immer noch eine »1« speichert.

Dafür musst du in dein Programm zum Drücken von Knopf »A« noch den Befehl »ändere »SmileyAn« auf »0« (unter den Befehl »Bildschirminhalt löschen«) einfügen.



Abb. 6–41 Setzt zusätzlich den Platzhalter »SmileyAn« auf 0

Es ist Zeit für einen nächsten Test. Kontrolliere dabei einmal die wichtigsten Funktionen deines Programms:

- ▶ Ertönt der Ton wirklich nur, wenn Spieler A im richtigen Moment Knopf »A« drückt, also wenn das Smiley leuchtet?
- ▶ Verschwindet das Smiley, wenn Knopf »A« gedrückt wurde und der Ton ertönt ist?
- ▶ Leuchtet das Smiley nach einer kurzen Pause (von 1 bis 5 Sekunden) erneut auf?
- ▶ Bleibt das Smiley so lange sichtbar, bis Knopf »A« gedrückt wurde?

Jetzt kann Spieler A keine weiteren Töne erzeugen, also Punkte sammeln, bis das Smiley nach einer weiteren zufälligen Pause wieder erscheint. Hmm. Punkte sammeln, das war ja das Ziel. Darum solltest du dich jetzt kümmern.

## 6.4.6 Punkte zählen

Leg dafür einen weiteren Platzhalter »PunkteSpielerA« an. Darin sollen die Punkte für Spieler A gesammelt werden. Dieser Platzhalter muss also bei jeder richtigen Reaktion (dem Drücken auf Knopf »A«, wenn das Smiley leuchtet) um 1 erhöht werden. Dazu musst du den Befehl »ändere PunkteSpielerA um« verwenden.

### »ändere auf« und »ändere um«

Bei der Auswahl des Befehls zum Ändern eines Platzhalters musst du ganz genau hinschauen. Der Befehl »ändere Platzhalter auf« setzt den Platzhalter immer auf den gleichen Wert. Der Spieler hätte also immer nur 1 Punkt. Mit dem Befehl »ändere Platzhalter um« erhält der Spieler in jeder Runde einen Punkt dazu. Hier wird also immer 1 Punkt addiert. Du könntest in den Befehl »ändere Platzhalter um« auch eine »-1« schreiben, dann würde dem Spieler ein Punkt abgezogen.

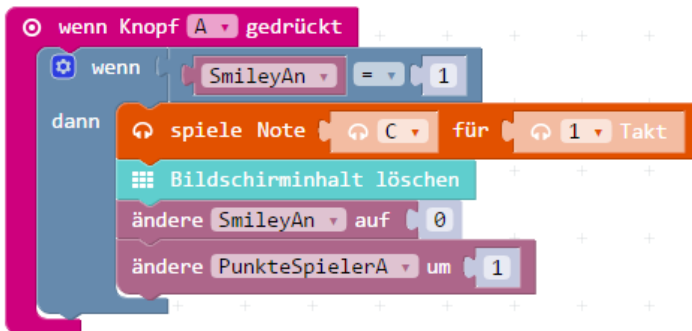


Abb. 6–42 Punktestand von Spieler A wird um 1 erhöht.

## 6.4.7 Gegenspieler B kommt ins Spiel

Das Programm, das du nun für Spieler A (mit Knopf »A«) gemacht hast, musst du für Spieler B (mit Knopf »B«) nun wiederholen. Du kannst entweder das Programm noch einmal nachbauen oder du duplizierst das Programm und machst dann passende Änderungen. Nicht vergessen: Spieler B braucht einen eigenen Platzhalter für die Punkte. Der Platzhalter »SmileyAn« muss dabei nicht dupliziert werden, da ja beide Spieler auf dasselbe Smiley reagieren. Es hilft dir beim Testen, wenn für Spieler B ein anderer Ton abgespielt wird.

Zeit für einen Test zu zweit. Such dir einen Gegner und testet eure Reaktionsgeschwindigkeit. Nun sammeln beide Spieler eifrig Punkte, aber am Ende wollt ihr natürlich auch sehen, wer öfter den Reaktionstest gewonnen hat und damit Gesamtsieger ist.

### 6.4.8 Punkte anzeigen

Jetzt sollen die Punkte, die in den Platzhaltern »PunkteSpielerA« und »PunkteSpielerB« gespeichert werden, auch angezeigt werden. Du könntest den Calliope mini so programmieren, dass der Spielstand immer angezeigt wird, wenn du den Calliope schüttelst. Dafür brauchst du als Erstes den Befehl »wenn geschüttelt«.

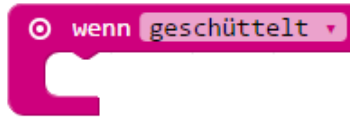


Abb. 6–43 Befehl »wenn geschüttelt«

Da hinein kannst du nun die Ausgabe der Punkte programmieren. Du könntest zum Beispiel erst die Zeichenfolge »Spieler A« anzeigen lassen, dann seinen Punktestand mit dem Befehl »zeige Nummer« und dem Platzhalter »PunkteSpielerA«.

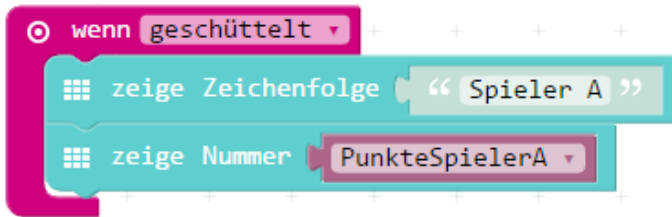


Abb. 6–44 Punkteausgabe für Spieler A

Die gleichen beiden Befehle kannst du nun auch für Spieler B einfügen. Na, hat es geklappt? Funktioniert dein Spiel? Ja, perfekt, dann fordere deine Familie und Freunde heraus. Testet, wer von euch die besseren Reaktionen hat. Falls das Spiel noch nicht ganz so funktioniert wie gewünscht, schau dir noch einmal die obigen Abschnitte genau an und prüfe, an welcher Stelle vielleicht ein Befehl fehlt.

### 6.4.9 Erweiterungen

Mit diesem Kapitel hast du eine erste Version des Reaktionsspiels gebaut, das du nun nach eigenen Wünschen erweitern und verschönern kannst. Hier findest du ein paar Ideen, wie du das Spiel noch verbessern könntest:

- ▶ Falls es dir zu lange dauert, die kompletten Zeichenfolgen »Spieler A« und »Spieler B« abzuwarten, könntest du stattdessen auch nur ein »A« und ein »B« anzeigen lassen.
- ▶ Du könntest einem Spieler, der zum falschen Zeitpunkt seinen Knopf drückt, also wenn das Smiley nicht leuchtet, auch einen Punkt abziehen. Dazu musst du in den Programmen für Knopf »A« und Knopf »B« in den »wenn-dann«-Befehl noch ein »ansonsten« (auf Englisch »else«) hinzufügen. Dazu einfach auf das kleine blaue Zahnrad klicken. Dort kannst du dann mit dem Befehl »ändere PunkteSpielerA« den Punktestand mit -1 verringern.
- ▶ Bei falschen Reaktionen könnte zusätzlich ein (möglichst unangenehmer) Ton abgespielt werden.
- ▶ Das Spiel könnte automatisch enden, wenn der erste Spieler 10 Punkte hat. Dazu musst du ein wenig Logik in die Dauerschleife einbauen. Tipp: Wenn »PunkteSpielerA = 10« oder »PunkteSpielerB = 10« dann...«. Du könntest als Finale einen Text mit dem Spielstand ausgeben lassen, die RGB-LED bunt leuchten lassen oder auch ein Feuerwerk auf dem LED-Display anzeigen.
- ▶ Auch könntest du eine maximale Spieldauer festlegen. Dazu musst du in der Dauerschleife ebenfalls einen »wenn-dann«-Befehl einbauen. Aber statt die Punktzahl mit 10 zu vergleichen, vergleichst du die Laufzeit des Calliope mini seit dem Start mit einer gewünschten Zeit (zum Beispiel 3 Minuten). Den Befehl »Laufzeit (ms)« findest du in der Gruppe »Eingabe« unter »Mehr«.

Bestimmt fallen dir aus den anderen Projekten noch tolle Ideen ein, wie du dein Reaktionsspiel weiter verbessern kannst. Viel Spaß dabei – und bloß nicht zu langsam drücken!

## 6.5 Farben-Bingo

### 6.5.1 Was ist denn Farben-Bingo?

Hast du schon einmal Bingo gespielt? Das Farben-Bingo wird fast genauso gespielt, allerdings mit Farben anstelle von Zahlen. Und das Beste ist, dass du die Farben selbst programmieren kannst. Das Farben-Bingo kannst du mit beliebig vielen Freunden spielen. Alles, was ihr braucht, ist ein Calliope mini, Papier und Stifte.

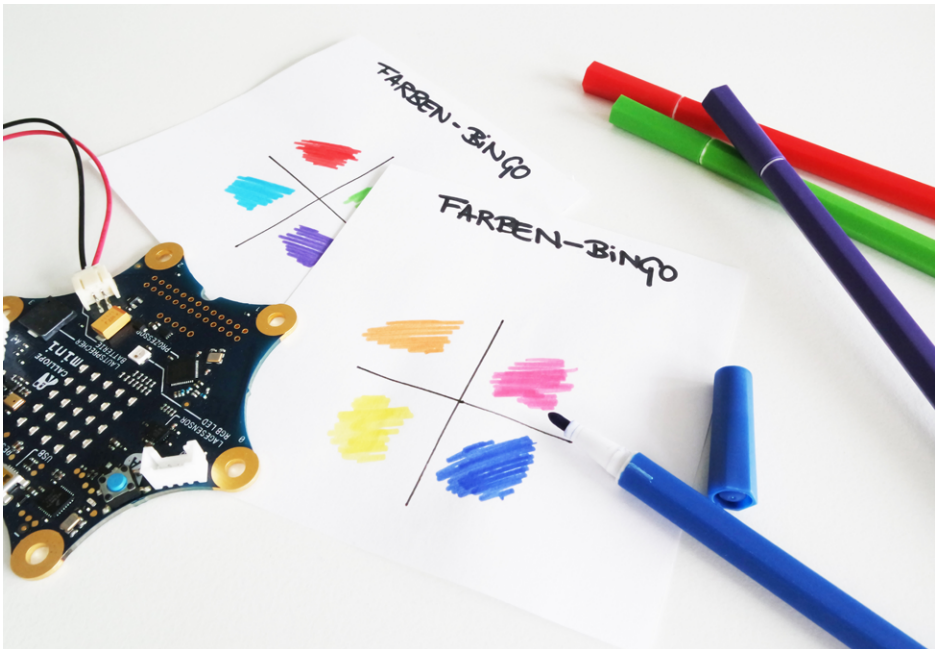


Abb. 6–45 Das fertige Farben-Bingo Spielfeld

## 6.5.2 Materialien und Werkzeuge

Wie viele?	Was	Anmerkung
1	Calliope mini	
1	Batterie-Pack	
2	Batterien	
1	Papier	5 cm
8	Stifte	div. Farben

Tab. 6–2 Materialien für das Farben-Bingo

## 6.5.3 Wie funktioniert Bingo?

Beim klassischen Bingo zeichnest du  $5 \times 5$  Kästchen auf ein Blatt Papier und trägst in jedes Kästchen eine Zahl von 1 bis 75 in das Spielfeld ein. Das mittlere Kästchen bleibt dabei leer. Nun wird zufällig eine Zahl zwischen 1 und 75 gezogen und laut vorgelesen. Immer wenn eine dieser Zahlen auf deinem Spielfeld steht, kannst du sie durchstreichen. Der erste Spieler, der alle Zahlen durchgestrichen hat, gewinnt.



Abb. 6–46 Spielfeld Zahlen-Bingo

Das Farben-Bingo ist ein Spiel, das sich wunderbar mit dem Calliope-Board spielen lässt, da bereits ein Knopf und eine mehrfarbige LED auf dem Board sind. Wie beim Zahlen-Bingo zeichnest du ein Kästchenfeld auf dein Papier, aber anstelle von Zahlen malst du unterschiedliche Farben in die Felder.

In deinem Bingo-Programm speicherst du beliebig viele Farben und lässt die LED per Knopfdruck zufällig in einer dieser Farben aufleuchten. Hast du die entsprechende Farbe in einem deiner Kästchen vermerkt, kannst du es durchstreichen. Wie beim Zahlen-Bingo gewinnt der erste Spieler, der alle Kästchen durchgestrichen hat. Wie wäre es, wenn wir zunächst mit 6 Farben und 4 Kästchen anfangen?

### Nicht genügend Buntstifte?

Falls du gerade keine Buntstifte in allen Farben hast, kannst du einfach den Namen der Farbe in das Kästchenfeld schreiben.

## 6.5.4 Das Farben-Bingo-Programm

Auch wenn das Spiel leicht zu spielen ist, steckt einiges an Logik und sogar Mathe dahinter. Du solltest also ein bisschen Programmiererfahrung mitbringen. Aber keine Angst, wenn du das Konzept einmal verstanden hast, wiederholt sich das Einfügen der Farben immer wieder. Schauen wir es uns in den nächsten Schritten doch genauer an.

## 6.5.5 Auslösen des Farbwechsels

Du arbeitest auch in diesem Projekt mit dem PXT-Editor. Der Link hierzu lautet <https://mini.pxt.io/>

Zu Beginn kannst du dir überlegen, wie eine neue Farbe angezeigt werden soll. Vielleicht, wenn du Knopf A oder B drückst? Oder sogar beide Knöpfe gleichzeitig? In unserem Beispiel leuchtet immer eine neue Bingo-Farbe auf, wenn Knopf A gedrückt wird. Den Baustein »wenn-Knopf-A-gedrückt« findest du unter »Eingabe«.



Abb. 6–47 Auslösen des Farbwechsels

## 6.5.6 Was passiert zwischen den Farbwechseln?

Immer wenn du Knopf »A« drückst, wird dir im Spiel eine neue Farbe angezeigt. Damit der Farbwechsel besser zu sehen ist, kannst du die LED für eine beliebige Zeit ausschalten, bevor eine neue Bingo-Farbe angezeigt wird. Das kann besonders bei sehr ähnlichen Farben von Vorteil sein. Hierfür baust du die folgenden drei Programmierbefehle aus der Gruppe »Grundlagen« in dein Programm ein:

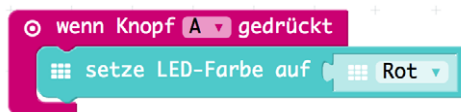


Abb. 6–48 Block »LED-Farbe« einsetzen

Anschließend setzt du den Grundlagenblock »Rot-Grün-Blau-Weiß« in den Platzhalter »Rot«.

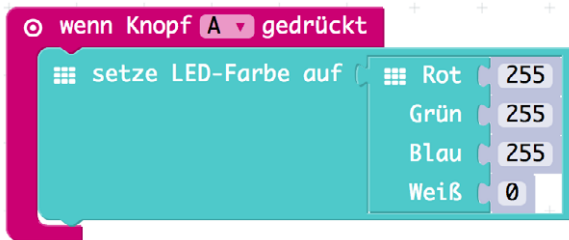


Abb. 6–49 Farben einsetzen

Damit deine mehrfarbige LED nicht mehr leuchtet, setzt du im nächsten Schritt alle Farbwerte auf 0. Per Doppelklick kannst du ein Feld auswählen und den Wert 255 durch 0 ersetzen.



Abb. 6–50 LED-Farben auf 0 setzen

Nun schiebst du den Block »pausiere« aus dem »Grundlagen«-Menü unter den »LED-Farbe«-Block. Weißt du schon, wie lange das Licht aus sein soll? In dem Block steht bereits eine Pause von 100 Millisekunden, also weniger als eine Sekunde. Das ist ganz schön kurz – oder kannst du die Pause etwa doch sehen? Am besten probierst du aus, wie viele Millisekunden deine Pause lang sein soll. In diesem Programm wird die Pause auf 1000 Millisekunden, also eine ganze Sekunde, verlängert.

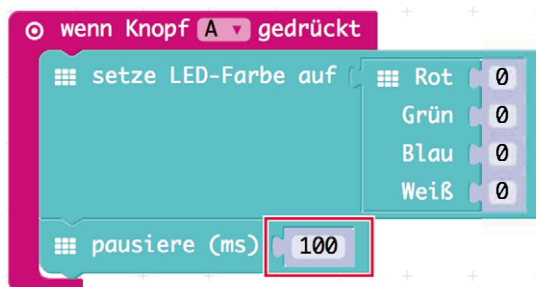


Abb. 6–51 Pausenlänge bestimmen

## 6.5.7 Die zufällige Farbausgabe programmieren

Wunderbar, dein Programm kann jetzt per Knopfdruck die mehrfarbige LED ausschalten, damit du den Farbwechsel siehst. Was natürlich noch fehlt, sind die eigentlichen Bingo-Farben und ihre zufällige Ausgabe. Hast du schon eine Idee, wie das programmiert werden kann?

Füge zunächst den Programmierbefehl »ändere-Platzhalter-auf« in dein Programm ein. Du findest ihn im Menü unter Platzhalter.

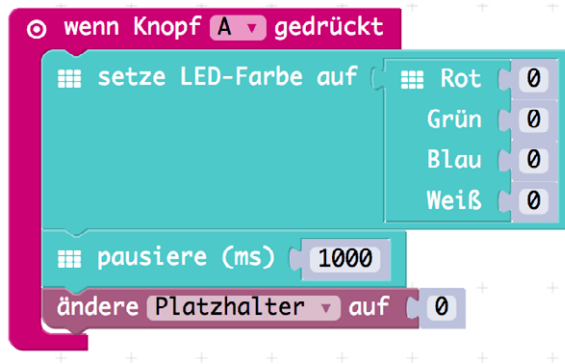


Abb. 6–52 »Bingo-Farben«-Platzhalter einsetzen

Per Doppelklick auf das Wort »Platzhalter« kannst du es umbenennen. Wie wäre es mit »Bingo-Farbe«? Danach klickst du auf das grüne OK-Feld, um deine Eingabe zu bestätigen.



Abb. 6–53 »Bingo-Farbe«-Platzhalter einsetzen

Anschließend brauchst du einen »Mathe«-Programmierbefehl, denn damit lassen sich Zufälle programmieren. Nimm hierfür den Programmierbefehl »Wähle-eine-zufällige-Zahl-zwischen-1-und-4«. Die Zahl 4 kannst du anschließend in eine beliebige Zahl ändern, je nachdem, mit wie vielen Farben du Bingo spielen möchtest.

Weiter vorne im Buch hast du ja schon gelernt, dass Programmierer mit dem Zählen immer bei 0 anfangen. In diesem Fall steht also 0 für die erste Farbe. In

diesem Programmbeispiel wurde die zweite Zahl zu einer 5 geändert. Jetzt kann also mit 6 Farben gespielt werden.



Abb. 6–54 Programmierung für die zufällige Ausgabe

### 6.5.8 Farben einfügen

Der letzte Schritt des Bingo-Spiels besteht darin, die Farben zu programmieren. Hierfür brauchst du den »Logik«-Block »wenn-dann«. Immer »wenn« der Knopf A gedrückt und zufällig eine Bingo-Farbe zwischen 0 und 5 ausgesucht wird, »dann« wechselt die LED zu der Farbe mit der entsprechend zugeordneten Nummer. Das Einfügen der Logik sieht als Programmierbefehl so aus:

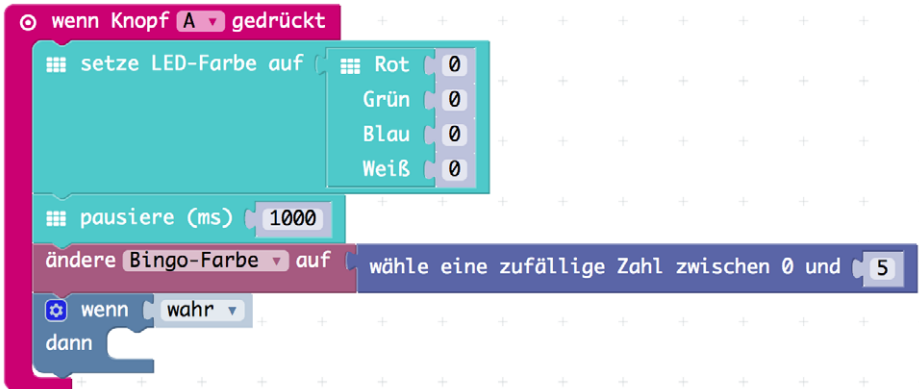


Abb. 6–55 Einfügen der Logik

Dann schiebst du einen zweiten »Logik-Block«, den Vergleich mit dem Gleichheitszeichen, hinter das »wenn« ein. Damit dein Programm die Bingo-Farbe mit der zufälligen Zahl vergleichen kann, ersetzt du die erste 0 mit deinem »Bingo-Farbe«-Platzhalter, den du im Menü »Platzhalter« wiederfindest.

Bitte wundere dich nicht: Da das Programm immer länger wird, ist der obere Teil der Programmierbefehle auf den folgenden Bildern abgeschnitten. Am Ende

des Projekts kannst du den ganzen Code noch einmal sehen und ihn mit deinem vergleichen.

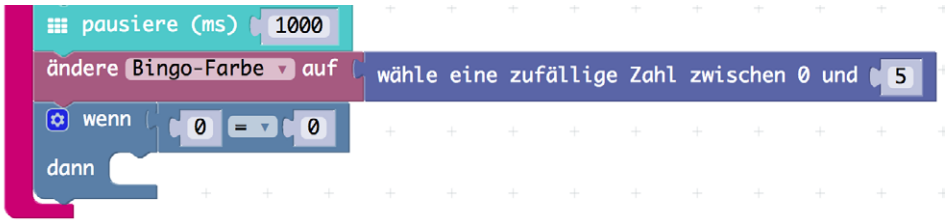


Abb. 6–56 Einfügen des Vergleichs

Nun kannst du den »setze-LED-Farbe-auf«-Block einschieben, den du unter »Grundlagen« findest. Dein Programm weiß jetzt, dass der Farbe »Rot« die Zahl 0 zugeordnet ist.

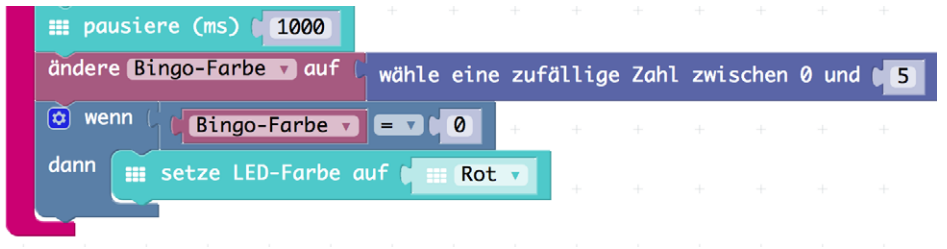


Abb. 6–57 Bingo-Farbe aussuchen

Diesen Schritt wiederholst du, bis jeder Zufallszahl, also 0 bis 5, eine Farbe zugeordnet ist. Mit einem Doppelklick auf die Farbe »Rot« öffnet sich ein Menü, und du kannst zwischen verschiedenen Farben wählen. In dem unteren Bild wurde zum Beispiel die Farbe »Orange« ausgesucht.

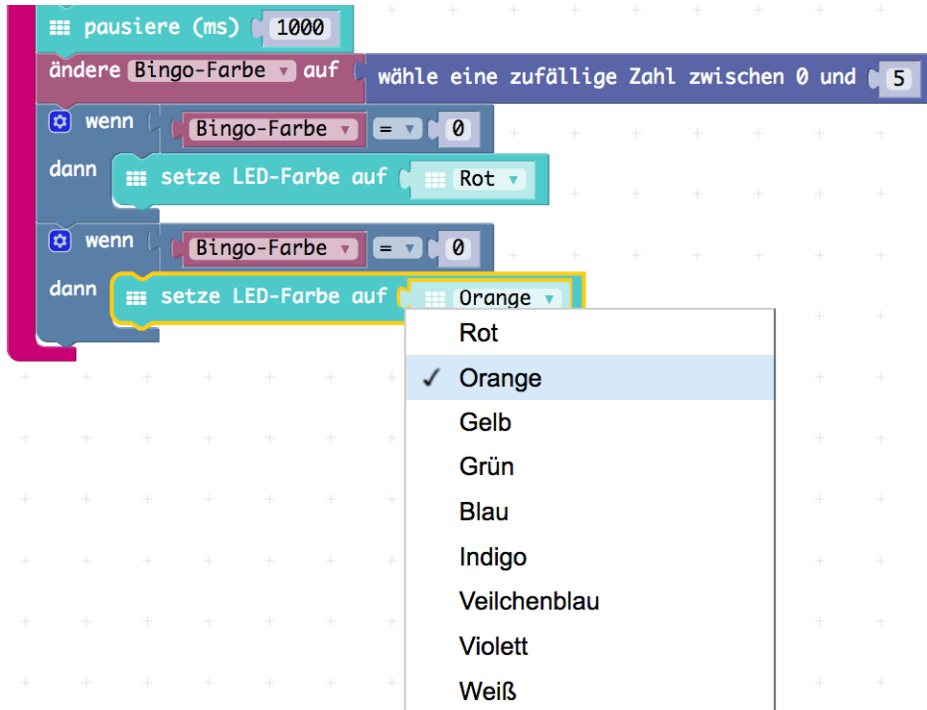


Abb. 6–58 Zwischen unterschiedlichen Farben wählen

Vergiss nach der Farbwahl nicht, auch deiner Bingo-Farbe eine neue Zahl zuzuweisen. Um es einfach zu machen, kannst du mit der Zahl 1 anfangen. Am Ende sollte dein Bingo-Programm in etwa so aussehen:

```
wenn Knopf A gedrückt
  setze LED-Farbe auf [Rot]
  pause (ms) 1000
  ändere Bingo-Farbe auf wähle eine zufällige Zahl zwischen 0 und 5
  wenn [Bingo-Farbe] = 0
    dann setze LED-Farbe auf [Rot]
  wenn [Bingo-Farbe] = 1
    dann setze LED-Farbe auf [Orange]
  wenn [Bingo-Farbe] = 2
    dann setze LED-Farbe auf [Gelb]
  wenn [Bingo-Farbe] = 3
    dann setze LED-Farbe auf [Grün]
  wenn [Bingo-Farbe] = 4
    dann setze LED-Farbe auf [Blau]
  wenn [Bingo-Farbe] = 5
    dann setze LED-Farbe auf [Violett]
```

The image shows a Scratch script for a color bingo game. It starts with a 'when button A is clicked' event. The first block is 'set LED color to' with a dropdown menu showing 'Rot', 'Grün', 'Blau', and 'Weiß', all set to '0'. This is followed by a 'pause (ms)' block set to '1000'. Then, the 'change bingo color to' block is used, with a 'choose a random number between 0 and 5' block. Below this, there are seven 'if-then' blocks. Each 'if' block checks the 'Bingo-Farbe' variable against a number from 0 to 5. The corresponding 'then' block sets the LED color to a specific color: 0 to Rot, 1 to Orange, 2 to Gelb, 3 to Grün, 4 to Blau, and 5 to Violett.

Abb. 6–59 Fertiges »Farben-Bingo«-Programm

Wunderbar, du hast dein erstes eigenes Bingo-Spiel programmiert. Jetzt ran an Stift und Papier, um zu testen, ob alles gut funktioniert. Das Spielfeld kann so aussehen wie in diesem Beispiel:



Abb. 6–60 Viel Spaß beim Spielen

## 6.5.9 Erweiterung des Programms

Natürlich gibt es noch viele Möglichkeiten, wie du dein Bingo-Programm erweitern kannst. Vor allem, wenn du mit mehr als nur 6 Farben spielen möchtest, sollten sich die Farben besser nicht wiederholen, oder? Vielleicht hast du ja sogar schon eine Idee, wie das programmiert werden kann. Viel Spaß beim Bingo-Spielen!

## 6.6 Das Wortschatzspiel

### 6.6.1 Wie viele Wörter kennst du?

Wie viele Wörter kennst du? Genau das kannst du mit dem nächsten Spiel herausfinden. Dafür wirst du in diesem Projekt den Calliope mini als Buchstabengenerator programmieren. Der Calliope mini wird einen Buchstaben generieren und eine Stoppuhr starten. Dann liegt es an dir! Du musst dir Mitspieler suchen und dann könnt ihr den Calliope mini in der Reihe herumgeben und jeweils ein Wort mit diesem Buchstaben am Anfang nennen. Irgendwann ertönt ein Signal und der Countdown ist abgelaufen. Wer nun den Calliope mini in der Hand hält, hat diese Runde verloren. Es heißt also, schnell und kreativ in der Wortwahl sein.

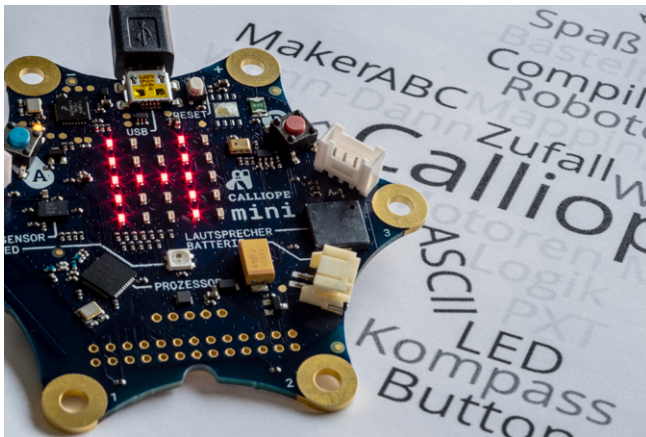


Abb. 6–61 Wie viele Wörter kennst du?

### 6.6.2 Die Vorbereitung

Für das Projekt solltest du bereits wissen, wie du Zufallszahlen mit dem Calliope erstellst.

### 6.6.3 Calliope mini unter der Lupe

Für das Spiel verwendest du verschiedene Bestandteile des Calliope-Boards. Du benutzt verschiedene Funktionen, um das Spiel zu starten, und verwendest den kleinen eingebauten Lautsprecher als Start- und Stoppsignal.

## 6.6.4 Die Programmierung

Die Grundidee des Spieles ist, dass der Calliope mini einen zufälligen Buchstaben auswählt, der auf dem LED-Display angezeigt wird, und einen Countdown mit unterschiedlicher Zeitlänge startet. Du hast bereits gesehen, wie du mit dem Calliope mini zufällig Zahlen erstellen kannst, eine Funktion zum Erstellen von zufälligen Buchstaben gibt es aber nicht. Jedoch als erfahrener Programmierer kannst du einen kleinen Trick verwenden, um die zufälligen Zahlen in Buchstaben zu wandeln. Der Calliope mini versteht ASCII-Code.

### ASCII-Code

ASCII ist eine internationale Zeichencodierung, die bereits 1963 von der American Standards Association als Standard etabliert wurde. Der ASCII-Code besteht insgesamt aus 128 verschiedenen Zeichen, die das lateinische Alphabet in Groß- und Kleinbuchstaben, die zehn arabischen Ziffern sowie einige Interpunktionszeichen und Sonderzeichen enthält. Jedem Zeichen ist auch eine normale Zahl zugewiesen, sodass normale Zahlen in Buchstaben umgewandelt werden können. Die folgende Tabelle zeigt dir, wie das aussieht:

Dezimalzahl	Schriftzeichen ASCII
65	A
66	B
67	C
68	D
69–90	E–Z

Es lassen sich somit aus den Dezimalzahlen 65 bis 90 normale Großbuchstaben erstellen, die du bereits kennst. Der ASCII-Code enthält auch kleine Buchstaben, die mit den Zahlen 97 (a) bis 122 (z) erstellt werden können.

**Eine vollständige Tabelle mit allen ASCII Zeichen findest du auch bei Wikipedia:**

[https://de.wikipedia.org/wiki/American\\_Standard\\_Code\\_for\\_Information\\_Interchange](https://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange)

## 6.6.5 Der Start

Das Spiel soll immer mit einem Druck auf den Knopf A starten. Wähl dazu aus der Gruppe »Eingabe« den Programmierblock »Wenn Knopf A gedrückt« aus. Als Nächstes musst du einen Platzhalter mit dem Namen »Zufallszahl« erstellen und ihm eine zufällige Zahl zwischen 0 und 25 zuordnen.

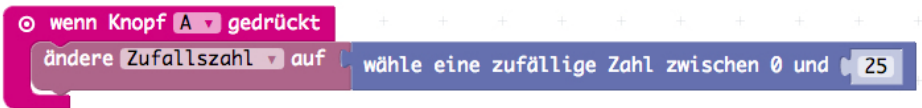


Abb. 6–62 Ziehen der Zufallszahl bei Knopfdruck

## 6.6.6 Das Erstellen der Buchstaben

Nachdem du die Zufallszahl erstellt hast, kannst du sie in einen Buchstaben umwandeln. Für diese Funktion wählst du Block »Text aus ASCII-Code« aus der Gruppe »Mathematik – More« aus. Damit du die Zufallszahl in einen Buchstaben mit Hilfe des ASCII-Code umwandeln kannst, musst du zur Zufallszahl noch 65 addieren. So deckst du den Bereich zwischen 65 und 90 ab (siehe dazu auch die Themeninsel zum ASCII-Code). Mit dem Befehl »zeige Zeichenfolge« aus der Gruppe »Grundlagen« kannst du den Zufallsbuchstaben dann auf dem LED-Display anzeigen lassen.

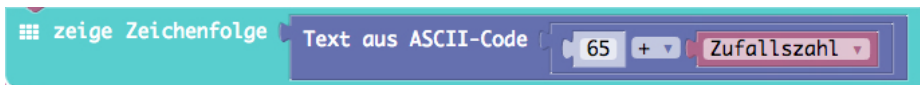


Abb. 6–63 Umwandlung der Zufallszahl mit Hilfe des ASCII-Codes

## 6.6.7 Das Startsignal

Nun sind die ersten wichtigen Schritte geschafft. Es wird Zeit, das Spiel auf den Calliope zu übertragen und einen ersten Text zu machen. Übertrage das Programm und teste, ob der Calliope verschiedene Buchstaben erstellt.

Für das Spiel musst du das Programm nun so erweitern, dass ein Ton abgespielt wird, während die Zeit läuft. Dazu kannst du aus dem Menü »Musik« den Block »Klingelton« auswählen und einfügen. Bei der Zahl kannst du verschiedene Varianten ausprobieren. Wenn du zum Beispiel 20 (Hz) eintippst, macht der Calliope mini einen Ton, der sich wie eine sehr schnell laufende Uhr anhört.

Das Spiel wird umso spannender, wenn das Spiel in jeder Runde unterschiedlich lang ist. Auch hier arbeitest du wieder mit der Zufallszahl. Du benötigst also einen neuen Platzhalter, den du »Zeit« nennst, und setzt ihn zu Beginn auf die Zahl 15000. Wusstest du, dass die Zeit in Millisekunden angegeben wird? Daher ist die Zahl so groß. Der Calliope mini soll nun eine Pause machen, die durch den Platzhalter »Zeit« bestimmt wird. Die Pause findest du in den Grundlagen unter »pausiere (ms)«.

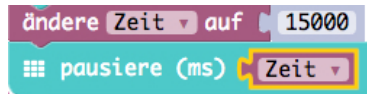


Abb. 6–64 Pausieren des Programms

Wenn du nun das Programm überträgst, wirst du feststellen, dass die Zeit, die das Programm pausiert, immer gleich lang ist. Um die Zeitschritte unterschiedlich lang zu gestalten, addierst du zum Platzhalter »Zeit« eine Zufallszahl. Damit die Zufallszahl nicht zu klein ist, multiplizierst du sie mit dem Wert 10000.

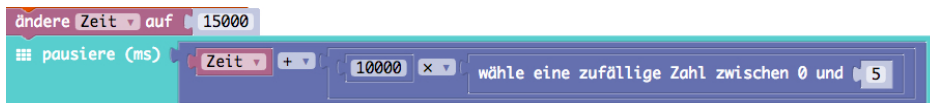


Abb. 6–65 Pausieren des Programm mit Hilfe einer zufälligen Zahl

## 6.6.8 Der letzte Schritt

Du hast es geschafft! Dein Programm hat immer eine unterschiedliche Spielzeit. Weißt du, was jetzt noch fehlt? Genau, es fehlt noch ein Signal, das das Ende der Zeit ankündigt, und ein Neustart des Programms für die nächste Spielrunde. Das Signal für das Ende der Spielzeit kannst du dir mit Hilfe der Programmierbefehle »spiele Note« aus der Gruppe »Musik« holen. Hier kannst du verschiedene Töne ausprobieren oder auch eine kurze Tonfolge komponieren. Zuletzt holst du dir aus der Gruppe »Fortgeschritten – Steuerung« den Programmierbefehl »zurücksetzen«, um das Programm nach jedem Durchgang neu zu starten.

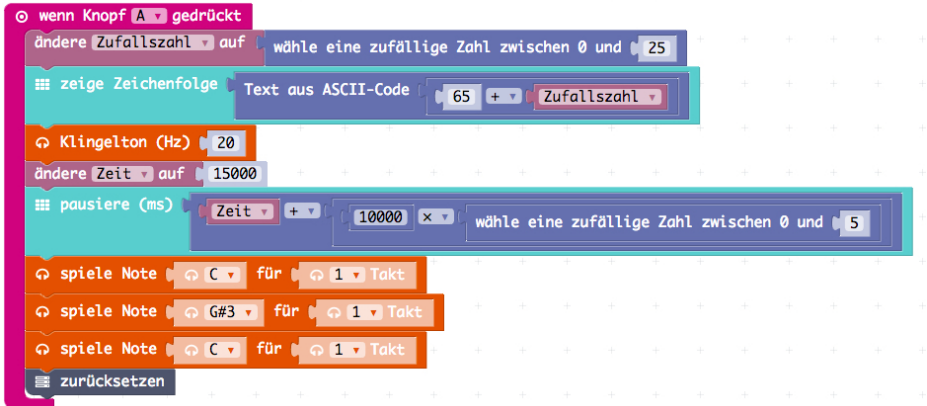


Abb. 6–66 Das komplette Spiel

### 6.6.9 Idee für den Schulunterricht

Dieses Spiel kann auch gut im Schulunterricht eingesetzt werden. Es gibt den Schülerinnen und Schülern einen Anreiz, verschiedene Wörter zu sagen und kreativ zu sein. Möglich ist es auch, das Spiel zu verändern, sodass der Buchstabe nicht am Anfang, sondern nur innerhalb des Wortes vorkommen muss. Es können auch nur bestimmte Themengebiete zugelassen sein oder man verwendet den Cal-liope-Buchstabengenerator für das beliebte Spiel Stadt-Land-Fluss.

# 7

## Es wird logisch ...

In diesem Kapitel lernst du, wie du deinem Calliope mini beibringst, zwischen Wahr und Falsch zu unterscheiden.

Die Einführung zu diesem Kapitel wurde von Jan Thar geschrieben. »Würfelglück« wurde von Nadine Bergner erstellt und »Tischbillard« von Julia Kleeberger.

Bisher sind die Programme in diesem Buch fast immer einfache Abfolgen gewesen, bei denen der Calliope mini immer die gleichen Bilder abspielt oder Messwerte der Sensoren direkt wiedergibt. Entscheidungen kannst du mit dem Calliope mini durch einfache »Wahr oder Falsch«-Fragen treffen, der Calliope mini kann dann je nach Antwort unterschiedliche Aktionen ausführen, was du ja schon in Abschnitt 3.1 kennengelernt hast. In diesem Kapitel lernst du, wie dein Calliope mini diese Entscheidungen trifft.

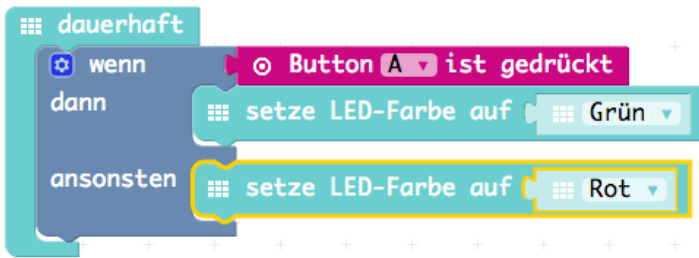


Abb. 7-1 Ist Knopf A gedrückt?

Ein einfaches Beispiel: Hier schaut dein Calliope mini nach, ob der Knopf (auf Englisch: Button) gedrückt ist. Ist das wahr, führt er die »dann«-Aktion aus. Wenn der Button nicht gedrückt ist (»Button A ist gedrückt« stimmt also dann nicht), dann folgt die »ansonsten«-Aktion. Das Schöne bei diesen Sachen ist, man kann sie ganz leicht miteinander verschachteln und umkehren: Möchtest du zum Beispiel, dass dein Calliope mini genau umgekehrt auf das Drücken des Buttons reagiert, könntest du hier noch recht einfach die LED-Farben austauschen. Es reicht aber auch, einen »Nicht«-Block vor den lilafarbenen Programmierbefehl zu setzen – was besonders, wenn du später kompliziertere Programme schreibst, interessant ist.

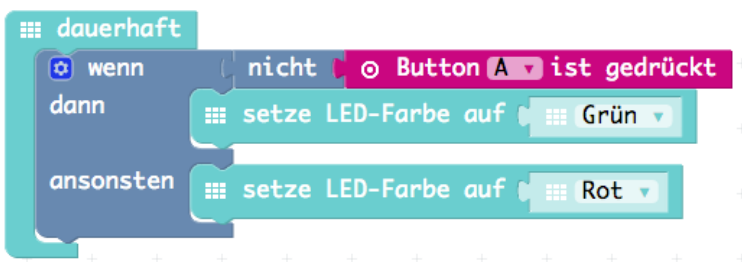


Abb. 7-2 Ist Knopf A nicht gedrückt?

Da »Nicht wahr« falsch ist und »nicht falsch« wahr, reagiert der Calliope mini nun auf den Knopfdruck genau umgekehrt: Drückst du den Button, sieht der Calliope mini, dass »Button A ist gedrückt« wahr ist. Da aber davor noch das »nicht« steht, wird in der Schleife nun der »ansonsten«-Zweig durchgeführt, da das, was nach »wenn« steht, zusammen falsch ist. Natürlich könntest du auch noch ein

weiteres »nicht« vor das »nicht« setzen (oder beliebig viele), aber vielleicht weißt du schon, was dann passieren wird. Als Tabelle:

Button A ist gedrückt	Ergebnis für nicht A
wahr	falsch
wahr	falsch
falsch	wahr
falsch	wahr

Tab. 7-1 Verneinung: Das Gegenteil wird gemacht

Zurück zum Originalprogramm ohne den »Nicht«-Befehl. Wie du schon aus Kapitel 3 weißt, kannst du auch auf das kleine Dreieck neben »A« klicken und »A+B« wählen (siehe Abschnitt 3.1). Wähle das einmal aus und lade es auf deinen Calliope mini.

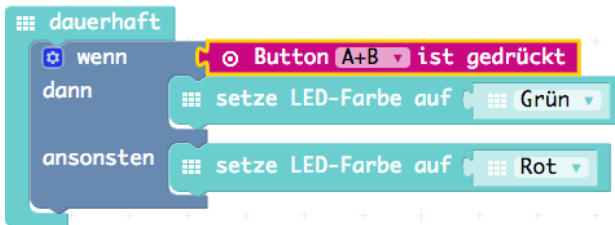


Abb. 7-3 Wird die LED grün oder rot?

Wann wird nun die LED grün? Nur wenn beide Buttons gleichzeitig gedrückt werden, das entspricht dem logischen »und«: Der Calliope mini betrachtet nun beide Buttons beziehungsweise Knöpfe, und nur wenn beide gedrückt sind (die Frage, ob ein Button gedrückt ist, kann also mit »wahr« für jeden der beiden Buttons beantwortet werden), ist auch die gesamte Frage, ob Button A und Button B gedrückt wurde, »wahr«. Ist entweder einer oder sind gar beide nicht gedrückt, ergibt sich ein »falsch«: Sowohl »falsch und falsch« wie auch »wahr und falsch« zusammen ergibt immer ein »falsch«. Das Ganze kann man auch etwas übersichtlicher als Tabelle schreiben:

Button A ist gedrückt	Button B ist gedrückt	Ergebnis für A und B
wahr	wahr	wahr
wahr	falsch	falsch
falsch	wahr	falsch
falsch	falsch	falsch

Tab. 7-2 Und: Zwei Ereignisse müssen gleichzeitig wahr sein

Der Editor erlaubt aber noch eine andere Schreibweise der gleichen Aussage:



Abb. 7-4 So geht's auch

Das ist zwar länger, aber der Vorteil dabei ist, dass du so auch ganz andere Fragen anstelle der Button-Abfrage einbauen kannst:

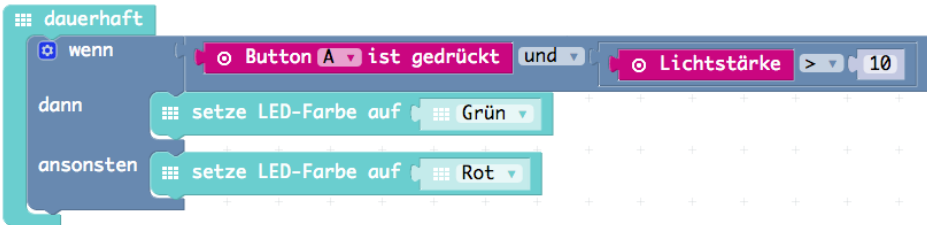


Abb. 7-5 Button wird mit Lichtstärke kombiniert

Zum Beispiel die Abfrage der Lichtstärke. Im Gegensatz zu »Button A ist gedrückt« liefert »Lichtstärke« allerdings keinen »wahr«- oder »falsch«-Wert, sondern stattdessen einen Zahl. Um den Wert in einen einfachen »wahr«- oder »falsch«-Wert zu überführen, kannst du einen Vergleich mit einem anderen Zahlenwert machen – hier fragst du also ab, ob der vom Lichtsensor gemessene Wert grösser (>) als 10 ist. Es gibt noch eine ganze Reihe anderer Vergleichsmöglichkeiten, aber alle entsprechen eigentlich den ganz normalen Vergleichen, die du auch im realen Leben so machen würdest: Eine Sache kann beispielsweise größer »>«, kleiner »<« oder gleich groß »=« sein. Die anderen Vergleiche sind nur dazu da, das noch etwas zu erweitern. »≠« heißt beispielsweise, dass beide Dinge nicht exakt die gleiche Größe haben, bei »≥« (>=) reicht es bereits, wenn ein Objekt gleich groß oder größer (kleiner) ist, um dies als »wahr« zu erfüllen.

Womit auch schon der letzte wichtige Punkt kommt: Statt »und« lassen sich beide Fragen hier auch durch ein »oder« dazwischen verbinden:

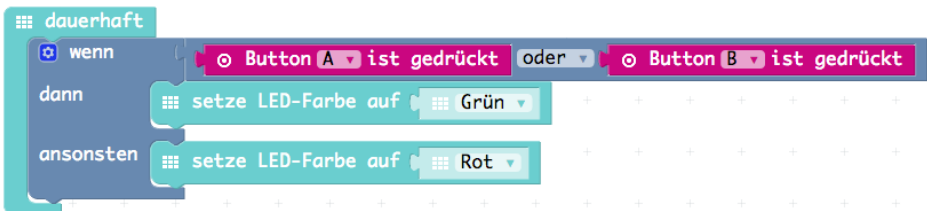


Abb. 7-6 Die Verbindung durch »oder«

Was passiert nun? Egal, welcher der beiden Buttons gedrückt wird, wird die LED grün leuchten. Nur wenn keiner gedrückt ist, bleibt sie rot. Für deinen Calliope mini ist also »wahr oder falsch« zusammen »wahr« (im Grunde sind Mikroprozessoren also sehr nett, sie nehmen im Zweifelsfall immer das Beste an). Auch hier wieder als Tabelle:

Button A ist gedrückt	Button B ist gedrückt	Ergebnis für A oder B
wahr	wahr	wahr
wahr	falsch	wahr
falsch	wahr	wahr
falsch	falsch	falsch

Tab. 7-3 Oder: Entweder das eine oder das andere muss wahr sein.

## Wofür nutzen wir Logik in der Programmierung?

Mit Logik fragen wir einfache »wahr oder falsch«-Fragen, entweder als direkte Fragen »Button gedrückt?« oder als Vergleiche »Wert 1 > Wert 2«, die wir durch »und« oder »oder« miteinander verknüpfen sowie durch »nicht« ins Gegenteil umkehren können.

Zum Abschluss eine kleine Rätselaufgabe: Bevor du das folgende Programm auf deinen Calliope mini lädst, kannst du bereits erkennen, mit welcher Farbe die LED leuchten wird? »wahr« und »falsch« sind dabei einfach nur Blöcke, die bereits einen entsprechenden Wahrheitswert haben. Hier werden sie zusammen mit den vorher vorgestellten Programmierblöcken zu einer großen Frage zusammengestellt.

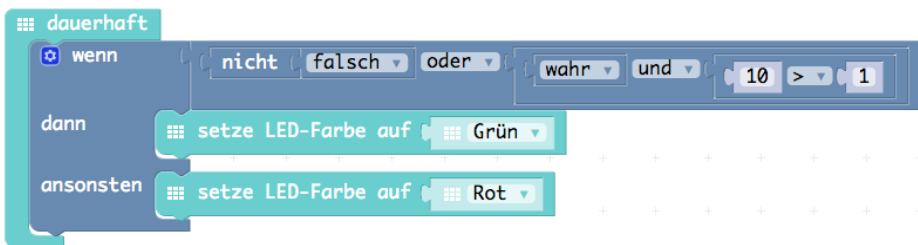


Abb. 7-7 Alles zusammengefasst

Na ja, auch ohne Herunterladen: Der Simulator verrät es direkt. »10>1« ist natürlich wahr, damit ergibt sich für den hinteren Teil, dass »wahr und wahr« auch wahr ist. Im vorderen Teil steht »nicht falsch«, was wahr ist, so dass zusammen »wahr oder wahr« auch wahr ist. Wie du hier sehen kannst: In manchen Fällen braucht man sich gar nicht so viele Gedanken zu machen: Sobald du weißt, dass der erste Teil wahr ist und der Rest mit einem »oder« damit verbunden ist, ist

nach der Tabelle für »oder« bereits klar, dass der gesamte Ausdruck dann wahr sein wird. Die Auswertung des alternativen Teils kannst du dir also sparen.

Falls man bei solch wild verschachtelten Fragen einmal den Überblick verliert: Man kann auch einfach Teilfrageblöcke der großen Frage in eine eigene »wenn-dann«-Schleife ziehen und einfach schauen, was da genau passiert. Wenn du noch etwas trainieren möchtest, bevor du dich an die spannenden Projekte in diesem Kapitel begibst: Versuch doch einfach einmal, ein paar möglichst komplexe Verschachtelungen aus ganz vielen »und«-, »oder«- und »nicht«-Fragen zusammenzustellen und schau, ob du auf die gleichen Antworten wie der Calliope mini kommst.

## 7.1 Würfelglück

Du kennst sie von Brettspielen. Manchmal freust du dich, manchmal ärgerst du dich. Richtig, die Rede ist von Würfeln! Die meisten Würfel haben sechs Seiten, manche auch mehr. Wenn du einen Würfel mit deinem Calliope mini nachbaust, hast du sogar die freie Wahl, wie viele Seiten er haben soll und welche Zahlen auf den Seiten stehen.



Abb. 7–8 20-seitige Würfel

### Ziel des Spiels »Würfelglück«

In diesem Kapitel erfährst du, wie du eine Grundstruktur für ein Würfelspiel bauen kannst. Dabei kannst du selbst wählen, wie viele Seiten der virtuelle Würfel hat, welche Zahlen auf den Seiten stehen sollen und wer das Spiel gewinnt.

Bei diesem ersten Beispiel baust du einen klassischen sechsseitigen Würfel nach, und es gewinnt ganz einfach derjenige, der die höhere Zahl würfelt. Das ist aber nur die Basis für deine weiteren Ideen, welche Spiele du mit deinem super flexiblen virtuellen Würfel gestalten möchtest.

Damit dieses Würfelspiel mit einem einzigen Calliope mini funktioniert, musst du dich später mit deinem Gegner beim Würfeln immer abwechseln. Die größte Herausforderung bei diesem Projekt ist es, nicht den Überblick zu verlieren, wer gerade dran ist.

## 7.1.1 Platzhalter steuert den Spielverlauf

Um den gesamten Spielverlauf zu kontrollieren, fügst du als Erstes einen Platzhalter ein, der den Spielverlauf regelt. Du könntest ihn zum Beispiel »Spielphase« nennen. Zu Beginn des Spiels soll der Platzhalter erst einmal die Zahl »0« speichern, dies kennzeichnet den Spielbeginn. Dann könnte eine Zeichenfolge erscheinen oder auch ein Ton erklingen, damit die Spieler wissen, dass das Spiel nun beginnt. Anschließend muss der Wert des Platzhalters auf »1« gesetzt werden, da nun der erste Spieler (Spieler A) startet. Das Programmstück, was beim Starten ausgeführt wird, könnte also zum Beispiel so aussehen.



Abb. 7–9 Beispiel für den Spielbeginn

Für die weiteren Phasen des Spiels sind hier folgende Zahlen im Platzhalter »Spielphase« angedacht:

- ▶ 0: Spielbeginn
- ▶ 1: Spieler A würfelt auf Knopf »A«
- ▶ 2: Spieler B würfelt auf Knopf »B«
- ▶ 3: Ergebnis des Würfelwurfs wird ausgegeben
- ▶ 4: Neustart des Spiels

## 7.1.2 Grundstruktur jeder Spielphase

Damit der Calliope mini das Spiel steuern kann, brauchst du eine Dauerschleife, die nacheinander die verschiedenen Phasen abläuft. Ein wichtiges Element dazu ist der »während-mache«-Befehl.

### »während-mache«-Befehl

Mit dem Befehl »während-mache« aus der Befehlsgruppe »Schleifen«, kannst du genau vorgeben, was der Calliope mini in jeder Spielphase (also zum Beispiel »während Spielphase = 3«) machen soll.

Du kannst nun vorne starten. Kombiniere dazu eine Dauerschleife mit dem »während-mache«-Befehl.



Abb. 7–10 Grundstruktur des Programms

## 7.1.3 Spielphase 1 – Spieler A würfelt

In der Bedingung hinter dem »während« musst du prüfen, ob das Spiel in der passenden Phase ist, also hier »Spielphase = 1«. Nun gibt es zwei verschiedene Möglichkeiten:

1. Spieler A würfelt
2. Spieler A würfelt noch nicht

In dieser Basisversion des Spiels geschieht das Würfeln einfach über das Drücken der Knöpfe. Dies lässt sich später problemlos verändern, zum Beispiel könnten die Spieler würfeln, indem sie den Calliope mini schütteln, oder ein Spieler kippt den Calliope mini nach rechts, der andere Spieler nach links. Wenn das Grundprogramm einmal steht, kannst du es vielfältig nach eigenen Wünschen frei verändern.



Abb. 7-11 Abfrage, ob Spieler würfelt in Phase 1

## 7.1.4 Das Würfeln

Falls der Spieler würfelt, müssen gleich mehrere Dinge passieren:

1. Es muss eine Zufallszahl berechnet und gespeichert werden.
2. Diese Zufallszahl soll auf dem LED-Display (für eine kurze Zeit) angezeigt werden.
3. Das Spiel wechselt in die nächste Phase.

Damit der Calliope mini die gewürfelte Zahl auch speichern kann, brauchst du als Erstes einen Platzhalter (zum Beispiel »WürfelwurfSpielerA«). In diesen Platzhalter kannst du dann die Zufallszahl speichern. Damit du eine Zufallszahl von 1 bis 6 (wie bei einem normalen Würfel) bekommst, musst du zu der Zufallszahl aus dem Befehl »wähle eine zufällige Zahl zwischen 0 und 5« noch eine »1« hinzu addieren.



Abb. 7-12 Berechnung einer Zufallszahl von 1 bis 6

Nun ist die Zufallszahl berechnet und auch abgespeichert, aber der Spieler möchte natürlich auch sehen, was er gewürfelt hat. Die Ausgabe des Ergebnisses im Platzhalter »WürfelwurfSpielerA« schaffst du doch sicher selbst. Danach solltest du auch eine kurze Pause einfügen, damit der Spieler Zeit hat, sein Ergebnis zu lesen, bevor das Spiel in die nächste Phase wechselt.

Und zum Abschluss nicht vergessen: Das Spiel wechselt nun in Spielphase 2.

Falls Spieler 1 noch nicht würfelt, könntest du den Buchstaben »A« anzeigen, damit der erste Spieler merkt, dass er an der Reihe ist. Alternativ könntest du auch einen Ton ausgeben oder die RGB-LED in der passenden Farbe leuchten lassen.

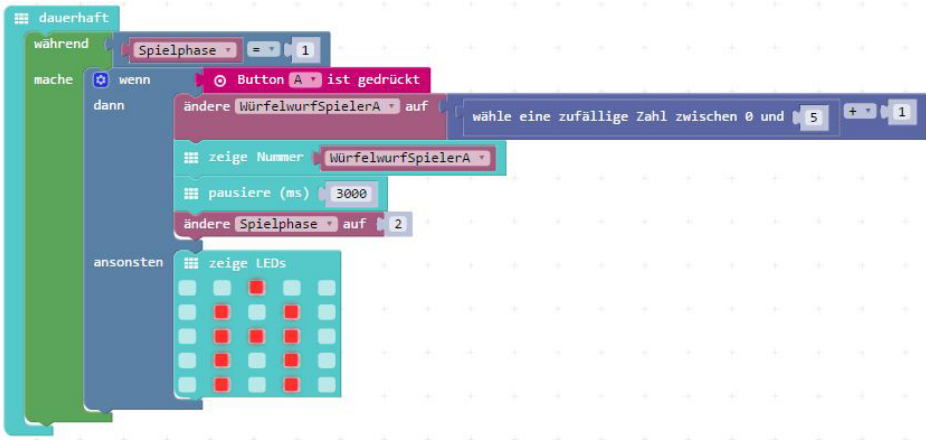


Abb. 7–13 Beispiel: das Würfeln in Phase 1

Nun hast du einen riesigen Schritt geschafft. Die Spielphase 1, in der der erste Spieler würfelt, ist geschafft. Du kannst es also schon ausprobieren.

- ▶ Wird dir dein Würfelergebnis angezeigt?
- ▶ Werden alle Zahlen von 1 bis 6 gewürfelt?
- ▶ Welche kommt bei 10-mal würfeln am häufigsten?
- ▶ Wie oft musst du würfeln, bis du eine Zahl dreimal würfelst?
- ▶ Kommt es vor, dass eine Zahl gleich zweimal nacheinander gewürfelt wird?

Um das Spiel neu zu starten, kannst du den »Reset«-Knopf verwenden.

### 7.1.5 Spielphase 2 – Spieler B würfelt

Nachdem Spieler A (in Spielphase 1) gewürfelt hat, ist nun Spieler B an der Reihe. Für die Spielphase 2 kannst du einiges aus Phase 1 übernehmen. Damit auch nichts schiefeht, hier noch einmal die wichtigsten Schritte im Überblick:

1. Es muss eine Zufallszahl von 1 bis 6 erstellt werden.
2. Diese Zufallszahl muss in einem weiteren (!) Platzhalter gespeichert werden.
3. Dann sollte das Würfelergebnis auch angezeigt werden.
4. Eine kleine Pause nicht vergessen, damit der Spieler Zeit hat, das Ergebnis anzuschauen.
5. Das Spiel wechselt in die nächste Phase.

Weiter sollte hier kein »A«, sondern ein »B« erscheinen, damit der zweite Spieler auch weiß, dass er nun würfeln darf.

Such dir nun deinen ersten Gegner. Testet das Würfelspiel zu zweit. Prüfe dabei Folgendes:

- ▶ Kann jeder Spieler zum richtigen Zeitpunkt würfeln? Und auch nur dann, wenn er dran ist?
- ▶ Wird bei beiden Spielern eine zufällige Zahl von 1 bis 6 angezeigt?
- ▶ Reicht die Anzeigzeit des Würfelergebnisses aus, dass beide Spieler das Ergebnis sehen?

Spielt am besten ein paar Runden, um zu prüfen, ob alles so funktioniert, wie ihr es erwartet. Allerdings müsst ihr hierbei eure Ergebnisse immer selbst im Kopf behalten und am Ende vergleichen, denn bisher fehlt ja noch die Spielphase 3, in der das Ergebnis angezeigt wird.

## 7.1.6 Das Ergebnis

Jetzt wird es spannend. Der Calliope mini soll das Ergebnis automatisch berechnen. Genau wie in den anderen Phasen, brauchst du auch hier wieder einen »während-mache«-Befehl, der sicherstellt, dass das Spiel nun in Spielphase 3 ist. Damit die Spieler wissen, dass nun der Sieger verkündet wird, könntest du:

- ▶ die Zeichenfolge »Gewinner« einblenden,
- ▶ die RGB-LED ein Farbmuster aus verschiedenen Farben zeigen lassen,
- ▶ ein kleines Lied abspielen
- ▶ oder was immer dir selbst gerade einfällt.

Jetzt kommt der Moment der Entscheidung. Der Calliope mini muss nun ausrechnen, wer gewonnen hat. Dabei gibt es drei Möglichkeiten:

1. Der Würfelwurf von Spieler A ist größer ( $>$ ) als der von Spieler B. → Spieler A hat gewonnen.
2. Der Würfelwurf von Spieler A ist kleiner ( $<$ ) als der von Spieler B. → Spieler B hat gewonnen.
3. Beide Spieler haben die gleiche Zahl gewürfelt.

Für alle drei Fälle brauchst du einen »wenn-dann«-Befehl mit der passenden Bedingung. Für den ersten Fall müsste er lauten: »wenn WürfelwurfSpielerA  $>$  WürfelwurfSpielerB, dann zeige Zeichenfolge »A««.



Abb. 7–14 Ergebnisausgabe für den Fall, dass Spieler A gewinnt

Genau nach diesem Beispiel kannst du auch die Ausgabe für den zweiten Fall (Spieler B gewinnt) programmieren.

Im dritten Fall kannst du dich entscheiden: Möchtest du lieber beide Spieler feiern, dann kannst du »beide« ausgeben lassen, sonst hat wohl »keiner« gewonnen.

Die Verkündung des Ergebnisses kannst du wieder nach Belieben mit Tönen oder Farben der RGB-LED ausschmücken. Anschließend solltest du daran denken, dass das Spiel noch in die letzte Phase (Nummer 4) wechselt.

## 7.1.7 Nach dem Spiel ist vor dem Spiel

Wenn nun der Sieger der ersten Runde lange genug gefeiert hat, willst du sicher die zweite Runde starten. Wenn du nun den »Reset«-Knopf drückst, beginnt das Spiel zwar von vorne, aber vielleicht möchtest du später noch die bisherigen Punkte speichern. Die sind bei einem Druck auf den »Reset«-Knopf allerdings immer weg, da das Programm ja komplett neu gestartet wird. Du brauchst also einen anderen Weg.

Das ist auch gar kein großes Problem, da du ja einen Platzhalter für die Spielphasen hast. Wenn du diesen Platzhalter wieder auf »1« setzt, startet das Spiel in die nächste Runde, und Spieler A beginnt wieder zu würfeln. Es gibt ganz verschiedene Ideen, wie man die zweite Runde starten könnte. Eine Möglichkeit wäre, dass beide Spieler gleichzeitig ihre Knöpfe drücken müssen. Dann ist auch sichergestellt, dass beide Spieler bereit für die nächste Runde sind.

Du könntest also in Spielphase 4 abfragen, ob Knopf »A« und »B« gedrückt ist, dann eine Zeichenfolge wie »Nächste Runde« anzeigen lassen und wieder in Spielphase 1 wechseln. Dann speicherst du in dem Platzhalter »Spielphase« die »1« und das Spiel beginnt von vorn.



Abb. 7-15 Neustart des Spiels in die nächste Runde

## Ausführungsgeschwindigkeit

Beachte, dass der Calliope mini alle Befehle super schnell nacheinander abarbeitet. Falls du also den Text »Nächste Runde« nicht anzeigen lässt und stattdessen direkt in Spielphase 1 wechselst, wird sofort der Würfelwurf für Spieler A angezeigt, da ja Knopf »A« bereits gedrückt wird. Du musst den Spielern also Zeit geben, die Knöpfe wieder loszulassen, bevor Spielphase 1 startet.

Jetzt wird es Zeit, dein Ergebnis zu erproben. Such dir einen Gegner und startet gemeinsam das Würfelspiel.

- ▶ Wer schafft es als Erster, 5-mal zu gewinnen?
- ▶ Wie oft müsst ihr würfeln, um 3-mal einen Gleichstand zu erzeugen?
- ▶ Schafft es einer von euch, gleich drei Siege mehr zu erlangen als der andere?

## 7.1.8 Erweiterungsmöglichkeiten

Du hast das gesamte Würfelspiel einmal von Beginn bis Ende programmiert, und es funktioniert. Klasse!

- ▶ Dein Spiel würfelt auf Knopfdruck automatisch, erst für Spieler A und dann für Spieler B.
- ▶ Die Ergebnisse werden verglichen.
- ▶ Der Gewinner wird bekanntgegeben.
- ▶ Das Spiel kann in die nächste Runde starten.

Wenn du Lust hast, kannst du nun dein Würfelspiel erweitern, verändern und umgestalten. Hier ein paar Ideen, wie das Spiel noch interessanter werden könnte:

- ▶ Du könntest den Bereich der Zufallszahl erweitern. Warum nur Zahlen zwischen 1 und 6 würfeln? Wie wäre es mit 1 bis 10, 1 bis 20 oder sogar 1 bis 100?
- ▶ Aktuell würfeln die Spieler durch einen Druck auf Knopf »A« und B. Aber das muss nicht sein. Du könntest den Calliope mini so programmieren, dass eine neue Zufallszahl erscheint, wenn der Spieler ihn auf den Kopf dreht. Dazu musst du hinter das »wenn« die Bedingung »Beschleunigung (mg)  $z < 0$ « setzen, da dies bedeutet, dass der Calliope mini mit dem LED-Display nach unten zeigt.

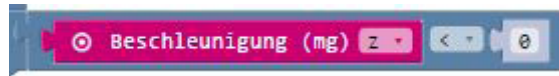


Abb. 7–16 Der Calliope mini zeigt mit dem LED-Display nach unten.

- ▶ Spannend wäre auch, wenn der Calliope mini selbst die Punkte aus mehreren Runden zusammenzählt. Dazu brauchst du Platzhalter für beide Spieler, zum Beispiel »PunkteSpielerA« und »PunkteSpielerB«. In Phase 3 bei der Auswertung der aktuellen Runde musst du dann zusätzlich den richtigen Platzhalter, also »PunkteSpielerA« oder »PunkteSpielerB« um »1« erhöhen. Bei einem Gleichstand musst du entscheiden, ob beide einen Punkt bekommen sollen oder lieber niemand. Beim Ende der aktuellen Runde, also in Spielphase 4, kannst du dann den Spielstand anzeigen lassen. Dazu zeigst du zuerst eine Zeichenfolge an, die den Spieler nennt, anschließend den Wert des Platzhalters. So erfahren beide Spieler nacheinander, wie viele Punkte sie aktuell haben.

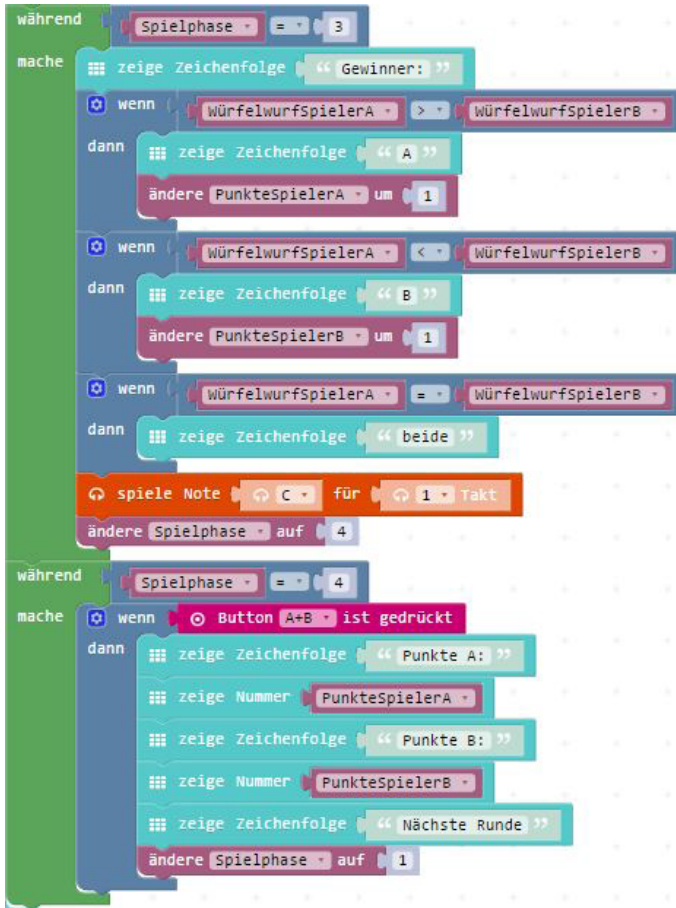


Abb. 7-17 Erweiterung um den Punktestand

- ▶ Falls dir die Anzeige des Spielstands nach jeder Runde zu lange dauert, kannst du auch ein neues Programmstück einfügen, damit der Punktestand immer dann angezeigt wird, wenn der Calliope mini geschüttelt wird. So können die Spieler selbst entscheiden, wann sie den Punktestand sehen wollen.



Abb. 7-18 Spielstand wird bei Schütteln angezeigt.

Ganz bestimmt hast du selbst noch viele weitere Ideen, wie du dein Würfelspiel erweitern könntest. Gerne kannst du auch Ideen aus anderen Projekten einbauen. Ganz viel Würfelglück dabei!

### 7.1.9 Unfares Spiel

Dieser Abschnitt ist nur etwas für Betrügerinnen und Betrüger. Vielleicht bist du ja sogar selbst schon auf die Idee gekommen. Mit einem solchen automatischen Würfel kann man auch ganz einfach betrügen. Dazu brauchst du nur die Berechnung der Zufallszahl bei Spieler A zu verändern. Zum Beispiel könntest du den Calliope mini so programmieren, dass bei Spieler A nur Zahlen zwischen 3 und 6 angezeigt werden.

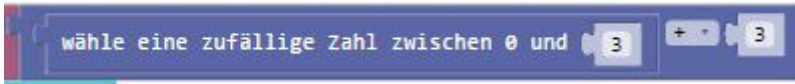


Abb. 7-19 Unfaire Berechnung der eigenen Zufallszahl von 3 bis 6

Wenn du nun dafür sorgst, dass du selbst immer Spieler A bist, hast du viel höhere Chancen zu gewinnen, da bei dir nur die Zahlen 3 bis 6 kommen können, bei deinem Gegner aber weiterhin die Zahlen 1 bis 6.

Wenn du noch gemeiner sein möchtest, kannst du auch die Berechnung der Zufallszahl deines Gegners noch manipulieren, sodass er nur die Ziffern 1 bis 4 würfeln kann

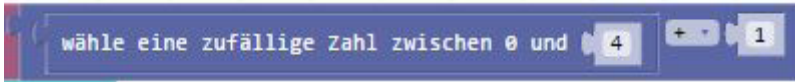


Abb. 7-20 Unfaire Berechnung der gegnerischen Zufallszahl von 1 bis 4

### Schummeln

Aber aufgepasst, wenn das Spiel zu unfair ist, wird dein Gegner Verdacht schöpfen und dann ganz sicher nie wieder mit dir spielen.

## 7.2 Tischbillard

Bei dieser Variante des Tischbillards geht es darum, die Spielkugel durch Schnippen in die Löcher zu bekommen, die sich auf dem Spielfeld befinden. Dein Calliope mini ermittelt dabei, in welches Loch die Kugel gefallen ist, und je nach Loch gibt es Punkte, die auf dem Display angezeigt werden.

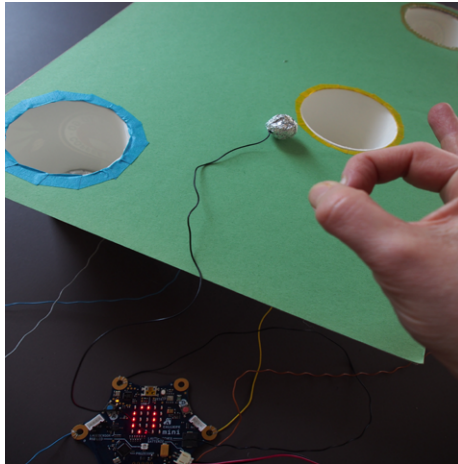


Abb. 7-21 Das fertige Tischbillard mit dem Calliope mini als Punktezähler

### 7.2.1 Materialien und Werkzeuge

Wie viele?	Was	Anmerkung
1	Calliope mini	
1	Battery-Pack	
2	Batterien (AAA)	Um sicherzugehen, dass du die richtigen Batterien verwendest, schau auf dein Battery-Pack
1	USB-Kabel (Mini)	Zum Anschließen des Minis an den Computer
1	Pappe (A3 oder größer)	Das wird dein Spielbrett
4	Pappbecher	
1 bis 4	Kreppklebeband/Washitape	Ggf. in 4 verschiedenen Farben
	Litze	Ggf. in 5 verschiedenen Farben
1	Ggf. Entmanteler	
1	Schere	

Tab. 7-4 Die Materialienliste

## 7.2.2 Wie funktioniert Tischbillard?

Um das Tischbillard umzusetzen, ist es notwendig, dass du den Calliope mini programmierst und das Spielfeld aufbaust. Für die Programmierung werden Platzhalter und Logikverknüpfungen benötigt. Eine grundlegende Einführung in Platzhalter bekommst du zu Beginn von Kapitel 5. Die allgemeine Funktionsweise von Logikverknüpfungen wird in zu Anfang dieses Kapitels erklärt. Das Tischbillard-Projekt eignet sich gut dafür, beides besser zu verstehen und richtig einzusetzen.

Um zu überprüfen, ob die Billardkugel in eines der Löcher gefallen ist, werden die Pins als Eingabesensoren genutzt. Sie liefern immer dann einen Wert, wenn der Stromkreis über die Pins geschlossen wird. In diesem Projekt ist das der Fall, wenn die Billardkugel in eines der Löcher gefallen ist. Wenn du noch einmal schauen willst, wie das mit dem Stromkreis und den Pins genau funktioniert, dann schau noch einmal in Kapitel 3.

In diesem Projekt lernst du, wie du ein Tischbillard aufbauen kannst. Aber es gibt natürlich auch jede Menge andere Spiele, die du umsetzen kannst. Überleg dir zu Beginn, welches Spiel du aufbauen willst.

### Und was ist mit Mini-Basketball?

Anstelle des Tischbillards kannst du dir eine Abwandlung des Spiels überlegen und zum Beispiel auch ein Tischbasketball aufbauen, ein Tischfußball oder ein Minigolf. Überleg dir, welches Spiel du gerne bauen willst. Deiner Fantasie sind keine Grenzen gesetzt.

Als Nächstes gilt es, die Regeln für das Spiel festzulegen. Das Tischbillard in diesem Projekt besteht aus einem Spielfeld mit vier Löchern, wobei jeder Treffer in ein Loch unterschiedlich gewertet wird: Je nachdem, in welches Loch die Spielkugel fällt, gibt es entweder einen Punkt, zwei Punkte, drei Punkte oder auch zwei Punkte Abzug.

## 7.2.3 Die Programmierung

Bevor du mit dem Projekt startest:

- ▶ Öffne mit einem Internetbrowser den Editor <https://mini.pxt.io/>
- ▶ Erstelle ein neues Projekt und nenne es z.B. »Tischbillard«.
- ▶ Schließ den Mini mit dem USB-Kabel an den Computer an.
- ▶ Eigne dir Vorwissen zu Platzhaltern (Kapitel 5), Logikverknüpfungen (Abschnitt 7.1) und Funktionsweise von Pins und Stromkreisen (Kapitel 3) an.

Beim Tischbillard prüft der Calliope mini, in welches Loch die Spielkugel gefallen ist, und zählt die entsprechende Punktezahl zum Spielstand dazu bzw. zieht die Punkte ab. Der aktuelle Punktestand wird auf dem Display angezeigt. Um dies zu programmieren, benötigst du einen Platzhalter, in dem der aktuelle Punktestand gespeichert wird. Gib dem Platzhalter einen Namen, der für dich eindeutig ist, z.B. »punkte«.

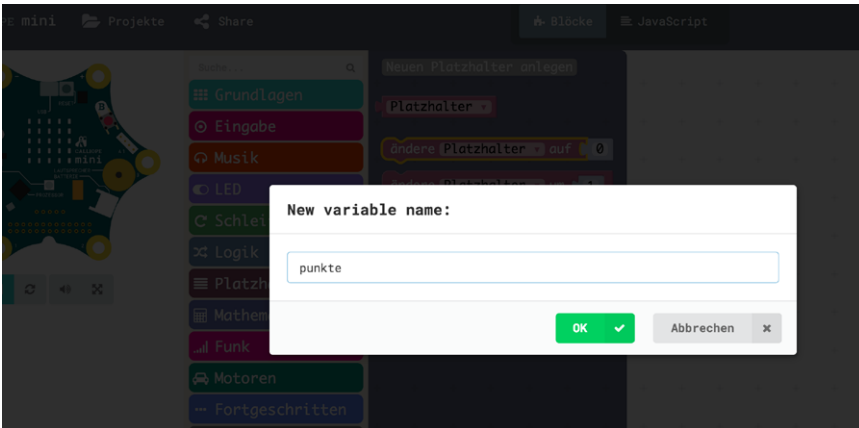


Abb. 7–22 Erstellen des neuen Platzhalters »punkte«

Zu Beginn des Spiels wird der Platzhalter »punkte« auf 0 gesetzt und im Display angezeigt:

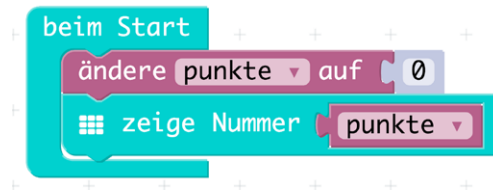


Abb. 7–23 Der Codeblock setzt den Platzhalter »punkte« zu Beginn jedes Spiels auf 0

Jetzt wird die Punktevergabe für die jeweiligen Pins festgelegt:

Pin	Punktzahl
Pin 0	1 Punkt
Pin 1	2 Punkte
Pin 2	3 Punkte
Pin 3	- 2 Punkte

Tab. 7–5 Punktevergabe

Im Editor benötigst du jetzt den Logikblock »wenn – dann« um die Zuordnung der Pins zu den Punktezahlen umzusetzen. Der Code sieht dann so aus.

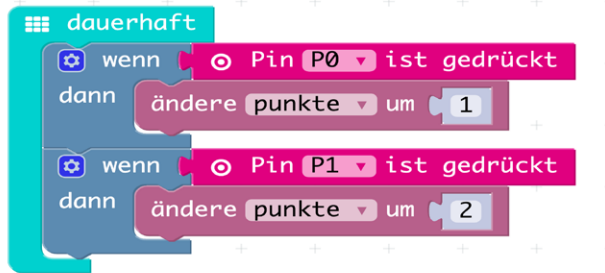


Abb. 7–24 Mit diesem Code wird die Zuweisung der Punktezahl zum jeweiligen Pin festgelegt

Der Minus-Pin wird nicht explizit programmiert. Er ist vielmehr Voraussetzung für die Funktionsweise der Pins: Nur wenn der Minus-Pin und ein weiterer Pin gleichzeitig kontaktiert werden, ist der Stromkreis geschlossen und ein Signal wird übermittelt.

## Pin-Belegung bei der Programmierung testen

Die Funktionsweise der Pin-Belegung lässt sich nicht im Simulationsmodus des Browsers testen. Hierfür musst du das Programm immer auf den Calliope mini überspielen. Um dir die Arbeit zu erleichtern, kannst du zum Testen des Programmcodes im Simulator z. B. den Funktionsblock wählen »Knopf A ist gedrückt«. Wenn du mit der Funktionsweise zufrieden bist, tauschst du zum Schluss den Eingabeblock wieder mit »Pin 0 ist gedrückt« aus, um es dann auf deinen Mini zu überspielen.

Bisher ist auf dem Calliope mini aber noch nichts zu sehen. Um den Punktstand auf dem Display anzuzeigen, brauchst du den Programmierblock »Zeige Nummer«, in den du den Platzhalter einfügst, damit der aktuelle Punktstand dargestellt wird. Zudem siehst du, dass eine Verzögerung von 5 Sekunden (also 5000 Millisekunden) im Code eingefügt wurde. Das ist die Zeit, die der Mini wartet, bevor er erneut Punkte von diesem Pin, also diesem Loch zählt. Das ist also die Zeit, die du zur Verfügung hast, um die Spielkugel aus dem Loch zu holen. Die Zeitdauer ist frei gewählt, und du kannst sie nach eigenem Ermessen anpassen.



Abb. 7–25 Erweiterung des Programmiercodes um die Anzeige der Punkte und einer Pause zum Herausholen der Spielkugel aus dem jeweiligen Spielloch

Diese Programmierblöcke gilt es nun für die anderen drei Pins zu ergänzen. Der vollständige Code innerhalb der Dauerschleife sieht so aus:

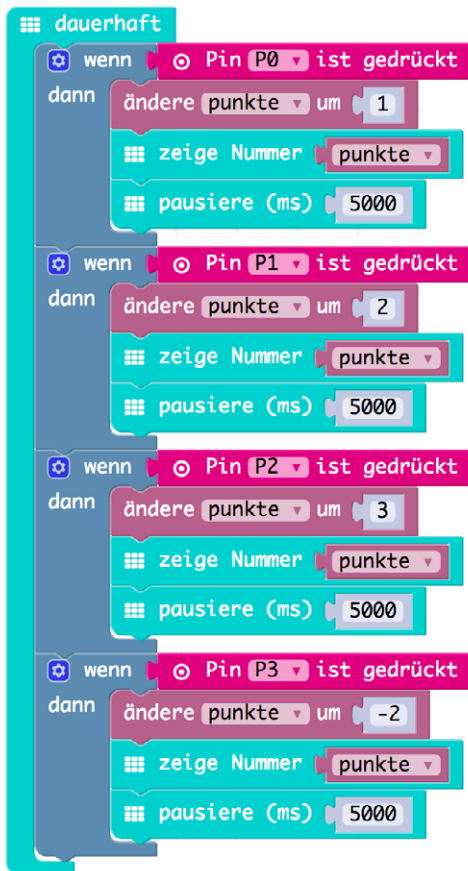


Abb. 7–26 Der vollständige Programmcode für die Dauerschleife

Um das Spiel jederzeit wieder von Neuem starten lassen zu können, kannst du noch eine Funktion auf Knopf A hinzufügen, die den Punktestand auf 0 setzt, wenn der Knopf gedrückt wird.

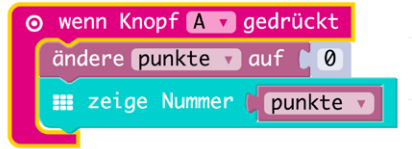


Abb. 7–27 Durch das Drücken von Knopf A wird der Punktestand auf 0 gesetzt und das Spiel startet von Neuem

Damit ist die Programmierung für den Mini abgeschlossen. Diesen Code kannst du jetzt auf dein Calliope-Board laden und mit dem Aufbau des Spielfeldes beginnen.

### 7.2.4 Der Aufbau des Spiels

Für den Aufbau des Tischbillards beginnst du mit der Grundfläche (A3-Pappe) und überlegst zunächst, wo sich die Spielfeldlöcher befinden sollen.

#### Spielgeschick ist gefordert

Um den Schwierigkeitsgrad etwas zu erhöhen, kannst du das Spielfeldloch mit der höchsten Punktzahl so positionieren, dass es sich hinter dem Loch mit dem Punktabzug befindet. Welche Möglichkeiten fallen dir noch ein, um Hindernisse oder Schwierigkeiten in das Spielfeld einzubauen, die dein spielerisches Geschick erfordern?



Beim Zeichnen und Ausschneiden der Löcher ist es wichtig, darauf zu achten, den Durchmesser etwas kleiner als die Becher zu zeichnen, so dass sie gut aufliegen und die Becher nicht durchrutschen.

Abb. 7–28 Der Durchmesser der Löcher, die du ausschneidest, muss etwas kleiner als der Durchmesser der Becher sein.

Um die vier Löcher besser auseinanderhalten zu können, kannst du sie farbig markieren. Hierfür eignet sich Kreppband bzw. Washitape in verschiedenen Farben.

Pin	Punktzahl	Farbzuordnung
Pin 0	1 Punkt	blau
Pin 1	2 Punkte	grau
Pin 2	3 Punkte	goldbraun
Pin 3	- 2 Punkte	gelb
Minus-Pin		schwarz

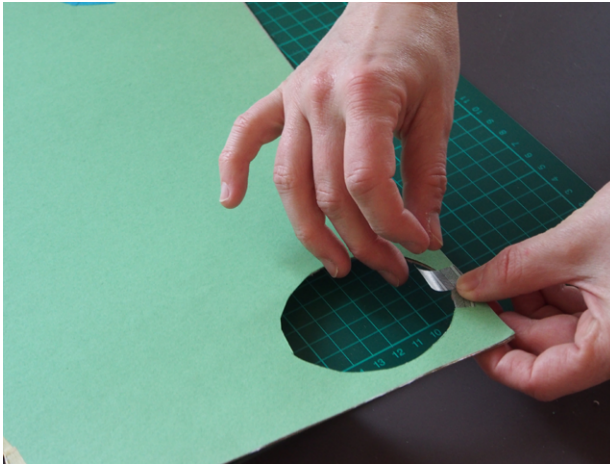


Abb. 7–29 Farbige Markierung der Spielfeldlöcher mit Washitape

Nun gilt es, den Calliope mini in das Spiel zu integrieren. Hierfür verbindest du die einzelnen Pins des Boards mit der Litze, wofür du sie zunächst entmanteln musst.

## Was ist Litze und warum muss man sie entmanteln?

Unter »Litze« versteht man dünnen Draht, der ggf. eine Isolierung besitzt, d. h., er ist von einer farbigen Schicht aus Kunststoff umschlossen: dem Mantel. Dies ist sinnvoll, damit nicht unabsichtlich ein Kontakt entsteht, falls sich die einzelnen Litzen in deinem Spielfeldaufbau zufällig berühren. Damit aber eine Verbindung zwischen dem Pin des Minis und der Litze hergestellt werden kann, muss die Isolierung entfernt, also die Litze entmantelt werden. Am besten eignet sich hierfür ein Entmanteler. Solltest du keinen zur Hand haben, kannst du hierfür je nach Festigkeit des Kunststoffes auch ein kleines Messer oder die Schere nehmen und damit den Mantel vorsichtig einritzen, sodass du ihn dann abziehen kannst. Hierbei musst du aufpassen, dass du nicht den Draht mit erwischst und komplett durchtrennst. Manchmal lässt sich der Mantel der Litze auch leicht mit dem Fingernagel entfernen.

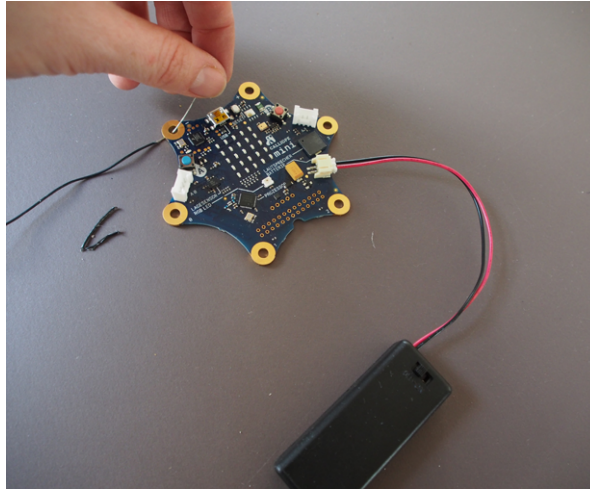


Abb. 7–30 Die Litze mit einem freien, entmantelten Ende, das mit dem Minus-Pin des Minis verbunden wird.

Dabei kannst du dieselbe Farbcodierung verwenden wie bei den Torlöchern (also blaue Litze für Pin 0, schwarze Litze für den Minus-Pin usw.).

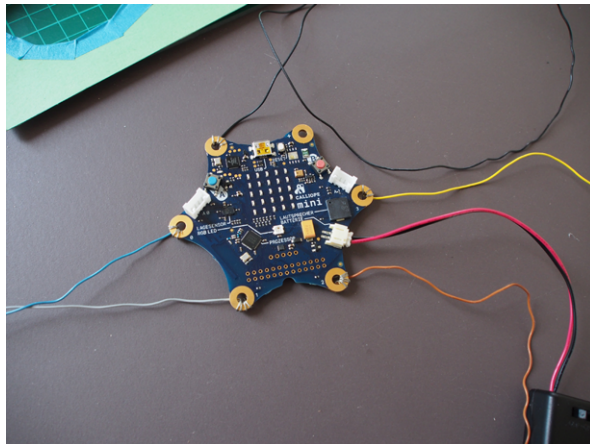


Abb. 7–31 An den Pins des Minis wird jetzt die Litze befestigt. Die verschiedenen Farben der Litze helfen dabei, sie gut auseinanderzuhalten.

Das zweite Ende der Litze verbindest du mit dem jeweils dazugehörigen farbigen Spielfeldloch (also blaue Litze mit blauem Loch usw.). Hierfür stichst du ein kleines Loch in den Unterboden des Tor-Bechers, um die Litze hindurchzufädeln. Damit der Draht nicht gleich wieder herausrutscht, klebst du den Draht mit etwas Kreppband am Becherboden fest.

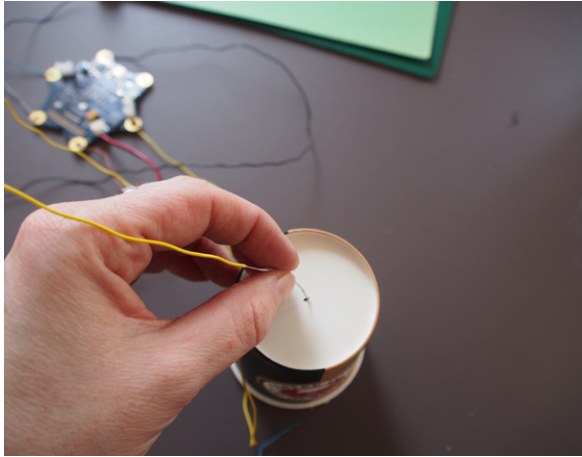


Abb. 7–32 Das andere Ende der Litze wird durch den Becher gefädelt.

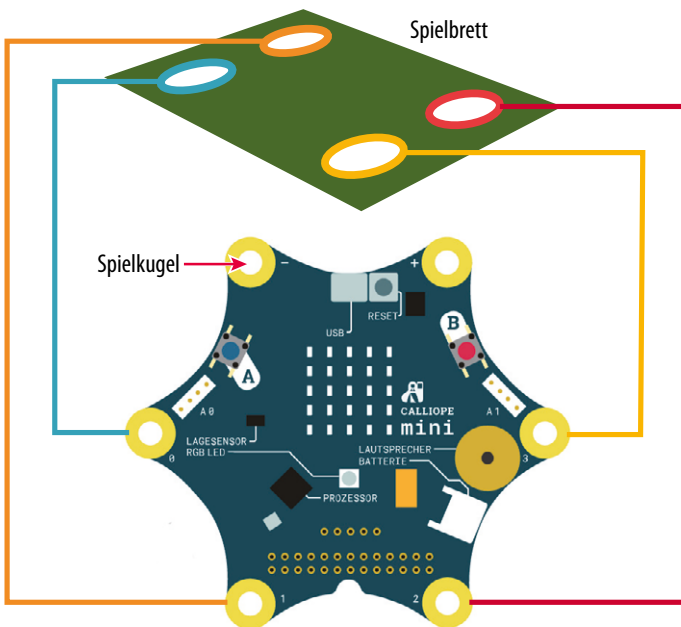


Abb. 7–33 Schematische Darstellung der Verbindungen

Jetzt vergrößerst du die Kontaktfläche des Tor-Bechers auf der Innenseite, indem du ein Stück Alufolie auf dem Becherboden anbringst. Damit die Alufolie besser hält, kannst du sie mit doppelseitigem Klebeband anheften. Achte dabei darauf, dass sich der Draht über dem Klebestreifen befindet und nicht darunter, damit ein guter Kontakt zwischen Alufolie und Litze bestehen bleibt. Diesen Schritt wiederholst du jetzt auch für die anderen drei Tor-Becher.

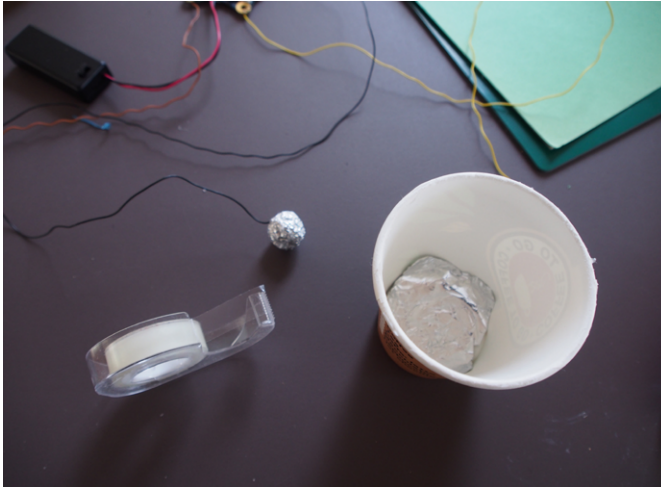


Abb. 7-34 Auf den Innenseiten der Tor-Becher wird Alufolie angebracht.

Das zweite Ende der schwarzen Litze wird nicht mit einem Tor-Becher verbunden, sondern mit der Spielkugel. Hierfür formst du eine Kugel aus Alufolie, die mit der Litze verbunden sein muss. Am besten klappt das, wenn du etwas Alufolie um die Litze wickelst und dann immer mehr Folie herumwickelst, bis ein Ball entsteht. So ist deine Litze rundum gut mit Alufolie umschlossen. Wenn du nun die Spielkugel in die Becher hältst, zeigt der Mini dir die Punkte an. Teste alle Spielfeldbecher, ob sie die gewünschte Punktzahl anzeigen.

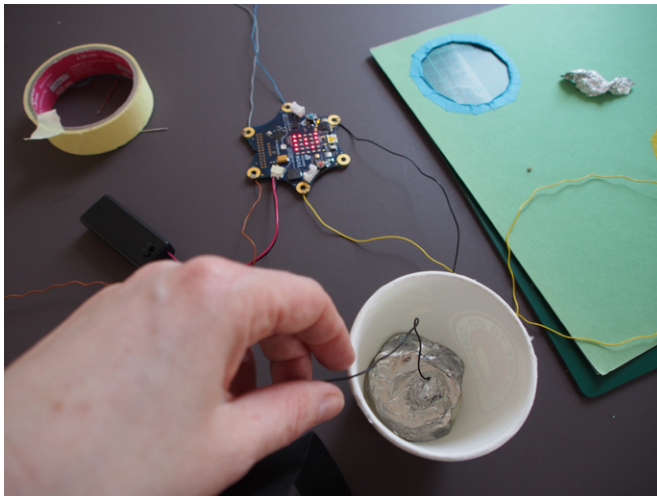


Abb. 7-35 Beim Berühren der Spielkugel mit dem Becherinnenboden werden auf dem Calliope mini die Punkte angezeigt.

Zum Schluss klebst du noch die Becher unter dem Spielfeld an. Jetzt ist dein Aufbau fertig und das Spielen kann beginnen.

## 7.2.5 Erweiterung des Programms

In diesem Beispiel ist das Tischbillard so aufgebaut, dass es vier Tore mit unterschiedlicher Punktzahl und eine Spielkugel gibt. Wenn du magst, kannst du auch noch mehr Spielkugeln ins Spiel bringen. Dafür verbindest du einfach jede zusätzliche Spielkugel ebenfalls mit dem Minus-Pin.

Alternativ kannst du das Spiel auch so aufbauen, dass die Kugeln unterschiedliche Punktezahlen bedeuten. Hierfür verbindest du die Spielkugeln mit den Pins 0–3 und die alle Tore mit dem Minus-Pin.

Fallen dir noch weitere Spielideen ein? Eventuell willst du auch gern eine Minigolf-Anlage nachbauen oder lieber ein Tischbasketball-Spiel? Probier es einfach mal aus!

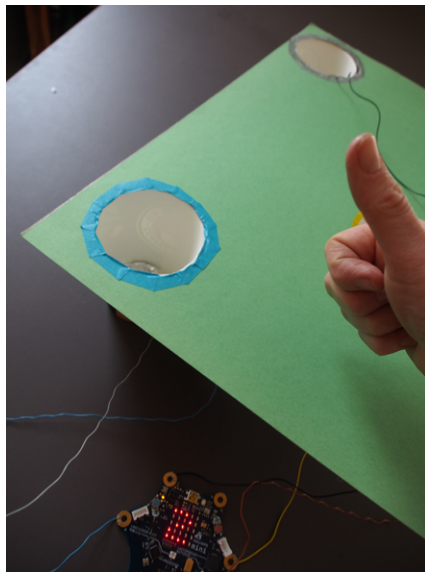
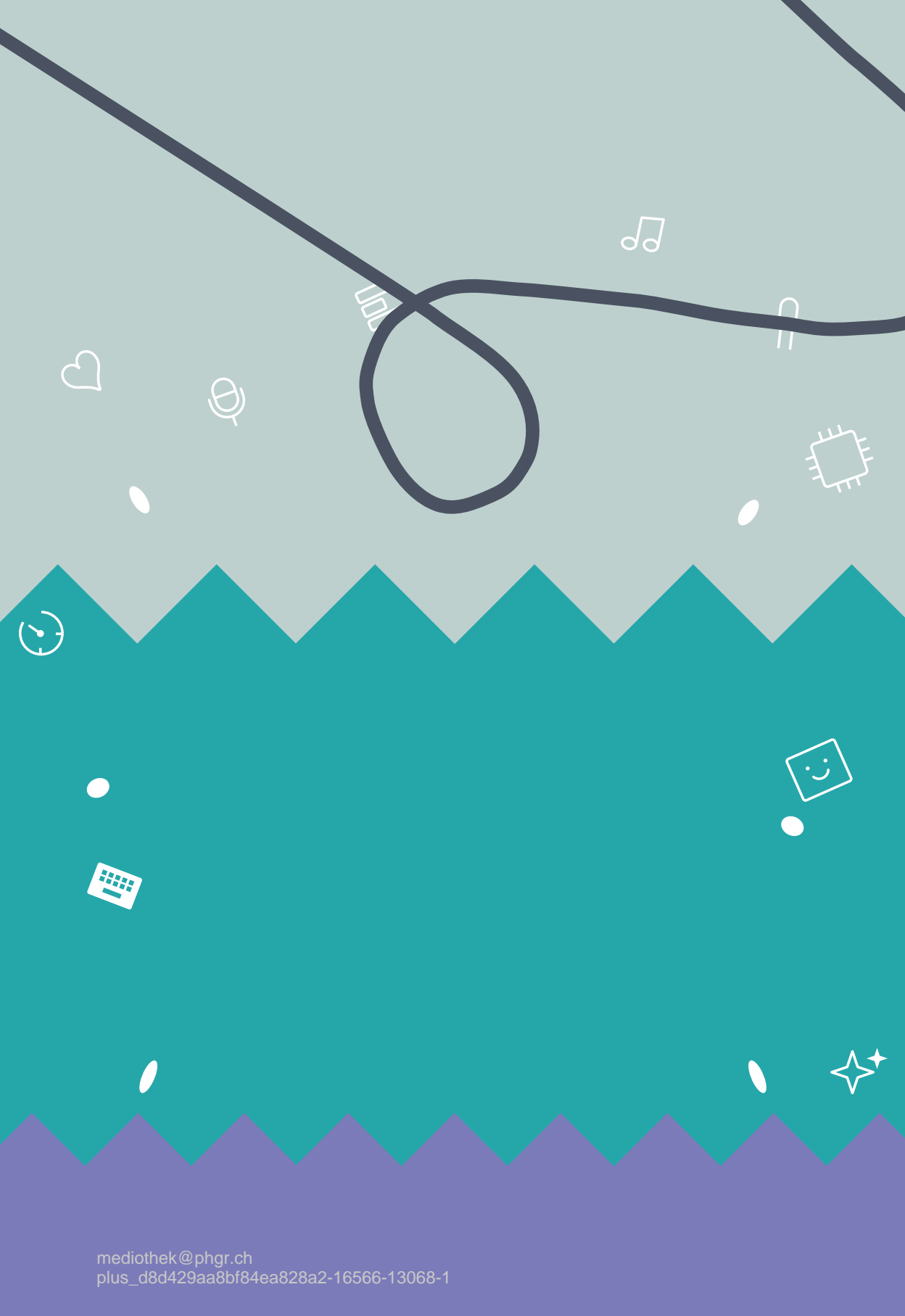


Abb. 7–36 Viel Spaß beim Spielen!



# 8

## Mapping – so hinein und anders wieder hinaus

In diesem Kapitel geht es um Mapping. Das ist ein wichtiges Thema, wenn du Projekte mit Sensoren und Aktoren umsetzen willst.

Die Einführung zu diesem Kapitel stammt von Thiemo Leonhardt, ebenso wie die Projekte »Farbverlauf«, »Luftgitarre« und »Fang den Dot«.

## 8.1 Mapping verstehen

In diesem Kapitel wirst du das Informatik-Prinzip »Mapping« kennenlernen, das es dir ermöglicht, noch mehr tolle Projekte mit den Sensoren und Aktoren des Calliope mini umzusetzen. Auf den folgenden Seiten warten folgende drei Projekte zum Thema »Mapping« auf dich: der Farbverlauf (eigene Lichterketten programmieren), die Luftgitarre mit echten Tönen (eigene Instrumente herstellen) und das Spiel »Fang den Dot« (eigene Spiele verbessern).

### 8.1.1 Worum geht es hier?

Schön, dass du dich mit Mapping beschäftigen möchtest. Mapping ist ein sehr wichtiges Thema, wenn du Projekte mit Sensoren und Aktoren umsetzen willst. Wie du in den vorangegangenen Kapiteln und Projekten, z. B. beim Farbthermometer in Abschnitt 4.4, schon gesehen hast, kannst du eine Eingabe wie die Temperaturdaten des Temperatursensors durch ein eigenes Programm in eine Ausgabe wie zum Beispiel eine Farbe umleiten.

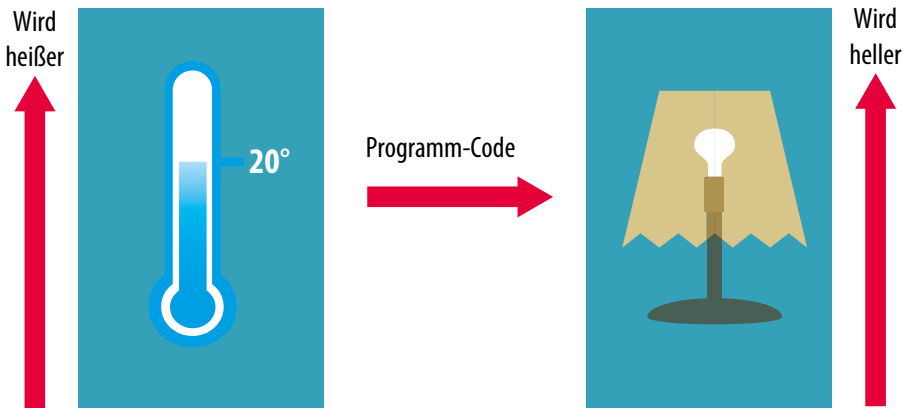


Abb. 8-1 Eingabe eines Sensorwertes umwandeln in die Ausgabe einer Lampe

Auf diese Weise kannst du zum Beispiel jemanden warnen, wenn eine Herdplatte so heiß wird, dass es für unsere Haut gefährlich wird (ab ca. 45 Grad), indem dann eine Warnlampe angeht. Damit du weißt, wie heiß es ist, kannst du den Block »Temperatur« benutzen.

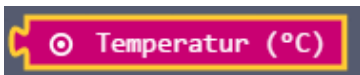


Abb. 8-2 »Temperatur«-Block

Die Frage ist jetzt aber, wie du die eine Lampe nicht nur bei einem bestimmten Temperaturwert aufleuchten lässt, sondern zum Beispiel die Helligkeit der roten Lampe so steuerst, dass die rote Lampe umso heller leuchtet, je höher die Temperatur ist. Schau dir dazu einmal dieses vereinfachte Beispiel an.

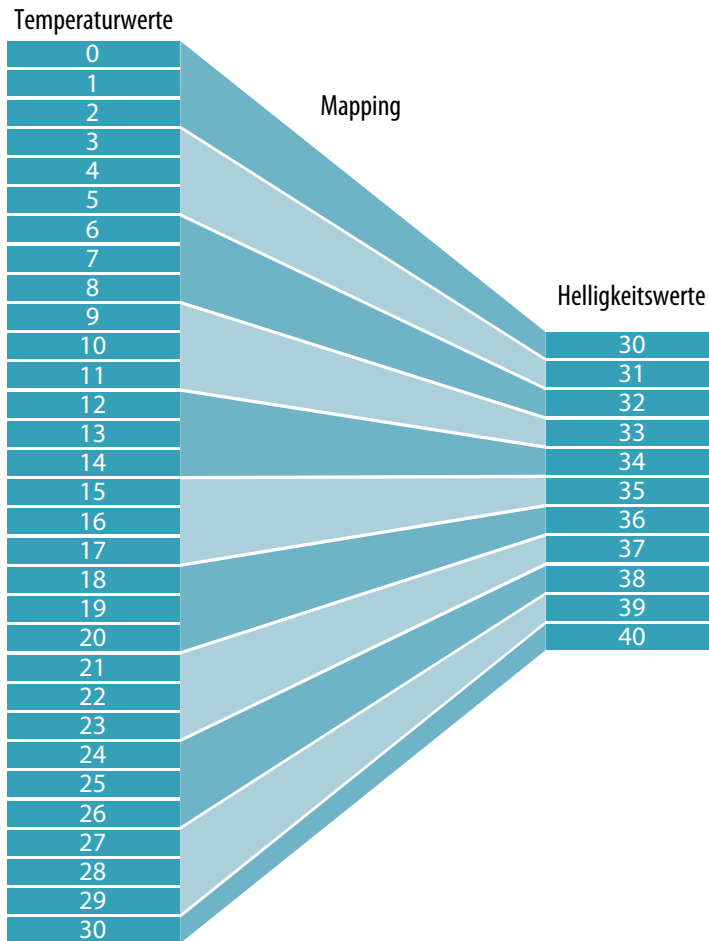


Abb. 8-3 Beispiel des Mappings der Sensorwerte von 0 bis 30 auf Temperaturwerte von 30 bis 40

Der Temperatursensor gibt dir Zahlenwerte von 0 bis 30 als Eingabe in deinen Platzhalter. Der Hersteller der Lampe sagt dir, dass 30 die kleinste Helligkeit und 40 die größte Helligkeit ist, mit der die Lampe leuchten kann.

Dein Temperatursensor gibt dir im Programm eine 15 aus. Die 15 halbiert genau den Abstand zwischen der 0 und der 30. Damit du nun den entsprechenden Temperaturwert herausbekommst, halbiert du den Abstand zwischen 30 und 40 der Temperaturwerte. Den Abstand berechnest du indem du von 40 die 30 abziehst.

## Rechentipp

Abstand = großer Wert minus kleiner Wert

Damit du die passende Helligkeit herausbekommst, musst du den Abstand der Helligkeitswerte der Lampe halbieren – genauso, wie du es auch bei den Zahlenwerten des Temperatursensors gemacht hast.

Du rechnest also 40 minus 30. Das ergibt 10. Dann teilst du 10 durch 2, um den Abstand zu halbieren. Dies ergibt 5. Das bedeutet jetzt aber noch nicht, dass die 15 als Zahlenwert auch 5 bei der Helligkeit der Lampe bedeuten. Du weißt ja, dass deine Lampe 30 als kleinsten Helligkeitswert anzeigen kann.

Damit deine Rechnung stimmt, schaust du dir noch kurz den Wert 0 des Temperatursensors an. Er entspricht 30 an Helligkeit bei der Lampe. 15 mehr bei den Temperatursensoren bedeuten damit 5 mehr Helligkeit bei der Lampe. Also rechnest du zu der kleinsten Helligkeit von 30 die 5 der errechneten Helligkeit für die 15 der Temperatursensorenwerte und bekommst eine Helligkeit von 35 bei der Lampe. Wie du schon oben lesen konntest, ist das eine vereinfachte Rechnung. Für die drei Projekte in diesem Kapitel reicht dieses Verständnis völlig aus. Sensoren geben ihre Werte in der Realität nicht immer so gleichmäßig aus, sodass zum Beispiel bei der Messung der ein kleiner Unterschied weniger Temperaturveränderung bei niedrigen Temperaturen verursacht als bei hohen Temperaturen. Genauso verhält es sich auch mit der Helligkeit von Lampen. Doppelter Helligkeitswert bedeutet nicht automatisch, dass die Lampe doppelt so hell wird. Hier ist dein eigenes Austesten gefragt.

## Rechnung (kurzgefasst)

Der Temperatursensor zeigt 15 an. Die Temperatursensorenwerte reichen von 0 bis 30 und die Lampe kann eine Helligkeit zwischen 30 und 40 anzeigen.

15 in der Messung entspricht dann der Umrechnung  $(40 - 30) / 2 + 30 = 35$  als Helligkeit der Lampe.

Das bedeutet aber auch, dass es zu mehreren Sensorwerten des Temperatursensors die gleiche Helligkeitsstufe der Lampe geben muss, da es mehr Werte auf Seiten des Sensors als auf Seiten der Helligkeit der Lampe gibt. Genau kannst du dir das noch einmal in folgender Tabelle ansehen:

Temperatursensorwert	Helligkeitswert der Lampe
0–2	30
3–5	31
6–8	32
9–11	33
12–14	34
15–17	35
18–20	36
21–23	37
24–26	38
27–29	39
30	40

Tab. 8–1 Beispiel Umrechnung: Temperaturwerte in Helligkeitswerte

Die ganzen Berechnungen musst du nicht für jeden Sensor durchführen, da dies die Programmierbefehle schon für dich übernehmen.

Wenn du also Daten von einem Sensor bekommst und diese zum Beispiel in die Helligkeit der Lampe umrechnen möchtest, wird dies nicht vom Calliope mini vorgegeben, sondern das musst du als Programmierer selbst festlegen.

Damit dies aber nicht mit großen Tabellen und »wenn-dann«-Programmierblöcken gemacht werden muss, gibt es den »verteile«-Programmierbefehl des Calliope mini.

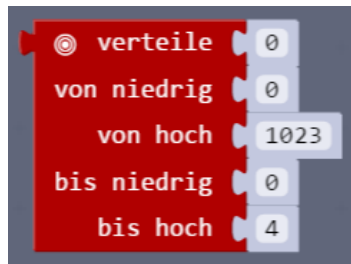


Abb. 8–4 »verteile«-Programmierbefehl

Der »verteile«-Befehl hat fünf Möglichkeiten, andere Programmierbefehle oder Blöcke anzudocken.

- ▶ »verteile«: Hier gibst du an, welche Daten du in einen anderen Wertebereich umrechnen möchtest. Das können ein Platzhalter oder auch direkte Sensordaten sein.
- ▶ »von niedrig«: die kleinste Zahl vor dem Umrechnen

- ▶ »von hoch«: die größte Zahl vor dem Umrechnen
- ▶ »bis niedrig«: die kleinste Zahl nach dem Umrechnen
- ▶ »bis hoch«: die größte Zahl nach dem Umrechnen

Wenn du den »verteile«-Befehl aus dem Untermenü »Pins« aus dem Menü »Fortgeschritten« auswählst, ist eine Verteilung standardmäßig schon angegeben, die du selbst verändern kannst. Wie das genau geht, lernst du in den drei Projekten in diesem Kapitel. Die Verteilung ähnelt dann der im obigen Beispiel und wird in der folgenden Skizze verdeutlicht.

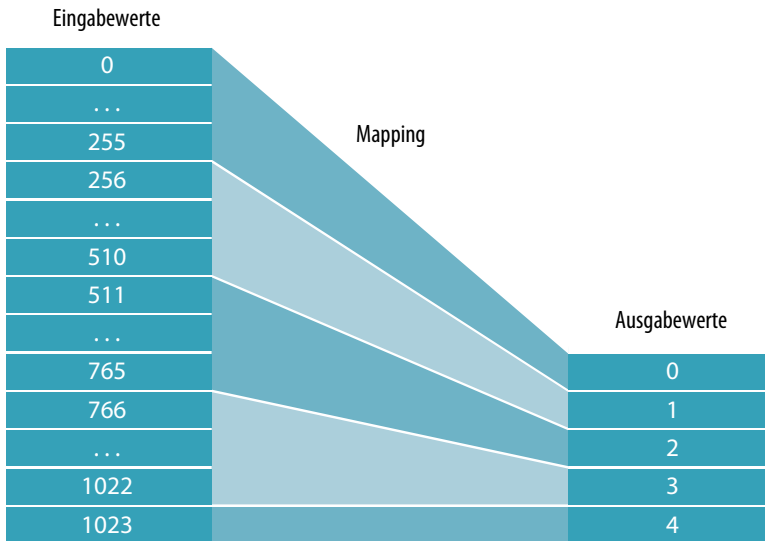


Abb. 8-5 Skizze der Standardumrechnung des »verteile«-Programmierbefehls

## 8.1.2 Berechnung des »verteile«-Programmierbefehl

Der »verteile«-Programmierbefehl rechnet anhand der Eingabewerte und der gewünschten Ausgabewerte die Bereiche selbst aus, so dass du dich darum nicht mehr kümmern musst. Für den Fall, dass es dich aber trotzdem interessiert, wie der Calliope mini dies berechnet, kannst du dir folgende Rechnung anschauen.

Nimm einmal an, du benutzt den »verteile«-Befehl in der Standardumrechnung wie oben gezeigt und der Sensor gibt den Wert 256 an den Calliope mini zurück. Dann sind die Werte folgendermaßen vorhanden:

Andockmöglichkeit im »verteile«-Programmierbefehl	Werte
verteile	256
von niedrig	0
von hoch	1023
bis niedrig	0
bis hoch	4

Tab. 8-2 Beispielrechnung Mapping durch den »verteile«-Programmierbefehl

Der Calliope mini berechnet jetzt die Formel:

$(\text{verteile-Wert} - \text{von niedrig}) \times (\text{von hoch} - \text{bis niedrig}) \div (\text{von hoch} - \text{von niedrig}) + \text{bis niedrig} = \text{Ergebnis}$

In dem Beispiel also

$(256 - 0) \times (4 - 0) \div (1023 - 0) + 0 = 1024 \div 1023 = 1,0009775$

Da die Zahl über 1 ist, schneidet der Calliope mini alle Ziffern hinter der 1 ab und gibt die 1 weiter an das Programm.

### 8.1.3 Sensoren und ihre Werte

In der folgenden Tabelle findest du eine Übersicht über die Sensoren des Calliope mini und ihre Wertebereiche sowie die Ausgabemöglichkeiten.

Sensor (Eingabe)	Niedrigster Zahlenwert	Höchster Zahlenwert
Beschleunigung	-1023	1023
Lichtstärke	0	255
Temperatur	-5	50

Tab. 8-3 Übersicht Wertebereiche der Sensoren (Eingabe)

Sensor (Eingabe)	Niedrigster Zahlenwert	Höchster Zahlenwert
Lautsprecher (Hz) <sup>1</sup>	20	20000
LED-Helligkeit	0	255

Tab. 8-4 Übersicht Wertebereiche der Ausgabe

Der Lautsprecher kann auch noch niedrigere und höhere Frequenzen abspielen, aber das menschliche Ohr kann diese nicht hören.

## 8.2 Projekt »Farbverlauf«

### 8.2.1 Farbverlauf – Sensoren durch Farben sichtbar machen

Wie du im »Mapping«-Kapitel schon lesen konntest, wirst du in diesem Projekt die Veränderung der Temperatur durch Farben sichtbar machen. Bisher konntest du dies nur in kleinen Stufen machen, die du selbst festlegen musstest. Das Mapping ermöglicht es dir nun, die Veränderungen, die ein Sensor misst, direkt auf zum Beispiel einer LED sichtbar zu machen. Je wärmer es wird, desto heller wird die LED rot leuchten.

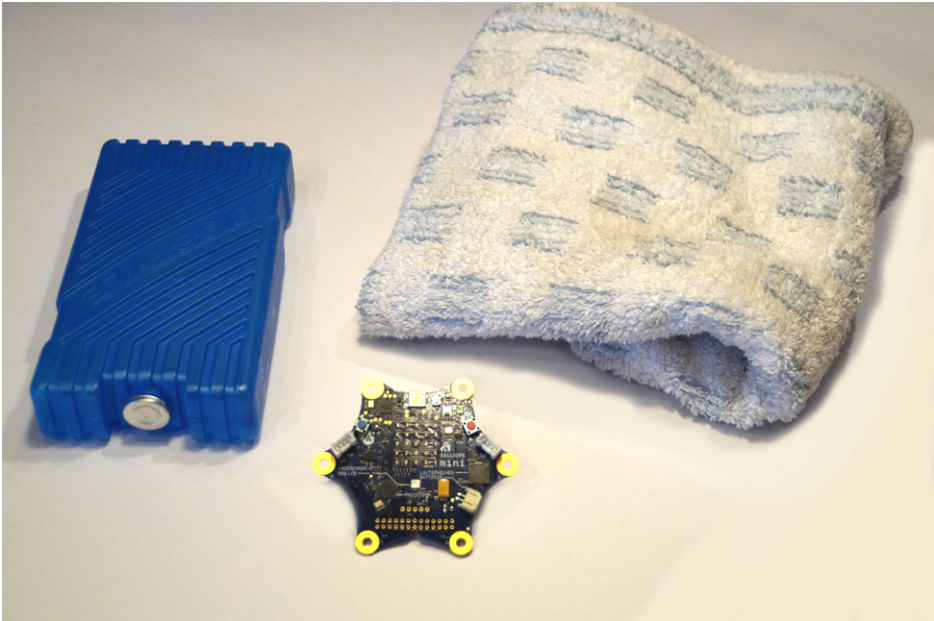


Abb. 8–6 Calliope mini – mal kalt, mal warm

### 8.2.2 Was brauchst du für dieses Projekt?

Das Projekt »Farbverlauf« kannst du auch ohne diese Materialien durchführen. Da du mit Temperatur experimentieren wirst, kannst du mit etwas Kaltem (z. B. einem Kühl-Akku) schneller die Effekte sehen. Alternativ kannst du auch einen Ventilator benutzen oder einfach selbst pusten.

Wie viele?	Was	Anmerkung
1	Handtuch	Nicht zu groß und ruhig ein altes Handtuch
1	Kühlpack	Findest du normalerweise im Eisfach des Kühlschranks.

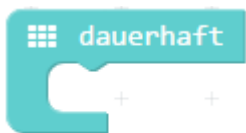
Tab. 8–5 Materialien für das Projekt »Farbverlauf«

## 8.2.3 Alles noch einmal überprüfen, bevor es losgeht!

Dies ist das erste Projekt zum Thema »Mapping«. Es schließt direkt an die Erklärungen am Anfang des Kapitels »Mapping« an. Lies dir dies am besten jetzt durch, wenn du es noch nicht gemacht hast. Du solltest dir folgende Dinge ebenfalls schon angeschaut haben:

- ▶ Grundlagen »dauerhaft«-Schleifen (Abschnitt 2.3.2)
- ▶ Platzhalter (Abschnitt 5.1)
- ▶ Sensoren (besonders den Temperatursensor in Abschnitt 4.1.1)
- ▶ Abschnitt 8.1 zu »Mapping«

## 8.2.4 Es geht los



Als Erstes brauchst du die »dauerhaft«-Schleife, da der Farbverlauf die ganze Zeit angezeigt werden soll.

Wähle die »dauerhaft«-Schleife aus den Grundlagen und ziehe sie in die Programmierfläche.

Abb. 8–7 »dauerhaft«-Schleife

Wie du in den vorherigen Projekten schon gelernt hast, brauchst du zur Speicherung der Daten, die du vom Sensor bekommst, einen Platzhalter. Im Menü »Platzhalter« kannst du dir einen neuen Platzhalter anlegen und zu deinem Programm hinzufügen.

Lege einen neuen Platzhalter mit dem Namen »helligkeit« an. Der neue Platzhalter wird dann im Menü »Platzhalter« sichtbar.



Abb. 8–8 Platzhalter zur Speicherung der Daten anlegen

Damit du etwas in dem Platzhalter speichern kannst, brauchst du den Programmierbefehl, der den Wert eines Platzhalters auf einen neuen Eingabewert ändert.

Zieh den Programmierbefehl »ändere Platzhalter auf« in die »dauerhaft«-Schleife deines Programms.

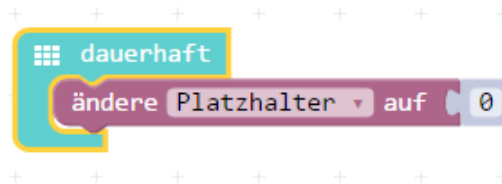


Abb. 8–9 Platzhalter in »dauerhaft«-Schleife eingefügt

Da du den neuen Platzhalter »helligkeit« angelegt hast, kannst du ihn jetzt im »ändere Platzhalter auf«-Programmierbefehl auswählen.

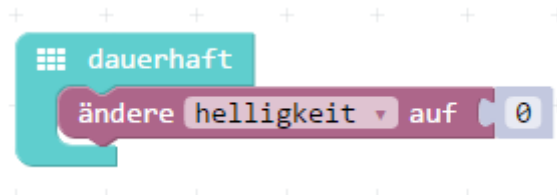


Abb. 8–10 Platzhalter umgestellt auf »helligkeit«

Klicke auf Platzhalter und ändere den zu ändernden Platzhalter auf »helligkeit«.

Nachdem du deinen Platzhalter jetzt im Programm untergebracht hast, kannst du Werte in ihm speichern. Da du die Werte durch das Mapping in Werte für die Helligkeit der roten LED umrechnen willst, brauchst du den neuen Programmierbefehl »verteile«.

Du findest ihn im Menü »Fortgeschritten«, wenn du das Untermenü »Pins« anwählst.

## Was meint Mapping?

Der Begriff »Mapping« kommt wie fast alle Programmierbegriffe aus der englischen Sprache und kann auf Deutsch mit »Verteilung« übersetzt werden. Die Sensordaten des Temperatursensors werden umgerechnet und so auf die Eingabewerte der Helligkeit der RGB-LED verteilt.



Abb. 8–11 Untermenü »Pins« des Menüs »Fortgeschritten«

Klicke auf »Fortgeschritten« im Menü. Anschließend klicke auf »Pins« und zieh den Programmierbefehl »verteile« auf die Programmierfläche.

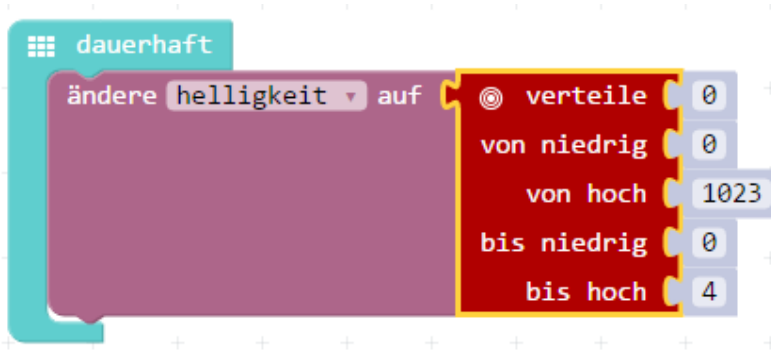


Abb. 8-12 »verteile«-Programmierbefehl

Setz den »verteile«-Befehl an den »ändere helligkeit auf«-Befehl heran.

Die Daten können jetzt durch den »verteile«-Befehl umgerechnet und dann im Platzhalter gespeichert werden. Jetzt musst du den »verteile«-Befehl so anpassen, dass er die richtigen Daten bekommt und dann weiß, wie die Daten umgerechnet werden sollen. Da du die Daten des Temperatursensors umrechnen willst, muss er zuerst an den »verteile«-Befehl ganz oben angehängt werden.

Klicke in dem Menü »Eingabe« auf den Block »Temperatur (°C)« und zieh ihn in die Programmierfläche.

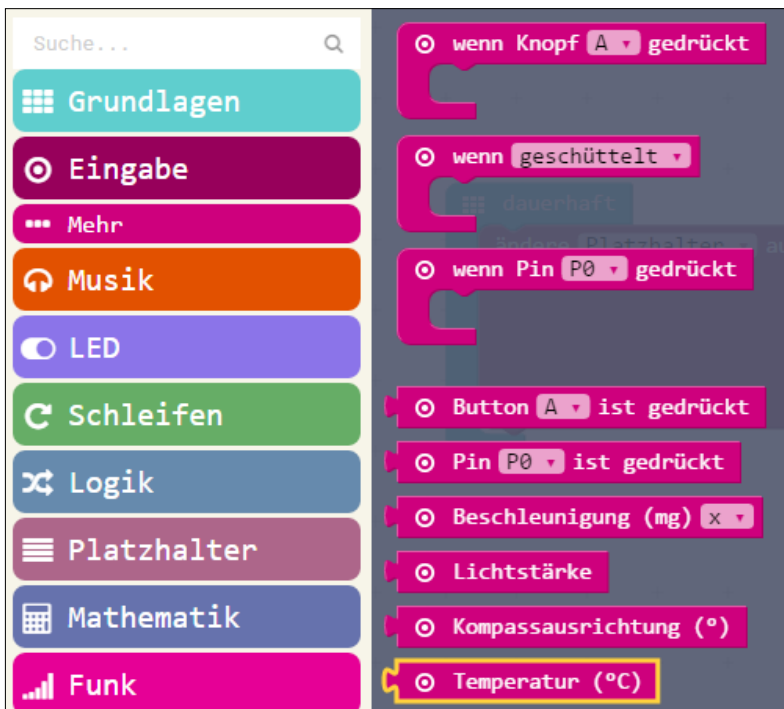


Abb. 8-13 Temperatursensor auswählen

Häng den Block »Temperatur (°C)« an den »verteile«-Befehl an.

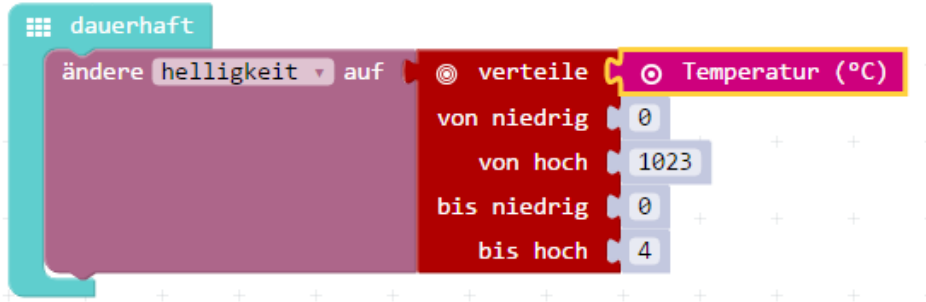


Abb. 8–14 Mapping der Temperaturdaten auf den Platzhalter »helligkeit«

Die Daten des Temperatursensors, die dir durch den Befehl »Temperatur (°C)« in Grad Celsius zur Verfügung stehen, werden jetzt an den »verteile«-Befehl übergeben. Dieser Befehl rechnet die Daten um. Im Anschluss werden sie dann im Platzhalter »helligkeit« gespeichert. Dies wird immer wieder wiederholt, da sich dein Programmcode in einer »dauerhaft«-Schleife befindet.

Die Standardwerte für die Umrechnung sind »von niedrig« 0 und »von hoch« 1023 auf »bis niedrig« 0 und »bis hoch« 4. Das bedeutet, dass alle Werte des Sensors, die zwischen 0 und 1023 liegen, jetzt auf die Werte von 0 bis 4 aufgeteilt werden. Wie das genau aussieht, kannst du dir noch einmal am Anfang des »Mapping«-Kapitels durchlesen, wenn du dir unsicher bist.



Abb. 8–15 Mapping Temperaturanpassung

Temperaturen von 1023 Grad sind viel zu hoch und würden dich und den Calliope mini gefährden. Damit du schnell Änderungen in der Temperatur siehst, wählst du am besten eine Temperatur zwischen 28° Celsius und 35° Celsius, da der Calliope mini in einem normal beheizten Raum in etwa diese Temperatur annimmt.

Ändere die 0 bei »von niedrig« auf 28 und danach die 1023 bei »von hoch« auf 35.

Super, dein Calliope mini wird jetzt Temperaturen von 28° Celsius bis 35° Celsius umrechnen. Jetzt musst du dich noch um die Helligkeit der LED kümmern.

### Werte unter- und oberhalb

Alle Zahlen über 35 und unter 28 werden weiter umgerechnet nach der Formel. Willst du also keine höheren oder niedrigeren Temperaturen messen, musst du dies selbst vor dem »verteile«-Programmierbefehl verhindern. Ansonsten kann dein Programm sich in den Bereichen, an die du nicht gedacht hast, anders verhalten, als du es erwartest.

Wie du bereits gelernt hast, können die Farben der LED auf eine Helligkeit von 0 (LED aus) bis 255 (LED volle Helligkeit) eingestellt werden. Da du möglichst viele Veränderungen in der Helligkeit sehen möchtest, nutzt du am besten die volle Skala aus.

Ändere die 4 bei »bis hoch« auf die volle Helligkeit, also 255.

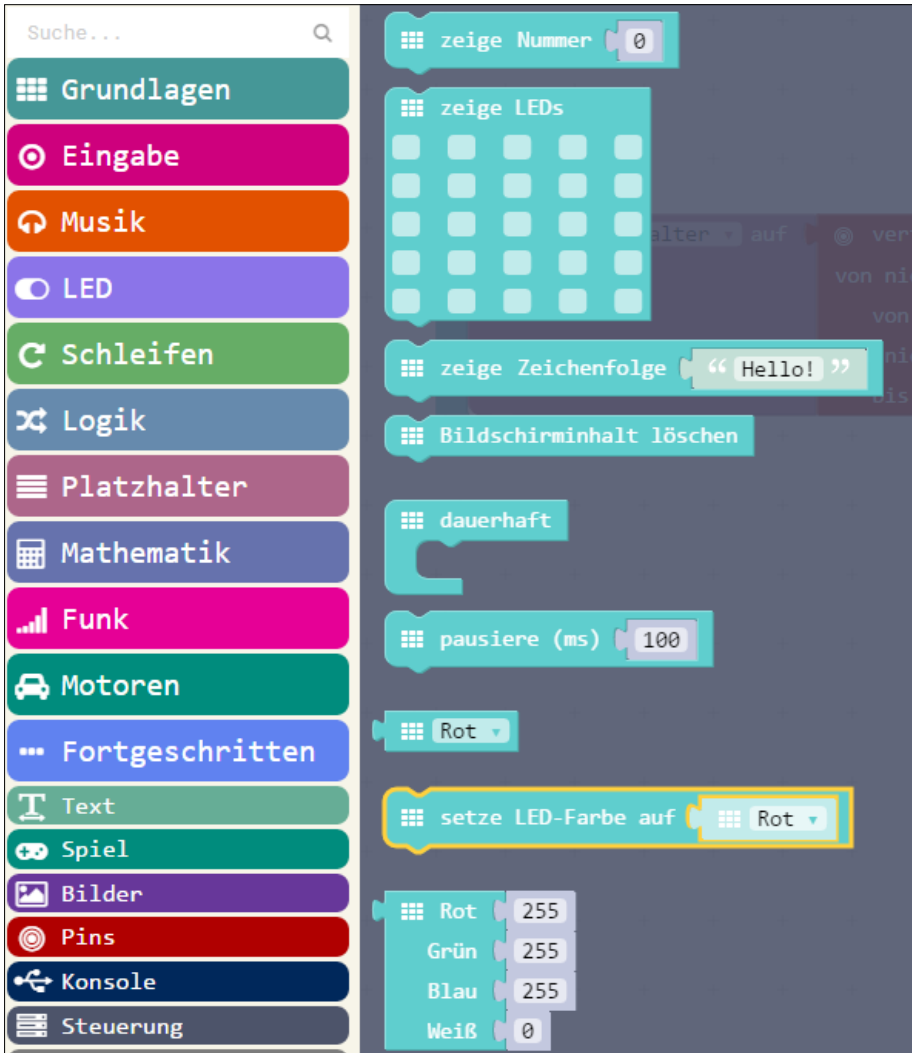


Abb. 8-16 Menü »Grundlagen« – LED-Farbe einstellen

Prima. Die »Mapping«-Funktion hast du so eingestellt, dass die Temperatur, die der Temperatursensor misst, in die Werte der Helligkeit umgerechnet wird. Anschließend wird der errechnete Wert automatisch in den Platzhalter »helligkeit« gespeichert.

Jetzt muss dein Platzhalter nur noch die Helligkeit der LED steuern. Dafür brauchst du als Erstes den »LED«-Befehl aus dem Menü »Grundlagen«.

Zieh aus dem Menü »Grundlagen« den Programmierbefehl »setze LED-Farbe auf Rot« in der Programmierfläche.

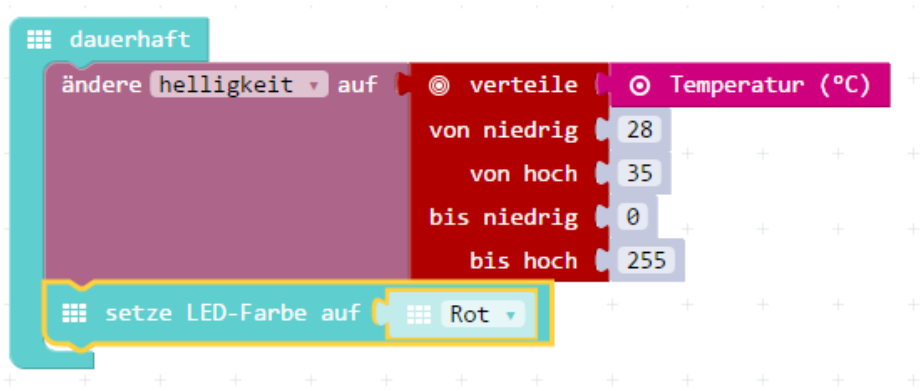


Abb. 8–17 Eingesetzter »LED«-Programmierbaustein

Setz den Befehl als letzten Block in die »dauerhaft«-Schleife ein.

Die LED-Farbe wird jetzt in deinem Programm auf Rot gesetzt. Da du aber die Helligkeit der LED verändern willst, brauchst du noch einen weiteren Programmierbefehl, bei dem du die Helligkeit aller Farben direkt angeben kannst.



Abb. 8–18 »Farb-LED«-Programmierbefehl

Geh in das Menü »Grundlagen« und zieh den »Farb-LED«-Programmierbefehl in den Programmierbereich.

Da du die LED-Farbe durch den Platzhalter »helligkeit« verändern willst, musst du das »Rot« durch den neuen Programmierbefehl ersetzen.

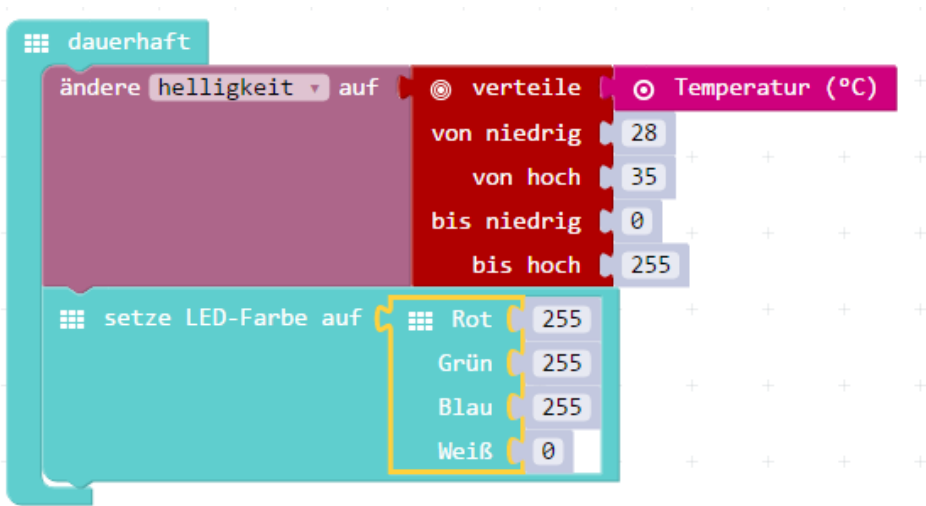


Abb. 8–19 Eingesetzter »Farb-LED«-Programmierbefehl

Zieh den »Farb-LED«-Programmierbefehl an die Stelle der Farbe »Rot«.

Was dir jetzt noch fehlt, ist der Platzhalter »helligkeit«, damit du die Helligkeit der roten LED auf die Werte einstellen kannst, die durch die »Mapping«-Funktion schon im Platzhalter »helligkeit« gespeichert wurden.



Abb. 8–20 Platzhalter »helligkeit«

Zieh aus dem Menü »Platzhalter« den Platzhalter »helligkeit« in die Programmierfläche.

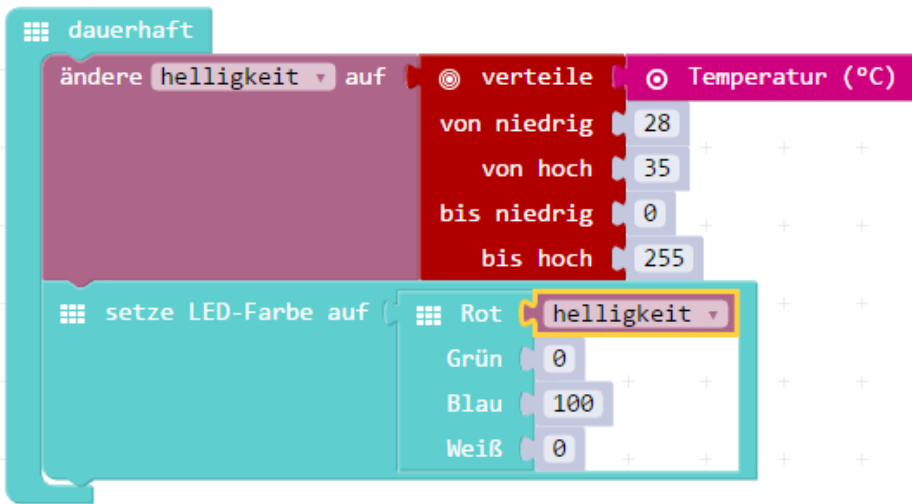


Abb. 8-21 Eingesetzter Platzhalter »helligkeit«

Zieh den Platzhalter »helligkeit« an die Farbe »Rot«. Ändere den Wert von »Grün« auf 0 und den Wert von »Blau« auf 100.

Toll, du hast das Projekt »Farbverlauf« richtig programmiert. Den Wert bei »Blau« benutzt du nur für den schöneren Farbton, da du in »Rot« noch »Blau« hinzufügst.

## 8.2.5 Testen

Macht dein Programm das, was es soll? Teste dein Programm beispielsweise, indem du den Calliope mini in die Hand nimmst oder einmal nah an die Heizung hältst. Steigt die Temperatur, sollte die rote LED heller werden, sinkt die Temperatur des Calliope mini wieder, indem du zum Beispiel pustest, sollte die rote LED immer dunkler werden.

Wenn etwas nicht funktionieren sollte, dann vergleiche deinen Programmcode noch einmal Schritt für Schritt mit der Anleitung. Manchmal schleichen sich kleine Fehler ein.

## 8.2.6 Experimentieren

Jetzt geht es ans Experimentieren. Wenn du ein Handtuch und ein Kühlpack aus dem Gefrierfach hast, kannst du das Kühlpack in das Handtuch einwickeln und so den Calliope mini schneller abkühlen. Wenn du den Calliope mini in die Sonne legst, sollte die Temperatur auch schnell ansteigen. Aber Vorsicht: Wasser und große Hitze beschädigen den Calliope mini. Überleg dir einmal, was du noch alles

messen könntest und an welchen Stellen eine Temperaturmessung mit dem Calliope mini helfen kann.

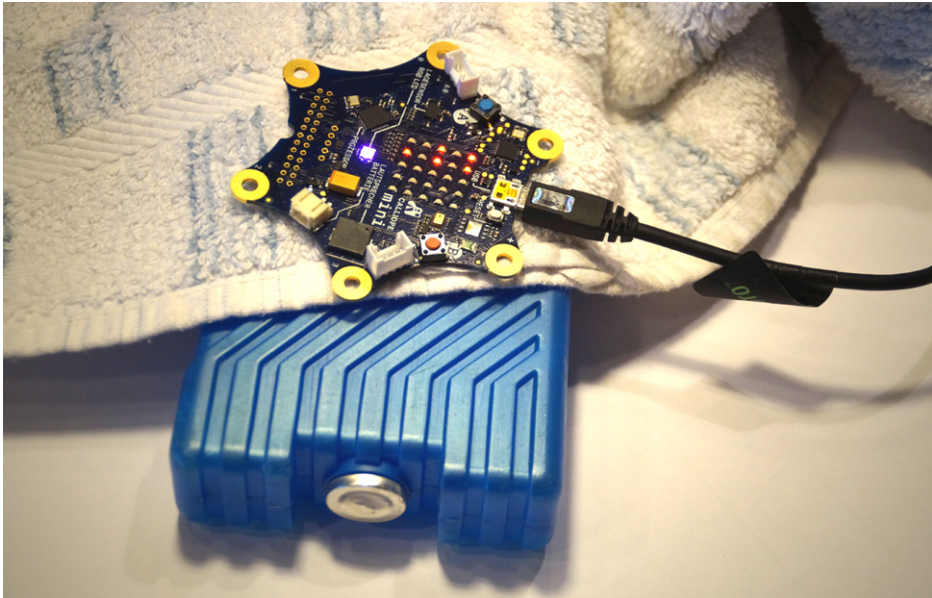


Abb. 8-22 Der Calliope mini wird gekühlt

## 8.2.7 Da geht noch mehr?! – Optimierte dein Projekt

Hier findest du weitere Ideen, wie du dein Projekt noch verbessern kannst. Das Projekt »Farbverlauf« kannst du noch zu einem tollen Thermometer weiterentwickeln.

- ▶ Füge eine Display-Anzeige hinzu und zeige die aktuelle Temperatur des Calliope mini an.
- ▶ Genauso wie die aktuelle Temperatur kannst du dir auch den momentanen durch die »Mapping«-Funktion umgerechneten Wert des »helligkeit«-Platzhalters ausgeben lassen.
- ▶ Durch »Bildschirm löschen« kannst du sogar beides hintereinander ausgeben lassen oder auch ein »Achtung«, falls die Temperatur zu hoch sein sollte.

## 8.3 Projekt »Luftgitarre«

### 8.3.1 Luftgitarre – Bewegung und Musik verbinden

Du kennst Luftgitarren wahrscheinlich schon. Wenn du eine tolle Musik hörst und eine Gitarre in dem Lied vorkommt, macht es Spaß, die Gitarre in der Luft nachzuspielen, während die Musik spielt. Dazu kannst du dir auch eine Gitarre basteln, dann macht es noch mehr Spaß.

Aber was hat das mit einem Mikrocontroller und speziell mit dem Calliope mini zu tun? Tja, du wirst deine Luftgitarre so verändern, dass deine Gitarre bei jeder Bewegung Töne von sich geben wird. Dadurch kannst du nicht nur zur Musik mit der Luftgitarre spielen, sondern auch allein mit der Luftgitarre Musik erzeugen.

Wie wird deine Luftgitarre funktionieren, wenn du sie fertig gebastelt und programmiert hast? Du wirst den Beschleunigungssensor des Calliope mini zusammen mit der »Mapping«-Funktion und der Möglichkeit des Calliope mini, auch Töne ausgeben zu können, kombinieren. Hältst du deine Luftgitarre und bewegst dich mit dem Oberkörper nach vorne und hinten, wird deine Luftgitarre einen anderen Ton abspielen. Ziehst du die Gitarre mit dem Griff nach oben, wird der Calliope mini die Töne umso schneller abspielen, je höher du den Griff hältst.

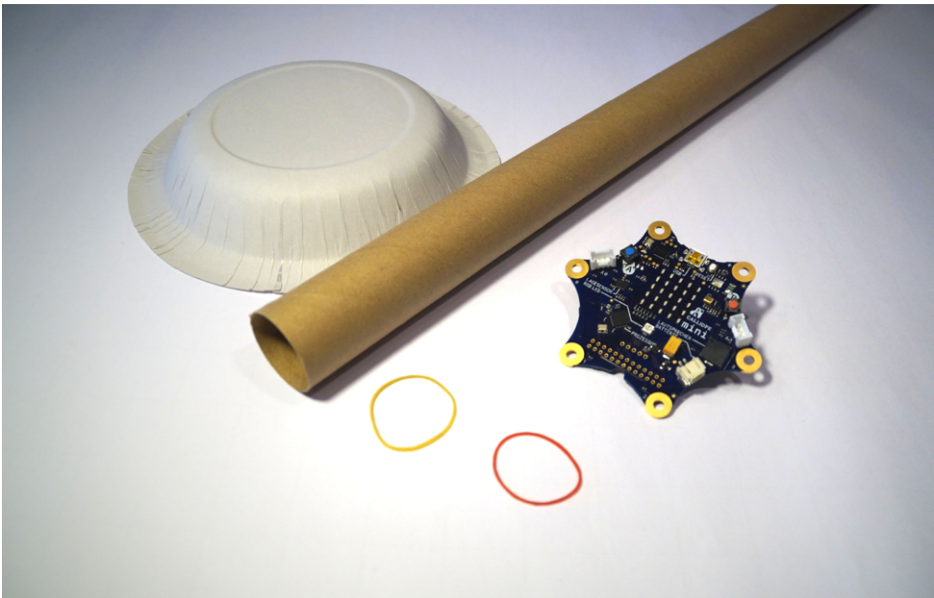


Abb. 8–23 Die Luftgitarre

## 8.3.2 Was brauchst du für dieses Projekt?

Das Projekt »Luftgitarre« kannst du auch ohne diese Materialien durchführen, aber mit ein wenig Bastelgeschick fühlt sich die Luftgitarre fast wie eine echte Gitarre an.

Wie viele?	Was	Anmerkung
1	Papprolle	Zum Beispiel aus einer Geschenkpapierrolle
1	Pappteller	Du kannst dir hier aber auch etwas aus einem Stück Papier rund ausschneiden.
1	Kleber oder Tesafilm	Achte nur darauf, dass der Kleber auch für Papier und Pappe geeignet ist.

Tab. 8–6 Materialien für das Projekt »Luftgitarre«

## 8.3.3 Alles noch einmal überprüfen, bevor es losgeht!

Mapping ist nicht einfach zu verstehen. Du kannst das folgende Projekt auch anhand der Beschreibungen Schritt für Schritt nachvollziehen, um deine Luftgitarre zu basteln. Besser ist es aber, wenn du dir den Abschnitt 8.1 zu »Mapping« vorher durchliest, dann kannst du dein Projekt noch weiter verbessern. Du solltest dir folgende Dinge schon angeschaut haben:

- ▶ Grundlagen »dauerhaft«-Schleifen (Abschnitt 2.3.2)
- ▶ Platzhalter (Abschnitt 5.1)
- ▶ Sensoren (besonders den Beschleunigungssensor (Abschnitt 4.1.2))
- ▶ Abschnitt 8.1 zu »Mapping«

## 8.3.4 Es geht los

Als Erstes brauchst du die »dauerhaft«-Schleife. Deine Luftgitarre soll nicht irgendwann mittendrin aufhören, Musik zu spielen.

1. Wähle die »dauerhaft«-Schleife aus den »Grundlagen« und zieh sie in die Programmierfläche.



Abb. 8–24 »dauerhaft«-Schleife

Als Nächstes beschäftigst du dich mit dem Menü »Musik«. Hier findest du alle Programmierbefehle, die mit der Musikfunktion des Calliope mini zu tun haben. Du kannst einzelne Noten abspielen lassen, einen Klingelton erzeugen oder die Geschwindigkeit der Töne verändern.

2. Klicke auf »Musik«, und das Musikmenü geht auf, das dir alle zugehörigen Programmierbefehle zeigt.



Abb. 8–25 Musikfunktionen des Calliope mini

Der für dich interessante Programmierbefehl ist »spiele Note«. Wie du siehst, hat »spiele Note« zwei Möglichkeiten, Blöcke einzuhängen: zuerst die Note und als Zweites den Takt.

3. Ziehe den Programmierbefehl »spiele Note« in die »dauerhaft«-Schleife.



Abb. 8–26 Der Programmierbefehl »spiele Note«

Wenn du auf das C drückst, bekommst du alle vorgegebenen Noten als Auswahl. Die niedrigste Note ist das C und die höchste auswählbare Note ist das B5. Dies kennst du aus dem Musikunterricht, ansonsten probier die verschiedenen Noten einfach vorher einmal aus.

In der zweiten Stelle kannst du den Takt einstellen. Je kleiner der Takt, desto kleiner ist auch die Pause zwischen zwei Noten. Vielleicht hast du die Musikfähigkeiten des Calliope mini auch schon in einem anderen Projekt umgesetzt.

Die Idee hinter deiner Luftgitarre ist, dass du die Bewegung der Luftgitarre auf die Noten und die Geschwindigkeit, in der sie abgespielt werden, umrechnest. Dazu brauchst du zuerst den »verteile«-Befehl und danach den richtigen Sensor.



Abb. 8–27 »Fortgeschritten«-Menü

Den »verteile«-Befehl findest du, wenn du auf das »Fortgeschritten«-Menü klickst und danach auf das Untermenü »Pins«.

4. Finde den »verteile«-Befehl und zieh ihn in die Programmierfläche.



Abb. 8–28 Programmierbefehl »verteile«

Der »verteile«-Befehl hat fünf Möglichkeiten andere Programmierbefehle oder Blöcke anzudocken.

- ▶ »verteile«: Hier gibst du an, welche Daten du in einen anderen Wertebereich umrechnen möchtest. Das können ein Platzhalter oder auch direkte Sensordaten sein.
- ▶ »von niedrig«: die kleinste Zahl vor dem Umrechnen
- ▶ »von hoch«: die größte Zahl vor dem Umrechnen
- ▶ »bis niedrig«: die kleinste Zahl nach dem Umrechnen
- ▶ »bis hoch«: die größte Zahl nach dem Umrechnen

Schau dir dazu auch noch einmal die Wertebereiche der Sensoren am Ende des »Mapping«-Kapitels an.

5. Ersetze die Note C durch den »verteile«-Befehl.



Abb. 8–29 Eingesetzter »verteile«-Befehl

Da deine Luftgitarre nicht nur Musik abspielen, sondern auch verändern kann, wenn du die Luftgitarre bewegst, brauchst du einen Sensor, der Bewegungen wahrnehmen kann. Der Calliope mini hat dafür zum Glück schon einen passenden Sensor eingebaut: den Beschleunigungssensor. Der Beschleunigungssensor kann auf drei Achsen die Bewegung messen.

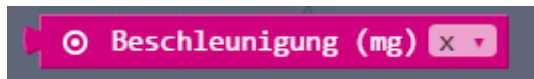


Abb. 8–30 »Beschleunigungssensor«-Block

## Hinweis

Wenn du dir unsicher bist, wie der Beschleunigungssensor genau funktioniert und welche Achsen genau welche Drehung bedeuten, dann schau noch einmal im Abschnitt 4.1.2 nach.

6. Zieh den »Beschleunigungssensor«-Block in den »verteile«-Befehl an die Stelle »verteile«.

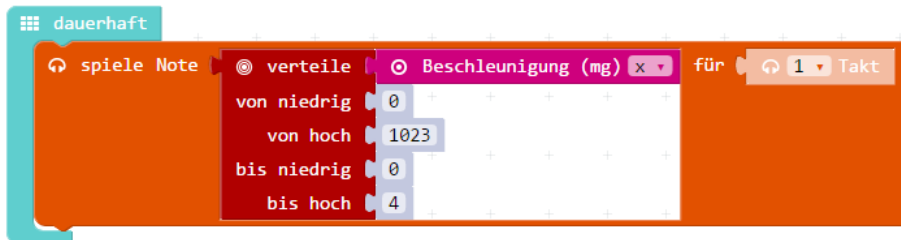


Abb. 8–31 Eingesetzter Beschleunigungssensorwert

Der Beschleunigungssensor gibt dir die Daten wieder, die er misst, wenn der Calliope mini über die x-Achse gedreht wird. Da du den Calliope mini bei der Luftgitarre auf dem Pappteller befestigen willst und die Töne sich ändern sollen, wenn du die Luftgitarre nach vorne und nach hinten bewegst, lässt du die schon ausgewählte x-Achse stehen.

Jetzt musst du den Wertebereich des Beschleunigungssensors eingeben, damit jede Bewegung der Luftgitarre auch einem Ton zugeordnet werden kann. Der Beschleunigungssensor hat einen Wertebereich von -1023 bis 1023. Wenn du den Calliope mini flach mit der Beschriftung nach oben auf den Tisch legst, gibt der Sensor mit dem Programmierblock »Beschleunigung (mg) x« eine 0 zurück. Kippst du den Calliope mini von dir weg, wird der gemessene Wert immer kleiner bis zum Wert -1023, wenn der Calliope mini senkrecht steht. Kippst du den Calliope mini stattdessen zu dir hin, dann erhöhen sich die Werte immer weiter bis 1023.

## Achtung

Drehst du den Calliope mini weiter, dann wechselt er direkt von 1023 auf -1023 oder andersherum.

- 7. Gib beim »verteile«-Befehl bei »von niedrig« -1023 ein und bei »von hoch« 1023.

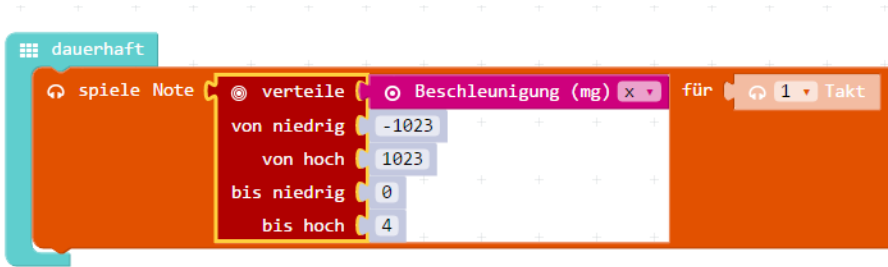


Abb. 8-32 »verteile«-Programmierbefehl mit Sensor-Wertebereich

Du hast den Wertebereich des Beschleunigungssensors gesetzt. Jetzt fehlt noch der neue Wertebereich, in den du die Daten des Beschleunigungssensors umrechnen willst. Da du verschiedene Musiktöne machen willst, musst du die Töne in der zugehörigen Frequenz angeben. In der Tabelle findest du zu bestimmten Noten die passende Frequenz.

Note	Frequenz
C	262
D	294
E <sub>b</sub>	311
E	330
F	349
FSharp	370
G	392
GSharp	415
A	440
B <sub>b</sub>	466
B	494
C <sub>5</sub>	523
...	...
B <sub>5</sub>	989

Tab. 8-7 Töne und Frequenzen

### Link

Eine Liste der Frequenzen und den zugeordneten Tönen findest du hier:  
<http://pianotip.de/frequenz.html>

Damit du die ganze Bandbreite der Töne abdecken kannst, solltest du den tiefsten Ton als niedrigsten Wert für die Frequenz wählen und den höchsten Ton als höchsten Wert der Frequenz.

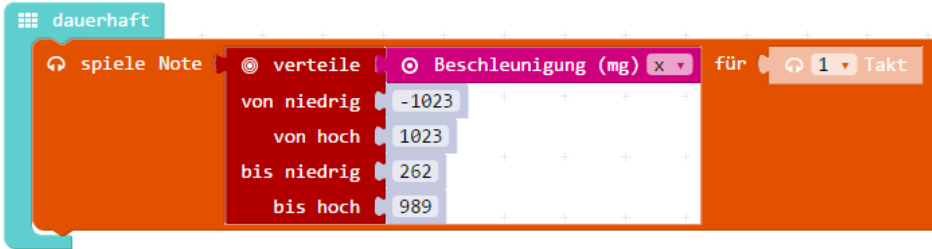


Abb. 8–33 »Mapping«-Programmierbefehl mit vollständigen Werten.

8. Trag bei »bis niedrig« 262 für den Ton C und bei »bis hoch« 989 für den Ton B5 ein.

Teste diesen Zwischenstand jetzt ruhig einmal aus. Du kannst später hier auch andere Werte ausprobieren und sehen, wie sich die Musik deiner Luftgitarre verändert.

Als nächsten Schritt beschäftigst du dich mit dem Takt. Da du eine Luftgitarre programmierst, soll die Musik schnell genug sein. Du musst also die Pausen zwischen den Tönen möglichst kurz stellen.

9. Stell bei Takt von 1 auf 1/16 um. Dadurch werden die Pausen zwischen den Tönen sehr kurz.



Abb. 8–34 Mapping vollständig für die Tonwiedergabe

10. Super! Du kannst das Programm jetzt schon einmal austesten. Überspiele dein Programm auf den Calliope mini und teste es aus.

Werden Töne gespielt, wenn du den Calliope mini bewegst? Du kannst auch die Wertebereiche anpassen, bis dir die Töne am besten gefallen.

Jetzt kann deine Luftgitarre schon Töne spielen, wenn sie nach vorne oder hinten geneigt wird. Was jetzt noch fehlt, ist, dass die Musik schneller wird, je mehr

du die Luftgitarre nach oben hältst. Dazu wirst du wieder den Beschleunigungssensor benutzen, um eine andere Achse auszulesen, und den »Mapping«-Programmierbefehl, um die Sensorwerte auf die Geschwindigkeit der Musik umzurechnen. Zu allererst brauchst du dazu einen neuen Platzhalter, den du »bpm« nennst.

## BPM

BPM steht für *Beats per Minute*. Das heißt übersetzt »Schläge pro Minute« und beschreibt die Geschwindigkeit, in der ein Lied abgespielt wird. Elektronische Musik in der Disco oder auch bestimmte Bereiche der Metal Music können eine Geschwindigkeit von über 200 BPM haben. Normalerweise haben Lieder unter 120 BPM.

11. Leg einen neuen Platzhalter an und nenn ihn »bpm«.
12. Aus dem Menü »Platzhalter« wählst du den Programmierbefehl »ändere Platzhalter auf« und fügst ihn an als letzten Befehl in die »dauerhaft«-Schleife ein.
13. Ändere den Platzhalter auf »bpm«.

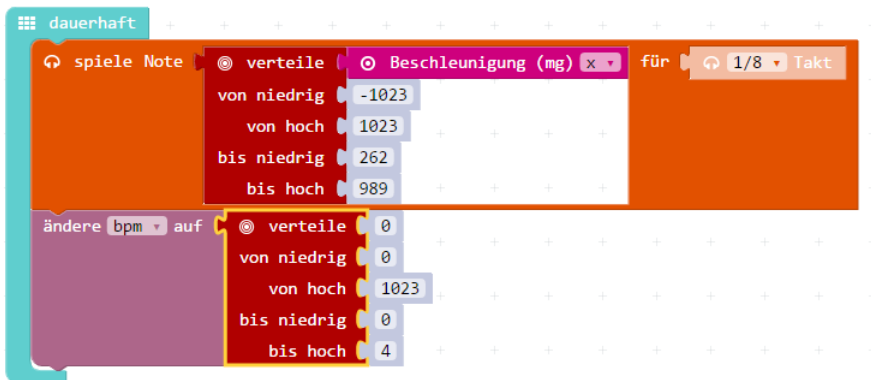


Abb. 8–35 BPM-Mapping

Jetzt hast du den Platzhalter hinzugefügt, in dem du die Werte, die der Sensor dir geben wird, speichern kannst. Da der Sensor andere Sensordaten zurückgibt, als du für die Geschwindigkeit der Musik benutzen kannst, musst du die Werte umrechnen. Dazu kannst du wieder den »verteile«-Programmierbefehl nehmen.

14. Füge dem »ändere bpm auf«-Befehl den »verteile«-Programmierbefehl aus dem »Fortgeschritten«-Menü des Untermenüs »Pins« an.

Genauso wie beim Mapping weiter oben brauchst du den Beschleunigungssensor und seinen Wertebereich. Diesmal benutzt du aber nicht die x-Achse zur Messung, sondern die y-Achse.

15. Setz den Beschleunigungssensor aus dem Menü »Eingabe« an die Stelle »verteile« und ändere die Achse von »x« auf »y«.
16. Jetzt passt du den Wertebereich genauso wie vorher an den Beschleunigungssensor an.



Abb. 8–36 Mapping mit der y-Achse des Beschleunigungssensors

Den Wertebereich der Musiknoten hattest du aus den Frequenzen der Töne herausgearbeitet, für den Platzhalter »bpm« musst du nun einen neuen Wertebereich bestimmen. Wie du schon in diesem Projekt gelesen hast, können sehr schnelle Musikstücke über 200 BPM schnell sein. Daher setzt du »bis hoch« am besten auf 200 und »bis niedrig« auf 1.

Für die Sensordaten des Beschleunigungssensors auf der y-Achse gibt der Sensor Werte von -1023 bis 1023 wieder. Wenn du den Calliope mini mit der flachen Seite gegen deinen Bauch mit dem USB-Anschluss nach rechts hältst und nun die linke Seite des Calliope mini nach oben neigst, dann gibt der Sensor dir immer höhere Werte, je stärker du den Calliope mini nach oben neigst. Genau andersherum verändern sich die Werte in die andere Richtung, sodass die Werte immer kleiner werden.

Der »Mapping«-Programmierbefehl rechnet also den Wertebereich von -1023 bis 1023 auf den neuen Wertebereich von 1 bis 200 für dich um.

17. Verändere »bis niedrig« auf 1 und »bis hoch« auf 200.

Du hast den Platzhalter »bpm« jetzt auf die passenden Werte der Musikgeschwindigkeit umgerechnet. Jetzt musst du noch die Geschwindigkeit der Musik auf den errechneten BMP-Wert setzen. Dazu findest du im Menü »Musik« den Pro-

grammierbefehl »ändere die Geschwindigkeit auf (bpm)«. Der Befehl ändert die Geschwindigkeit der aktuell gespielten Musik auf die angegebene Zahl. Da du die Musikgeschwindigkeit auf den Platzhalter »bpm« setzen willst, muss er dort angedockt werden.

18. Füge den Programmierbefehl »ändere Geschwindigkeit auf (bpm)« aus dem Menü »Musik« als letzten Befehl in die »dauerhaft«-Schleife ein.
19. Ersetze die angegebenen 120 BPM durch deinen Platzhalter »bpm«.



Abb. 8–37 »Luftgitarre«-Programmierung komplett

Riesig! Dein Programm zur Luftgitarre ist fertig. Jetzt kannst du die Luftgitarre austesten. Rock' los!

### 8.3.5 Testen

Macht dein Programm das, was es soll? Teste dein Programm, indem du den Callope mini über die x-Achse und y-Achse kippst. Du solltest die Musik nun in der Geschwindigkeit und den Tönen ändern können.

Wenn etwas nicht funktionieren sollte, vergleiche deinen Programmcode noch einmal. Manchmal schleichen sich kleine Fehler ein.

### 8.3.6 Basteln

Jetzt geht es ans Basteln, damit das Projekt noch mehr Spaß macht. Deine Luftgitarre kann jetzt schon vollständig benutzt werden. Was jetzt noch fehlt, ist eine Hülle, damit sich die Luftgitarre nachher fast wie eine echte Gitarre anfühlt. Dazu brauchst du ein längeres Stück Pappe, zum Beispiel eine Papprolle, die bei

Geschenkpapier übrigbleibt, und eine größere Fläche, auf der du den Calliope mini befestigen kannst – zum Beispiel einen Pappteller.

Klebe den Pappteller auf das untere Ende der Papprolle. Befestige danach den Calliope mini auf dem Pappteller mit ein paar Gummis. Du kannst die gebastelte Luftgitarre jetzt noch anpassen, indem du sie anmalst.

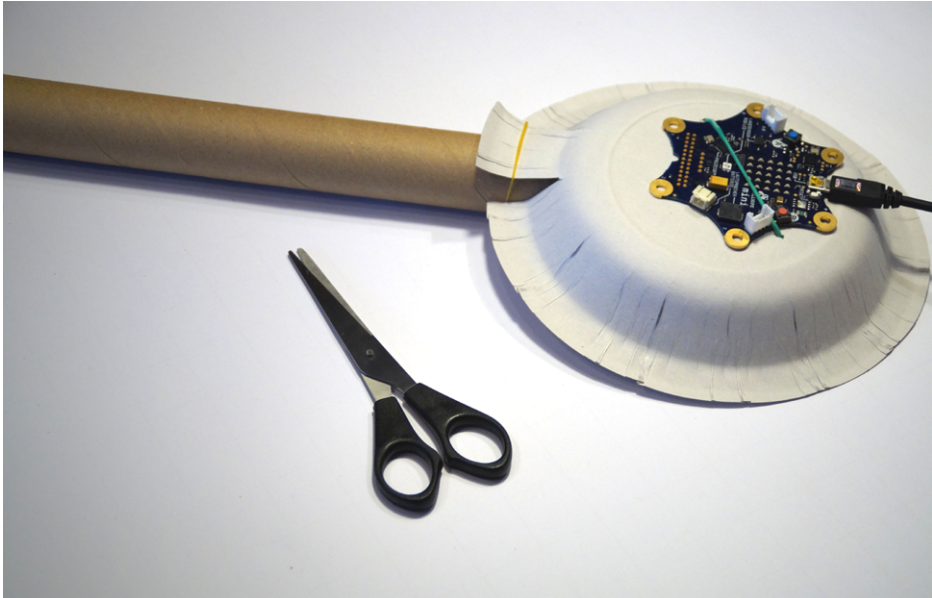


Abb. 8–38 Voila – die fertige Luftgitarre

### 8.3.7 Da geht noch mehr?! – Optimierte dein Projekt

Hier findest du weitere Ideen, wie du dein Projekt noch verbessern kannst. Es liegt an dir, die beste Calliope mini-Luftgitarre zu programmieren.

- ▶ Du kannst mit den Wertebereichen spielen und den Sound deiner Luftgitarre immer weiter optimieren.
- ▶ Ergänze Disco-Lichter an der Luftgitarre, indem du die LED und das Display am Calliope mini mit in dein Programm einbindest.
- ▶ Du kannst die Knöpfe des Calliope mini benutzen, um den Tonbereich auf Knopfdruck zu ändern, sodass deine Luftgitarre dann nur noch hohe oder tiefe Töne spielt.

## 8.4 Projekt »Fang den Dot«

### 8.4.1 »Fang den Dot« – Ein Geschicklichkeitsspiel

In diesem Profiprojekt wirst du ein Geschicklichkeitsspiel mit dem Calliope mini programmieren. In dem Spiel steuerst du einen kleinen Punkt mit dem Namen »Dot«. Den Dot kannst du steuern, indem du den Calliope mini über die x-Achse und y-Achse kippst. Auf diese Weise kann sich der Dot auf dem ganzen Display hin und her bewegen.

Das ist ja schon einmal ganz toll. Aber auf Dauer bekommt der Dot Langeweile und sucht sich einen zweiten Dot zum spielen. Das Spiel geht so, dass der erste Dot den zweiten Dot, der in seiner Umgebung erscheint, fangen muss. Der Spieler, der dein Spiel spielt, soll also möglichst schnell mit dem steuerbaren Dot den automatischen Dot fangen.

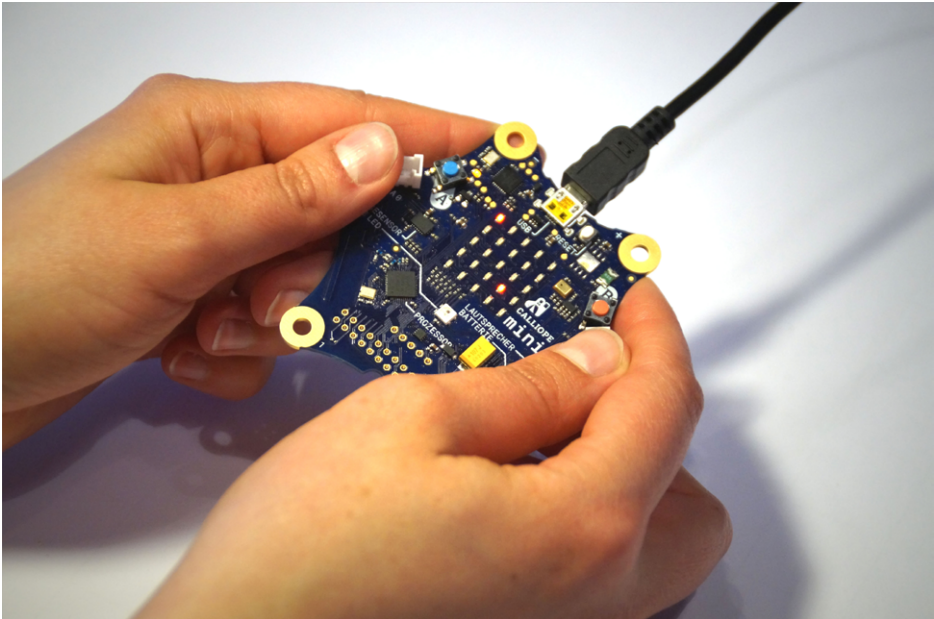


Abb. 8-39 Fang den Dot!

### 8.4.2 Alles noch einmal überprüfen, bevor es losgeht!

Du bist beim letzten Projekt des »Mapping«-Kapitels angekommen und bist jetzt schon ein Profi, was Mapping angeht. Damit du dieses Projekt gut umsetzen kannst, solltest du folgende Calliope mini-Funktionen schon beherrschen:

- ▶ Platzhalter
- ▶ Sensoren (besonders den Beschleunigungssensor)
- ▶ LED-Display
- ▶ Logikvergleiche
- ▶ Zufall
- ▶ Mapping

### 8.4.3 Es geht los

Da dein Spiel dauerhaft funktionieren soll, brauchst du die »dauerhaft«-Schleife. Als Erstes beschäftigst du dich mit einer der Achsen des Calliope mini. In diesem Projekt benutzt du zuerst die x-Achse. Beim Kippen der x-Achse soll der Dot sich auf dem Display genau dann nach links bewegen, wenn der Calliope mini nach links gekippt wird. Für die rechte Seite willst du das genauso realisieren. Dazu brauchst du den »verteile«-Programmierbefehl, einen neuen Platzhalter »rX«, der das Ergebnis speichert, und die Daten des Beschleunigungssensors für die x-Achse.

1. Lege einen neuen Platzhalter »rX« an und ändere den Wert des Platzhalters auf die verteilten Werte des Beschleunigungssensors der x-Achse.

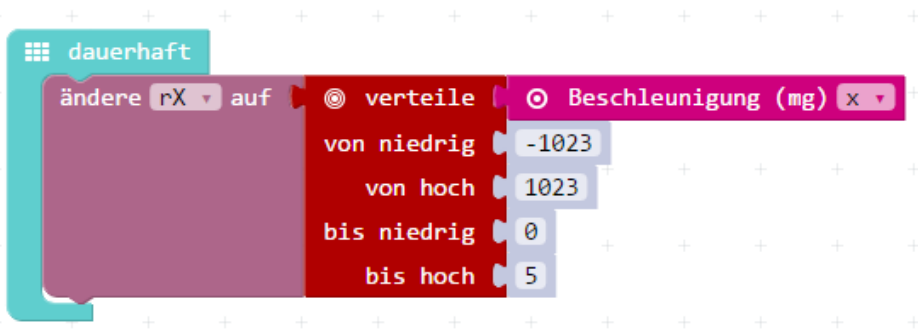


Abb. 8–40 Mapping der x-Achse

Der Beschleunigungssensor misst von -1023 bis 1023. Das LED-Display hat in jede Richtung fünf Lampen.

2. Mappe die Daten des Beschleunigungssensors der x-Achse auf die Anzahl der Lampen von links nach rechts auf dem LED-Display.

## Hinweis

Denk daran, dass man in der Informatik mit der 0 zu zählen anfängt und dass deshalb die erste Lampe des LED-Displays immer die Zahl 0 hat und nicht 1.

Da du aber in dem Spiel den Dot nicht nur in eine Richtung (von rechts nach links) steuern möchtest, sondern auch von oben nach unten, brauchst du eine zweite Achse. Wie du schon in den vorangegangenen Kapiteln lernen konntest, kann der Calliope mini über den Beschleunigungssensor drei Achsen auslesen. Über die y-Achse kann der Calliope mini ein Kippen nach vorne und nach hinten messen.

3. Lege einen neuen Platzhalter für die y-Achse an und nenne ihn »rY«.

Wenn du dir den letzten Programmblock mit dem Mapping der x-Achse ansiehst, kannst du erkennen, dass sich sehr viel mit dem Verhalten des Spiels auf der y-Achse ähnelt. Die Unterschiede sind zum einen der Platzhalter »rY« statt »rX« und zum anderen, dass der Beschleunigungssensor nicht die Daten der x-Achse, sondern die Daten der y-Achse herausgeben soll.

Da sich der Programmcode so stark ähnelt, kannst du ihn einfach duplizieren und dann anpassen.

4. Klicke mit der rechten Maustaste auf den »ändere rX auf«-Befehl und klicke anschließend in dem Menü, das nun auftaucht, auf »Duplizieren«.

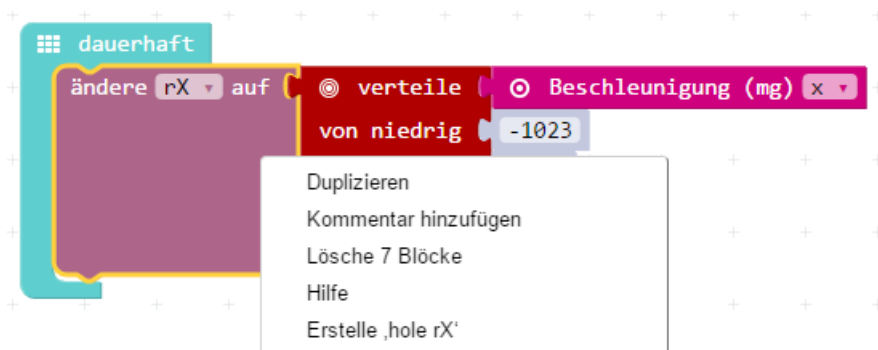


Abb. 8–41 Duplizieren des ersten Programmblocks zum Mapping der x-Achse

5. Pass den zweiten Programmblock für die y-Achse an und setz ihn unterhalb des Programmblocks für die x-Achse.

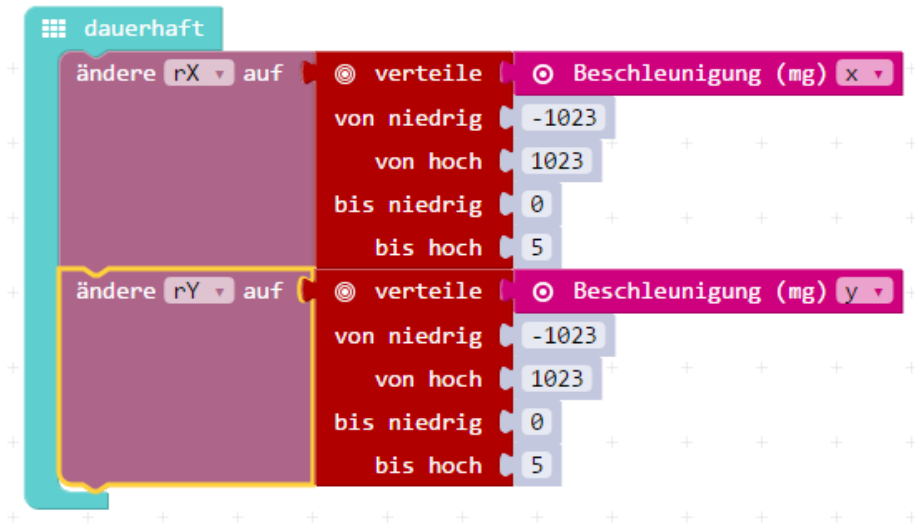


Abb. 8–42 Mapping der x-Achse und der y-Achse

Wenn du den Calliope mini jetzt in die verschiedenen Richtungen kippst, werden die Werte des Beschleunigungssensors von der x-Achse und der y-Achse auf die Ziffern 0 bis 4 verteilt. Jetzt musst du die Anzeige des Dots und die Bewegung des Dots daran anpassen.

Den Dot lässt du erscheinen, indem du den »Zeichne x y«-Befehl benutzt, der sich immer entsprechend der aktuellen Lage deines Calliope mini bewegt.

6. Zieh den »Zeichne x y«-Befehl unter den y-Achse-Programmblock und zieh für den x-Wert den Platzhalter »rX« sowie für den y-Wert den Platzhalter »rY« in den »Zeichne«-Programmierbefehl.

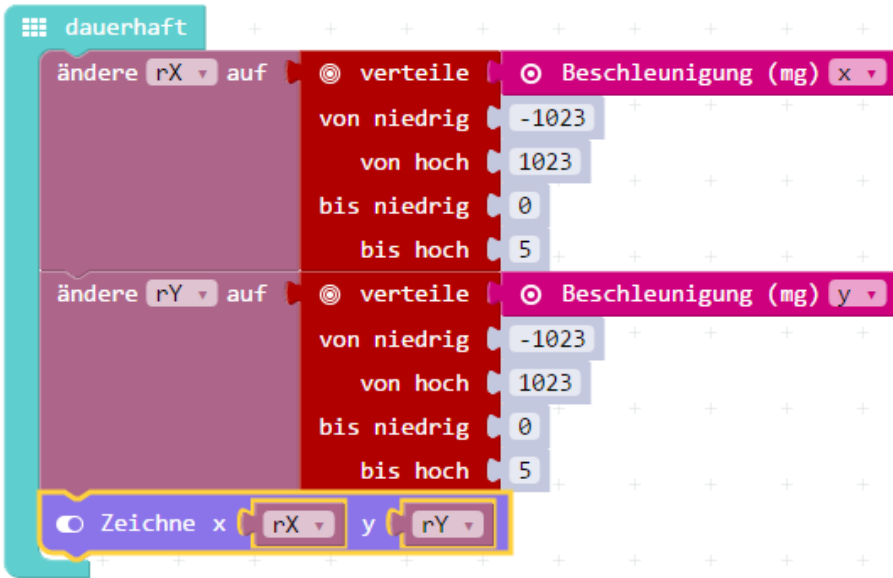


Abb. 8-43 Dot auf dem Display laufen lassen

Sehr gut! Dein Dot bewegt sich jetzt auf dem Display in alle Richtungen. Aber wie du beim Testen merkst, sind schnell alle Lampen des Displays an und du kannst nicht mehr sehen, wo der Dot sich gerade befindet.

Der Grund ist folgender: Nachdem der Dot auf einem Platz war und dort die Lampe angegangen ist, wird die Lampe nicht mehr ausgeschaltet, wenn der Dot sich in eine andere Richtung bewegt.

Das Ausschalten der Lampe kannst du über den Programmierbefehl »schalte Pixel x y« erreichen. Dieser Befehl schaltet eine Lampe des Displays (wird auch Pixel genannt) an einer bestimmten Stelle aus.

7. Zieh den »schalte Pixel«-Befehl an den Anfang der »dauerhaft«-Schleife und ergänze die »x«- und »y«-Stelle durch deine beiden Platzhalter für die beiden Achsen.

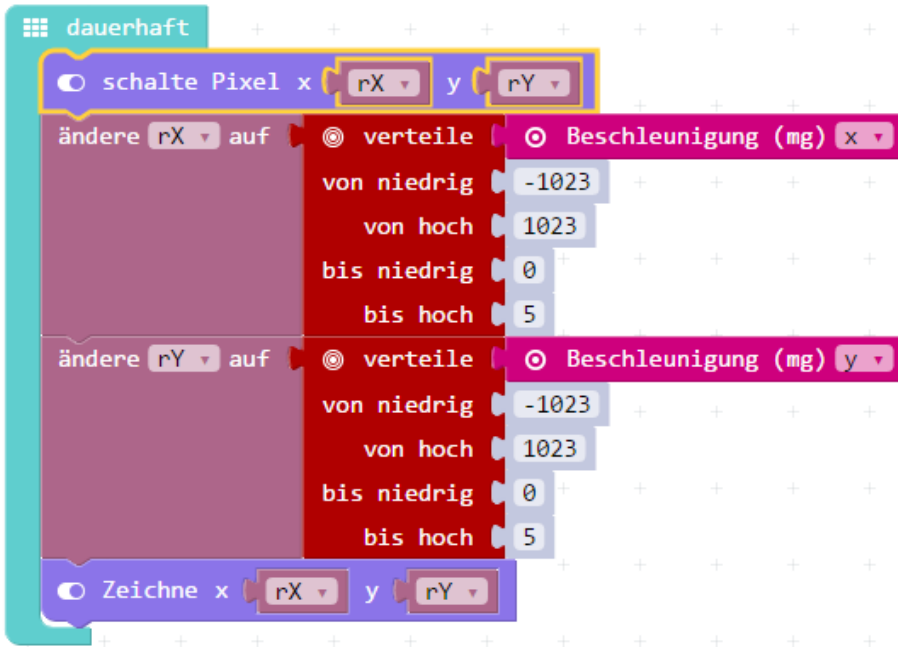


Abb. 8–44 Dot allein auf dem Display

Jetzt läuft dein Dot schon allein über das Display. Toll, damit hast du schon einen Großteil der Spielfunktionen umgesetzt. Jetzt brauchst du noch einen zweiten Dot auf dem Display, damit dein Dot zu diesem Punkt gehen kann, um dort den anderen Dot zu fangen. Dieser zweite Dot soll direkt beim Start des Programms sichtbar sein und an einer zufälligen Stelle auf dem Display auftauchen.

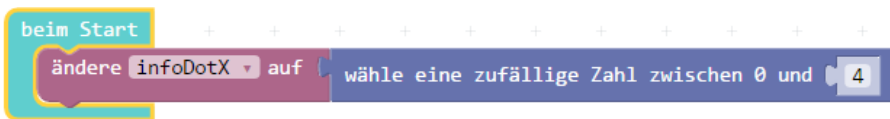


Abb. 8–45 Beim Start wird eine zufällige x-Position gewählt.

8. Benutze den »beim Start«-Programmblock und ändere einen neuen Platzhalter mit dem Namen »infoDotX« auf eine zufällige Zahl zwischen 0 und 4.

Genauso, wie du den Programmblock eben duplizieren konntest, kannst du dies auch an dieser Stelle für den y-Wert tun.

9. Leg einen neuen Platzhalter »infoDotY« an. Dupliziere dann den ersten Programmblock und ändere den Platzhalter auf »infoDotY«.

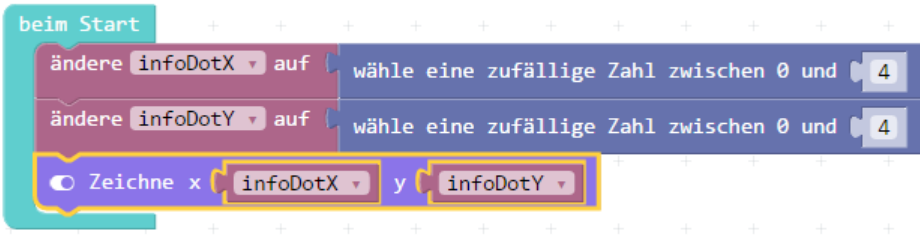


Abb. 8–46 Zeichne einen zweiten Dot an einer zufälligen Position beim Start

Jetzt fehlt nur noch das Zeichnen des neuen Dots an der zufälligen Stelle. Dies kannst du genauso wie beim ersten Dot machen.

10. Benutz den »Zeichne x y«-Befehl, um den neuen Dot an der zufälligen Position anzuzeigen.

Gut, du hast nun einen zufälligen Dot auf dem Display beim Start und kannst dich mit dem ersten Dot auf dem Display bewegen. Jetzt muss nur noch etwas passieren, wenn der erste Dot den zweiten Dot fängt. Dazu musst du wieder in die »dauerhaft«-Schleife deines Programms schauen.

Da du nur etwas machen willst, wenn der erste Dot den zweiten Dot erreicht, kannst du den »wenn-dann«-Programmblock benutzen.

11. Zieh den »wenn-dann«-Programmierblock unterhalb des »Zeichne x y«-Befehls in die »dauerhaft«-Schleife.



Abb. 8-47 »wenn-dann«-Programmierblock in der »dauerhaft«-Schleife

Die spannende Frage ist nun, wann du weißt, dass sich der erste Dot und der zweite Dot auf der gleichen Stelle des Displays befinden. Dies ist dann der Fall, wenn der x-Wert des ersten Dots und der x-Wert des zweiten Dots gleich sind und gleichzeitig der y-Wert des ersten Dots und der y-Wert des zweiten Dots gleich sind.

Genauso kannst du dies jetzt auch in deinem Programmcode ausdrücken. Als Erstes brauchst du den »und«-Block, indem danach die x-Werte und y-Werte verglichen werden.

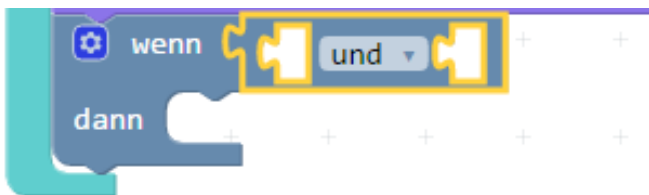


Abb. 8-48 »und«-Programmblock im »wenn-dann«-Programmblock

12. Zieh einen »und«-Block an das »wenn« des »wenn-dann«-Blocks.

Jetzt willst du die x-Werte und y-Werte der beiden Dots vergleichen. Dazu brauchst du zwei »gleich«-Programmierbefehle.

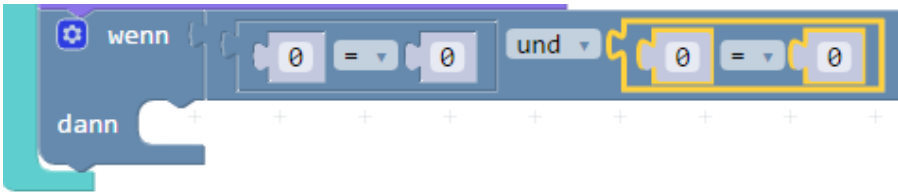


Abb. 8-49 Zwei »gleich«-Programmierbefehle

13. Zieh zwei »gleich«-Programmierblöcke hinter und vor das »und« im »und«-Block.

Jetzt brauchst du noch zwei Platzhalter für die x-Werte sowie die beiden Platzhalter für die y-Werte. Dann kannst du die x-Werte miteinander vergleichen und genauso auch die y-Werte. Wenn beide gleich sind, sind beide Dots auf derselben Position auf dem Display.

14. Vergleiche als Erstes die x-Werte der beiden Dots »rX« und »infoDotX«. Anschließend machst du das Gleiche für die y-Werte.

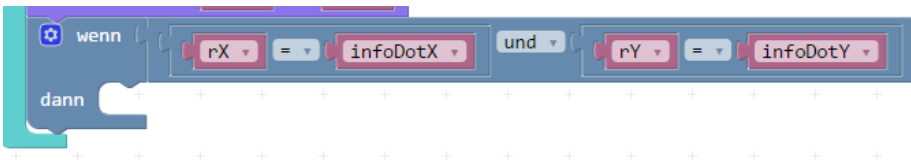


Abb. 8-50 Vergleich der Positionen der beiden Dots

Jetzt musst du den »dann«-Block noch füllen. Wenn der erste Dot auf der Position des zweiten Dots angelangt ist, soll der zweite Dot sich eine neue zufällige Position suchen. Erinnerst du dich, wo du diesen Block schon einmal programmiert hast? Genau! Im »beim-Start«-Programmblock.

Du musst also die Programmblöcke im »beim-Start«-Programmblock nur duplizieren und in den »dann«-Teil des »wenn-dann«-Programmblocks einfügen.

15. Dupliziere alle Programmblöcke aus dem »beim-Start«-Programmblock, sodass sich der zweite Dot eine neue Position sucht, wenn der erste Dot den zweiten Dot erreicht hat.

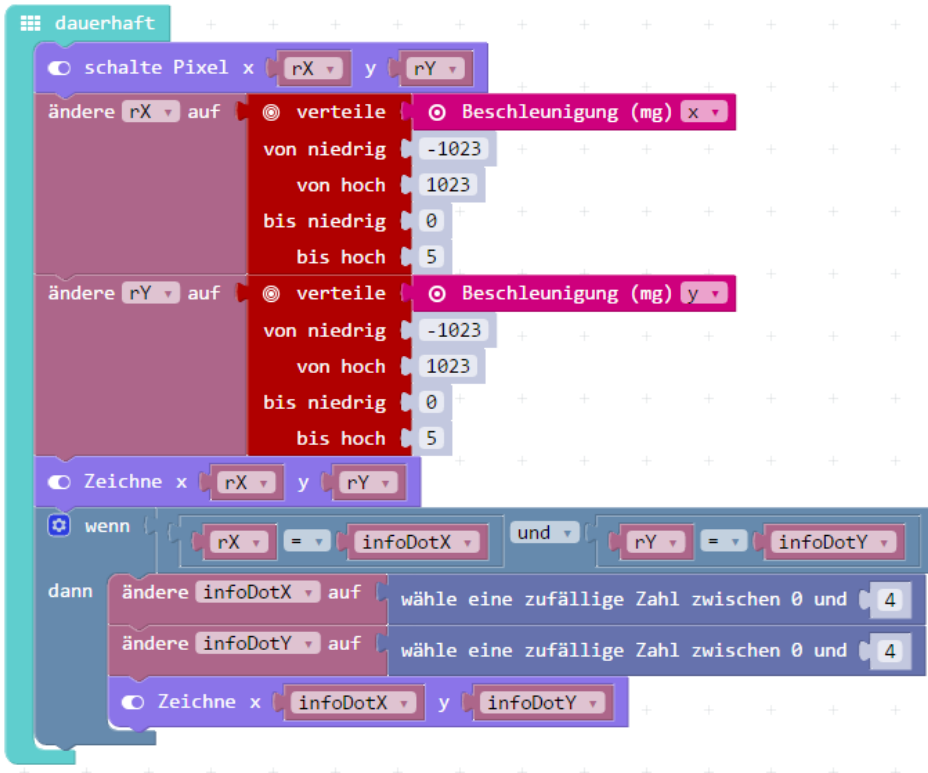


Abb. 8–51 Der zweite Dot wird gefangen und sucht sich ein neues Versteck.

Super! Das Spiel funktioniert jetzt schon. Du kannst das Spiel jetzt noch auf viele Arten erweitern. Schau dazu am Ende des Kapitels nach oder setz eigene Ideen um.

## 8.4.4 Testen

Macht dein Spiel genau das, was es soll? Teste es erst einmal ausgiebig und pass die Werte für den Bewegungssensor so an, dass beim Kippen des Calliope mini der Dot sich gut bewegen kann und es dein Spiel nicht stört, wenn du den Calliope mini zu viel kippen musst.

Denk immer daran: Kleine Fehler schleichen sich immer mal wieder ein. Wenn dein Spiel sich nicht so verhält, wie du es willst, schau dir die einzelnen Programmblöcke in diesem Buch noch einmal an und vergleiche sie mit deinen Programmblöcken.

## 8.4.5 Da geht noch mehr?! – Optimierte dein Projekt

Hier findest du weitere Ideen, wie du dein Projekt verbessern kannst. Das Spiel bietet noch sehr viele Erweiterungsmöglichkeiten.

- ▶ Du kannst viel mit den Wertebereichen spielen und das Verhalten des Dots verändern oder sogar verschiedene Schwierigkeitsgrade einbauen.
- ▶ Wenn ein Spieler dein Spiel spielt und ein Dot gefangen wurde, kannst du zum Beispiel die Punkte zählen und dann auf dem Display ausgeben lassen.
- ▶ Ergänze die Punktausgabe durch ein »YEAH«, wenn ein Dot gefangen wurde.
- ▶ Du kannst auch einen Countdown für den Beginn des Spiels einbauen oder sogar einem Spieler nur eine bestimmte Zeit lassen, bis wann er den zweiten Dot gefangen haben muss.

# 9

## Drahtlose Kommunikation

Über drahtlose Kommunikation ist es möglich, dass Calliope-Boards untereinander Nachrichten austauschen, ohne dass sie dafür mit einem Kabel am Computer verbunden sein müssen.

Die Einführung zu diesem Kapitel hat Julia Kleeberger geschrieben. »Morsen mit mehreren Calliope minis« ist von Patrick Franken, »Alarmanlage vernetzt« von Jan Thar. »Hau den Lukas« wurde von Mario Lukas geschrieben und »Das Roboterballett« von Julia Kleeberger. »Eckstein, Eckstein ... alles muss versteckt sein« ist von Mario Lukas und »Theremin« stammt von Jan Thar.

Bisher hast du deinen Calliope mini immer mit dem Computer verbunden, damit er Anweisungen in Form von Programmen bekommt, die du geschrieben und heruntergeladen hast. Über drahtlose Kommunikation ist es auch möglich, dass Calliope-Boards untereinander Nachrichten austauschen, ohne dass sie dafür mit einem Kabel am Computer verbunden sein müssen: eben drahtlos.

Wie bei jeder Kommunikation brauchst du auch bei der drahtlosen Kommunikation einen Sender, der ein Signal drahtlos an den Empfänger übermittelt. Für die Übermittlung des Signals werden Funkwellen verwendet. Dein Calliope mini kann über zwei verschiedene Wege drahtlos kommunizieren: über Funk (im Engl. »Radio«) und über Bluetooth. In diesem Kapitel wird dir die Arbeitsweise mit Funk erklärt.

## 9.1 Die Funktionsweise von Funk

Beim Funken sendet der Sender-Calliope das Signal über eine Antenne als Funkwelle in einem bestimmten Frequenzbereich aus. Alle Empfänger-Calliope in der Umgebung können das Funksignal empfangen. Das ist ähnlich wie beim Radio.

### Wie funktioniert Radio?

Eine Radioanstalt sendet sein Programm aus und alle Radiogeräte, die die Frequenz des Radiosenders eingestellt haben, können es empfangen. Man nennt diese Form der Kommunikation auch »one to many«, also ein Sender kann an viele Empfänger gleichzeitig ein Signal versenden. Die ankommende Funkwelle wird bei den Empfängern ebenfalls über eine Antenne aufgenommen, verstärkt und in das ursprüngliche Ausgangssignal zurückgewandelt. Bestimmt ist dir schon mal aufgefallen, dass du dein Lieblingsradiosender im Urlaub evtl. nicht empfangen konntest. Das liegt daran, dass Funkwellen nur eine bestimmte Reichweite haben und nur innerhalb dieser empfangen werden können. Dabei wird das Signal umso besser empfangen, je näher sich der Empfänger an der Sendequelle befindet. Auch Hindernisse wie z.B. Bäume, Wände oder Berge führen dazu, dass die Intensität der Funkwellen abnimmt.

Um diese Funktionsweise mit deinem Calliope mini auszuprobieren, kannst du im PXT-Editor in der Befehlsgruppe »Funk« den Programmierblock »Sende Nummer« auswählen. Mit diesem Befehlsblock wird eine Nummer an alle Empfänger-Calliope-Boards gesendet.

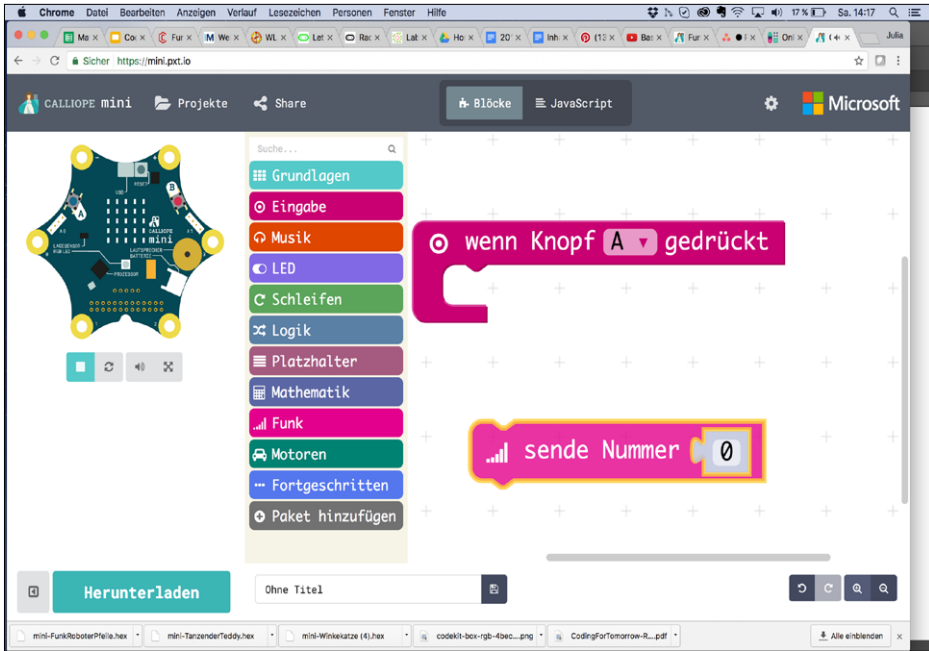


Abb. 9-1 Dieser Befehlsblock überträgt über Funk eine Nummer. Dabei werden alle Calliope minis angesprochen, auf die dieses Programm heruntergeladen wurde.

Kommt ein Datenpaket beim Empfänger an, so kann eine Funktion ausgelöst werden. Hierfür wird folgender Programmierbefehl verwendet:



Abb. 9-2 Dieser Befehlsblock registriert Funktionen, die ausgeführt werden, wenn ein Datenpaket über Funk empfangen wird.

Um diese Funktionsweise zu testen, ist es wichtig, den Programmcode auf alle Boards zu überspielen: sowohl auf den Sender-Calliope als auch auf alle Empfänger-Calliope-Boards.

Wenn du mehrere Sender-Calliope minis einrichten möchtest, die unterschiedliche Inhalte senden, ist es notwendig, Gruppen anzulegen. Damit stellst du sicher, dass bei Boards in derselben Gruppe keine Probleme beim Senden und Empfangen auftreten.

Um Gruppen zu erstellen, wählst du im PXT-Editor die Gruppe der Funk-Befehle aus und klickst dann auf »... Mehr«, damit dir weitere Programmierblöcke angezeigt werden.

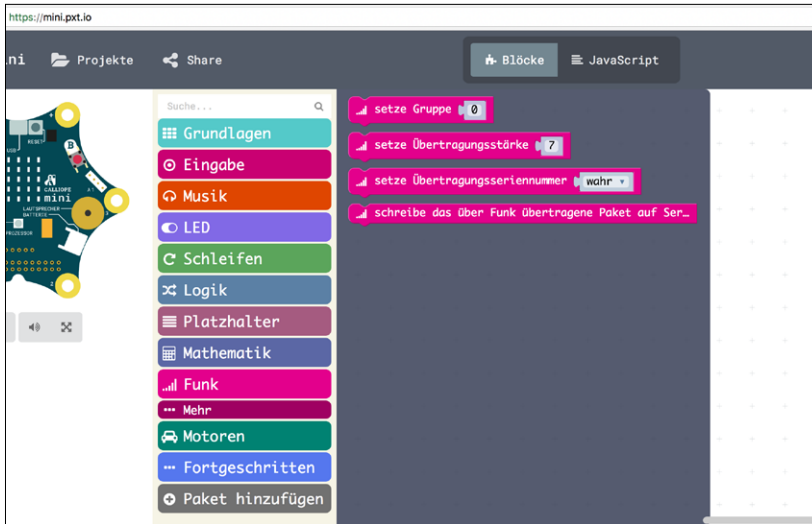


Abb. 9-3 Über die Menügruppe »Funk« → »Mehr« gelangst du zu dem Befehlsblock »Setze Gruppe«

Mit der Funktion »Setze Gruppe« kannst du den Calliope mini unterschiedliche Gruppen zuweisen. Somit hören sie nur auf die Signale einer Gruppe. Wenn du also zwei Sender mit unterschiedlichen Gruppennummern hast, kannst du damit festlegen, von welchem Sender dein Empfänger-Calliope das Signal empfangen soll. Jeder Calliope mini kann nur einer Gruppe zugewiesen werden.

## Die Namensgebung im PXT-Editor

Funkwellen werden im Englischen also »radio waves« bezeichnet. Daher findest du im Editor an manchen Stellen auch die Bezeichnung »radio« anstelle von Funk.

## 9.2 Morsen mit mehreren Calliope minis

Bisher bestanden alle deine Projekte aus nur einem Calliope mini. In diesem Projekt lernst du, dass sich mehrere Calliope auch miteinander unterhalten können. Vielleicht schreibst du dann demnächst nicht mehr über das Handy, sondern mit zwei Calliope? So einfach ist es dann doch nicht, aber in diesem Kapitel lernst du, wie du zwischen zwei Calliope minis Nachrichten hin und her sendest. Dazu werden die im Calliope-Board verbauten Antennen verwendet. Was du sendest, ist dir überlassen, aber damit du Nachrichten versenden kannst, lernst du in diesem

Kapitel das Morsen. Am besten setzt du dieses Projekt mit einem Freund oder einer Freundin um, denn so macht das Morsen am meisten Spaß.

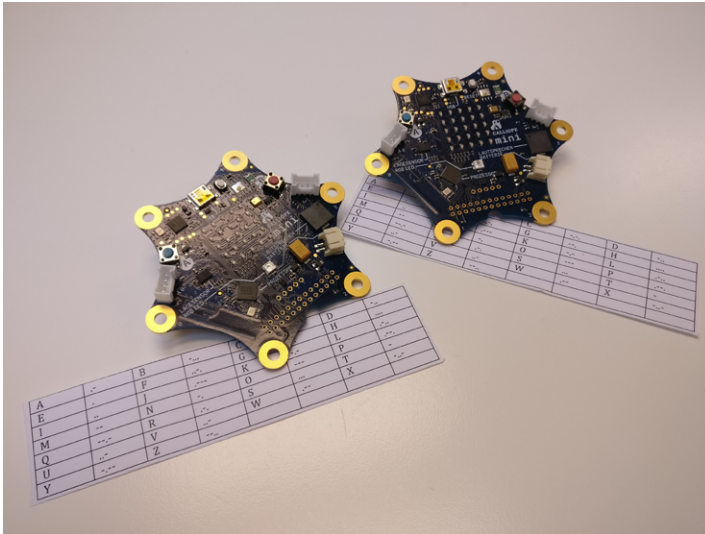


Abb. 9–4 Calliope minis unter sich

## Was ist Morsen?

Früher haben sich Schiffe auf hoher See mit Lichtsignalen verständigt. Jedes Schiff hat eine sehr große Lampe, die von einem anderen Schiff erkannt werden kann. Das andere Schiff hat ebenfalls eine solche Lampe, so können sich die Schiffe dann abwechselnd anleuchten. Es gibt also nur »Licht an« oder eben »Licht aus«. Das ist nicht sehr viel, weshalb sich schlaue Seeleute überlegt haben, wie man mit dieser Technik Nachrichten wie z. B. einen Notruf versenden kann. Als Erstes hat man sich auf zwei Signale geeinigt, ein kurzes und ein langes Leuchten. Dann hat man diese zwei Signale einzelnen Buchstaben zugeordnet. So bekommt der Buchstabe A das Signal, einmal kurz und einmal lang zu leuchten.

A	.-	B	-...	C	-.-	D	..
E	.	F	...-	G	--.	H	....
I	..	J	.-.-	K	-.-	L	.-..
M	--	N	-.	O	---	P	.-.-
Q	---	R	.-.	S	...	T	-.
U	..-	V	...-	W	.-.-	X	-.-
Y	-.-	Z	--..				

Tab. 9–1 Das Morse-Alphabet

Jetzt weißt du auch, weshalb ein Schiff in Seenot SOS funkt. Es sind die Buchstaben für dreimal kurz, dreimal lang und noch dreimal kurz. Das ist auch auf dem Meer nicht zu übersehen, jeder Seemann kennt diese Kombination. Das Morse-Alphabet hilft dir also mit nur zwei Signalen, alle Nachrichten, die du magst, versenden zu können. Du musst nur jeden Buchstaben einzeln in diese Signale übersetzen.

Was sollen deine zwei Calliope minis nun können? Sie sollen sich gegenseitig die zwei Signale »kurzes« und »langes« Leuchten senden können. Bei einem Druck auf Knopf »A« soll ein kurzes Leuchten verwendet werden, bei einem Druck auf Knopf »B« ein langes. Du beginnst mit dem Versenden des kurzen Leuchtens. Dazu startest du wie gewohnt ein neues Projekt und löschst die Beispielblöcke.

Für dieses Morseprogramm verwendest du Blöcke aus der Kategorie »Funk«:

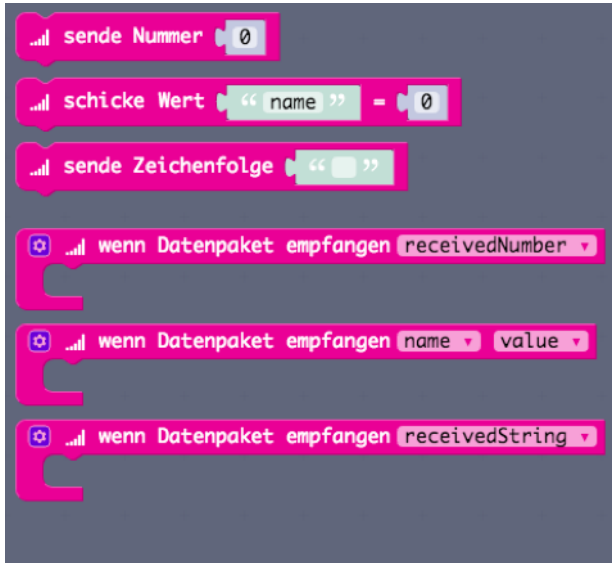


Abb. 9–5 Unter »Funk« findest du alles, um Daten wie Nummern und Texte senden und empfangen zu können.

Um zu starten, verwendest du den »sende Zeichenfolge«-Block. Er soll aufgerufen werden, wenn Knopf »A« gedrückt wird. Für den Anfang programmieren wir das Signal »kurz«.

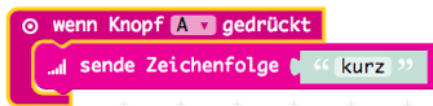


Abb. 9–6 Der Calliope beginnt zu senden

Nun wird bei einem Druck auf Knopf »A« die Nachricht »kurz« an alle anderen Calliope minis innerhalb der Reichweite versandt. Das hilft dir erst einmal nicht viel, solange dein Calliope-Board nicht auch so eine Morse-Nachricht empfangen kann. Also fügst du noch einen weiteren Block hinzu, der diese Aufgabe übernimmt.

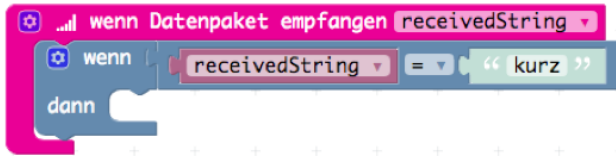


Abb. 9–7 So kannst du auf empfangene Signale reagieren

Der Block »wenn Datenpaket empfangen« wird aufgerufen, sobald dein Calliope mini eine Nachricht von einem anderen Calliope mini erhält. Die empfangenen Daten werden dann im Platzhalter »receivedString« gespeichert. Das ist Englisch und bedeutet »empfangener Text«. Du kannst dies auch umbenennen, verwende dazu den kleinen nach unten zeigenden Pfeil.

Die Verzweigung in diesem Block prüft, ob der empfangene Text dem Text »kurz« entspricht. Jetzt fehlt eigentlich nur noch, dass die LED dann auch kurz leuchtet.



Abb. 9–8 Der Text »kurz« wird empfangen

In diesem Beispiel werden 500 ms, also 500 Millisekunden, gewartet, bis die LED wieder ausgeht. Probier etwas rum und finde den Blinkrhythmus, der dir am meisten gefällt.

Du kannst dein Programm nun bereits testen. Lade es dazu auf beide Calliope minis. Nach einem Druck auf Knopf »A« sollte die LED am anderen Calliope mini kurz aufleuchten. Sollte das erfolgreich gewesen sein, kannst du auch testen, wie weit die Calliope-Boards maximal auseinander sein dürfen, um noch ein Signal vom anderen Board zu empfangen. Wie wirken sich Gegenstände aus, die dazwischen stehen?

Weiter geht es mit dem Signal »lange leuchten«. Du musst also nicht nur prüfen, ob »kurz«, sondern auch ob »lang« empfangen wurde. Dazu kannst du eine

zweite Verzweigung verwenden. Du prüfst dazu zuerst, ob der empfangene Text »kurz« ist, und dann, ob er vielleicht »lang« ist.

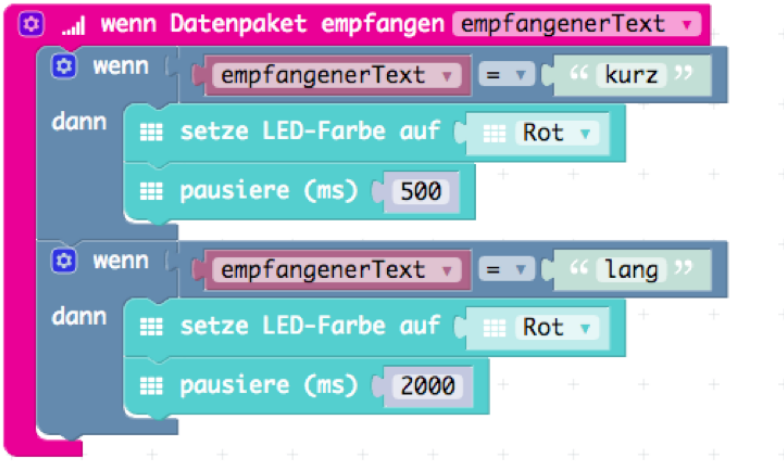


Abb. 9–9 Die vollständige Verzweigung

Nun ist dir vermutlich schon aufgefallen, dass du den Text »lang« noch nirgendwo versendest. Da »kurz« schon auf Knopf »A« liegt, kannst du für »lang« Knopf »B« verwenden. Du kannst also einfach den ganzen »wenn Knopf A gedrückt«-Block kopieren – das kann man auch duplizieren nennen – und dann »A« durch »B« und »kurz« durch »lang« ersetzen. Wenn das jetzt zu schnell ging, schau im Bild unten noch einmal nach.



Abb. 9–10 Fertiger Block für Knopf »B«

Dein Programm ist schon fertig! Das Programm ist so ausgelegt, dass es sowohl senden als auch empfangen kann. Du kannst also auf beide Calliope minis das gleiche Programm spielen und so können beide miteinander morsen. Probier doch einmal aus, das Wort »Hallo« zu versenden. Am besten nimmt die Person am zweiten Calliope mini ein Blatt und einen Stift zur Hilfe, um sich die einzelnen Buchstaben aufzuschreiben. Mithilfe des Morse-Alphabets kannst du das Wort »Hallo« folgendermaßen zusammensetzen:

H	....
A	._
L	.-..
L	.-..
O	---

Tab. 9–2 Das Wort »Hallo« im Morse-Code

Ein Punkt entspricht hier »kurz«, also Knopf »A«, und der Strich steht für »lang«, also Knopf »B«. Dein Gegenüber benötigt vermutlich etwas Zeit, wird aber mithilfe des Morse-Alphabets deine Nachricht entschlüsseln können.

Mit diesem Programm könnt ihr euch Nachrichten schicken, ohne dass andere den Inhalt erkennen können. Versuch doch einfach, deinem Freund oder deiner Freundin eine geheime Nachricht zu schicken.

```
...- ...-.-
...-.-.- ...-.-.-!
```

## 9.3 Alarmanlage vernetzt

Am Anfang des Buches hast du schon einmal eine einfache Alarmanlage mithilfe des Lichtsensors gebaut. Für eine Alarmanlage für deinen Raum fehlt allerdings noch die Möglichkeit, sie nach Verlassen des Raumes anzuschalten beziehungsweise selber wieder abzuschalten, ohne den Alarm auszulösen. Zu diesem Zweck lernst du in diesem Projekt, eine etwas komplexere Alarmanlage zu bauen, für die du zwei Calliope minis benötigst. Einer wird als Sensor an deiner Türklinke verwendet (mit ein paar Streifen Klebeband befestigt), den zweiten hast du als Fernbedienung dabei.



Abb. 9–11 Ein Calliope mini bewacht deine Tür

### 9.3.1 Aufbau

Anstelle des Lichtsensors wirst du hier den Lagesensor zum Auslösen des Alarms verwenden: Durch Herunterdrücken der Türklinke wird der Calliope mini im Raum bewegt, seine Lage wird also verändert und – zack! – wird der Alarm ausgelöst. Im Folgenden wird zunächst die eigentliche Alarmanlage programmiert und anschließend die Fernbedienung für den zweiten Calliope mini.

### 9.3.2 Alarmanlage

Wenn du noch einmal die Alarmanlage aus Abschnitt 4.3 anschaust, kannst du ganz leicht die Lichtstärke durch die Winkelmessung ersetzen:



Abb. 9–12 Alarmanlage mit Winkelmessung

Klappt es? Piepst der Calliope mini, wenn er bewegt wird? Funktioniert das auch an der Türklinke? Vielleicht musst du den Winkel (hier 10°) noch anpassen, damit deine Alarmanlage im richtigen Moment auslöst.

Wenn du jetzt noch eine Funkfernbedienung hinzufügen möchtest, fängst du am besten beim Starten des Calliope mini mit dem Bilden einer Gruppe an:



Abb. 9–13 Gruppe, erste Platzhalter für die Alarmanlage

Zunächst einmal wird »setze Gruppe 1234« festgelegt, damit nur ein Calliope mini, der die gleiche Gruppe, also »1234«, gesetzt hat, sich mit der Alarmanlage

verbinden und sie damit starten oder entschärfen kann. Im Grunde funktioniert das also als Passwort-Code der Alarmanlage, da hier festgelegt wird, welche Calliope minis miteinander reden können. Natürlich ist »1234« nicht besonders kompliziert, vielleicht wählst du besser eine nicht ganz so leicht zu erratende Nummer für deine eigene Alarmanlage aus.

Danach folgen zwei neu erstellte Platzhalter: Einer ist dazu da, zu speichern, ob ein Alarm ausgelöst wurde (»AlarmAktiv«), der zweite ist dazu da, festzulegen, ob die Alarmanlage eingeschaltet ist (»AlarmAngeschaltet«). Anfangs ist sie aus, und kein Alarm soll ertönen, daher ist in beiden Platzhaltern eine 0 gespeichert.

- ▶ »AlarmAngeschaltet«: Wenn eine 0 gespeichert ist, ist die Alarmanlage aus. Wenn eine 1 gespeichert ist, ist die Alarmanlage scharfgestellt, ist also bereit, einen Eindringling zu verscheuchen.
- ▶ »AlarmAktiv«: Zunächst ist hier eine 0 gespeichert. Sobald der Calliope mini bewegt wird und die Alarmanlage scharf ist, wird dieser Wert auf 1 geändert und der Alarm geht los. Er kann dann wieder durch Abschalten der Alarmanlage auf 0 gesetzt werden.

Um eindeutig sehen zu können, ob alles in Ordnung ist, wird zusätzlich beim Starten des Calliope mini kurz »Start« »AN« eingeblendet und die RGB-LED leuchtet gelb.



Abb. 9-14 Empfangen der Kommandos der Fernbedienung

Wenn Werte über Funk empfangen werden, sind das die Kommandos der Fernbedienung. Hier wird eine 1 gesendet, um die Alarmanlage einzuschalten, durch Senden einer 2 wird sie ausgeschaltet. Dazu wird beim Empfangen eines Datenpakets verglichen, welcher Wert gesendet wurde (er ist in »empfangeneNummer« gespeichert), und dann wird die Alarmanlage entsprechend ein- oder ausgeschaltet, indem die Werte von »AlarmAngeschaltet« geändert werden. Zusätzlich wird in beiden Fällen der Alarm abgeschaltet, falls gerade einer aktiv ist (ändere »AlarmAktiv« auf 0). Am Ende wird der Zustand der Alarmanlage wieder durch kurzes Einblenden eines Textes »AN« oder »AUS« sowie durch das Ändern der Farbe der RGB-LED angezeigt.

Jetzt musst du nur noch den Alarmauslöser so weiterentwickeln, dass der Alarm nur dann ausgelöst wird, wenn

1. die Alarmanlage angeschaltet ist und
2. der Calliope mini (um einen Winkel von über 10°) gekippt wird.

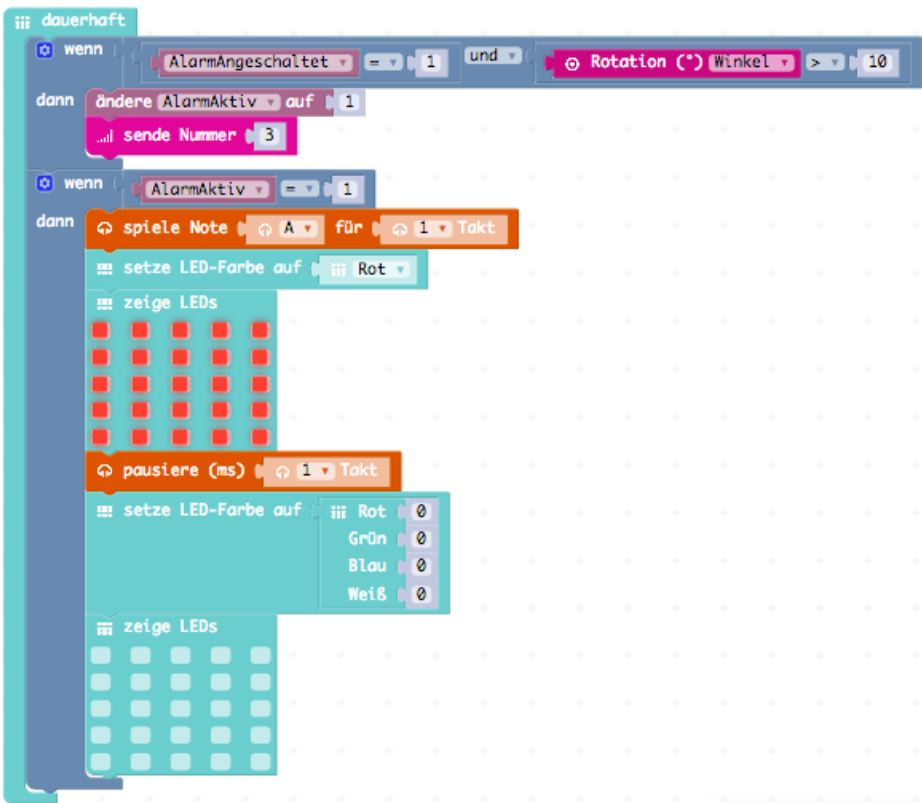


Abb. 9–15 Der Alarmauslöser

Während der Calliope mini läuft, überprüft er ständig den Zustand des Winkelsensors. Ist der Winkel größer als 10° und der Alarm angeschaltet, so wird der

Alarm ausgelöst (in der ersten »wenn-dann«-Entscheidung). In diesem Fall wird der Platzhalter »AlarmAktiv« auf 1 gesetzt, sodass der Alarm auf deinem Calliope mini losgeht und zusätzlich eine Warnmeldung an die Fernbedienung gesendet wird (hier eine 3, aber auch andere Zahlen sind natürlich möglich), damit auch die Fernbedienung weiß, dass eingebrochen wurde.

Nach dieser ersten Entscheidung überprüft der Calliope mini, ob dieser Wert (also »AlarmAktiv«) gleich 1 ist. Ist das der Fall, wird weiterhin wie bei der ersten Alarmanlage eine Note gespielt und dazwischen pausiert, zusätzlich werden aber noch die LED-Anzeige und die RGB-LED einmal an- und wieder ausgeschaltet. Die RGB-LED leuchtet natürlich auch in Rot, wie es sich für einen Alarm gehört. Zum Abschalten der RGB-LED werden beim Setzen der Farbe alle drei Farbwerte zusammen mit Weiß auf 0 gesetzt, sodass die LED wieder dunkel ist.

Jetzt ist es an der Zeit, deine Alarmanlage zu testen. Bevor du sie an der Türklinke anbringst, kannst du (durch Kippen des Calliope mini) schon einmal per Hand den Alarm auslösen. Funktioniert deine Alarmanlage? Hörst und siehst du das Signal, wenn du den Calliope mini kippst?

Wenn alles klappt, kannst du den Calliope mini mit einem Battery-Pack (zum Beispiel mit Klebeband) an der Türklinke befestigen, sodass beim Herunterdrücken der Alarm ausgelöst wird. Wenn es nicht gerade genug auf der Klinke liegt, kann man es mit etwas Material an einer Seite unterfüttern oder aber den Winkel, ab dem der Alarm losgeht (hier die  $10^\circ$ ), noch etwas vergrößern.

So, die Alarmanlage funktioniert schon einmal und verschreckt sicher den Eindringling. Damit die Alarmanlage nicht losgeht, wenn du selbst einmal die Türklinke herunterdrückst, kannst du sie nun mit einer Fernbedienung steuern.

### 9.3.3 Fernbedienung

Die Fernbedienung soll:

- ▶ die Alarmanlage anschalten können,
- ▶ die Alarmanlage ausschalten und
- ▶ und zusätzlich anzeigen, wenn ein Alarm ausgelöst wird.

Das Letztere ist nützlich, wenn du selbst nicht in Hörweite des Alarms bist und trotzdem mitbekommen möchtest, wenn jemand unerlaubt in dein Zimmer eindringt.

Für die Fernbedienung werden die beiden Knöpfe des Calliope mini als Ein- und Ausschalter für den Alarm verwendet. Wenn man aber eh schon eine Vernetzung mit der Alarmanlage hat, kann man auch gleich Einbrüche mit der Fernbedienung anzeigen lassen.

Hierfür programmierst du für den zweiten Calliope mini ein anderes Programm.



Abb. 9–16 Fernbedienung: Wähle die gleiche Gruppe wie die Alarmanlage.

Beim Starten des Fernbedienungs-Calliope passiert noch nichts besonders Aufregendes: Einmal wird die Zeichenfolge »Start« angezeigt. Wichtig ist aber vor allem, dass der Calliope mini in derselben Gruppe wie der Calliope mini der Alarmanlage ist. Wie vorher beschrieben, kann auch eine etwas einfallsreichere Nummer anstelle von »1234« gewählt werden.



Abb. 9–17 Fernbedienung: Knopf »A« zum Anschalten ...

Beim Drücken von Knopf »A« muss nun das passende Kommando an den zweiten Calliope mini gesendet werden: Es wird zunächst die passende Zahl zum Einschalten an die Alarmanlage gesendet, wodurch der Calliope mini beim Empfangen der Zahl die Alarmanlage aktiviert. Danach wird mit den folgenden zwei Programmierbefehlen durch eine Anzeige der Zeichenfolge »AN« und Ändern der Farbe der RGB-LED auf »Gelb« signalisiert, dass die Alarmanlage eingeschaltet ist.

Probier es aus! Schaffst du es, die Alarmanlage mit der Fernbedienung anzuschalten? Löst das dann beim Kippen des anderen Calliope mini den Alarm aus? Ja, aber er hört gar nicht mehr auf? Dann wird es Zeit, auch eine Möglichkeit zum Ausschalten der Alarmanlage zu programmieren.



Abb. 9–18 Fernbedienung: ... und Knopf »B« zum Abschalten

Bei einem Druck auf Knopf »B« geschieht im Grunde das Gleiche, nur dass eine andere Zahl zum Ausschalten der Alarmanlage gesendet wird. Danach wird der Text »Aus« angezeigt und LED-Farbe auf »Grün« umgestellt.

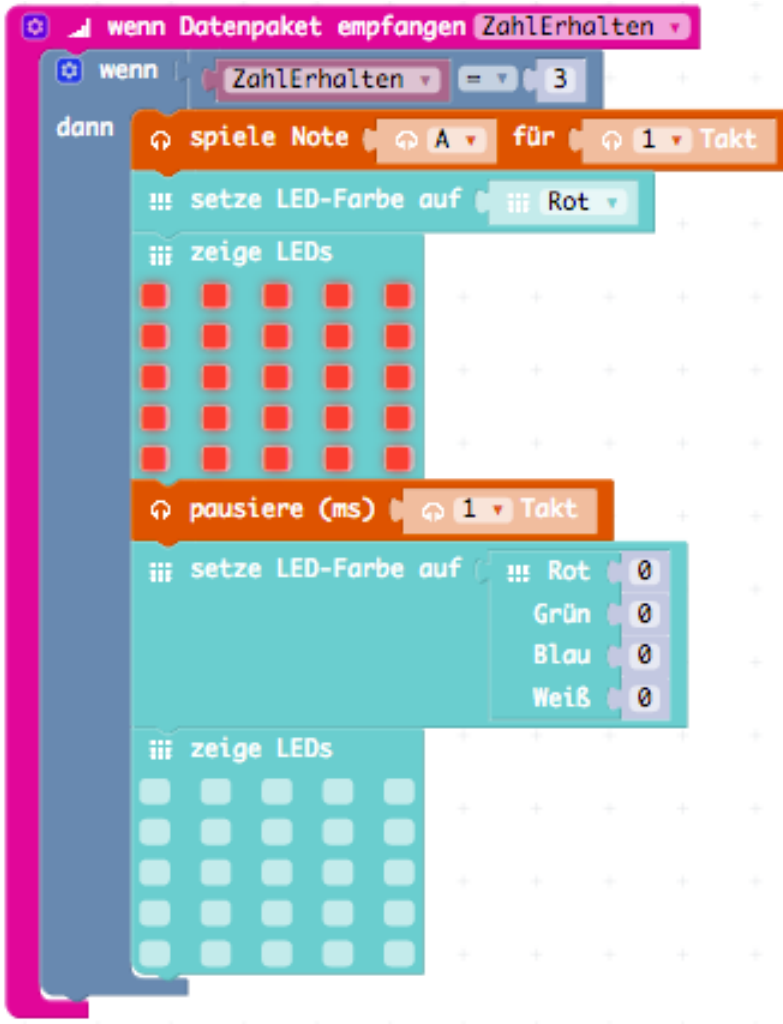


Abb. 9–19 Fernbedienung: Alarmanzeige

Für die Anzeige eines Alarms kannst du mehr oder weniger den Alarmteil der Alarmanlage kopieren, nur dass das Abspielen der Töne und das Blinken von LED-Matrix und RGB-LED dann durchgeführt wird, wenn der Calliope mini etwas von der Alarmanlage empfängt (hier die Zahl 3). In diesem Beispiel wird zwar auch noch überprüft, ob der empfangene Wert wirklich 3 ist, die Alarmanlage senden sollte. Da das aber der einzige Wert ist, den sie sendet, könnte man das genauso gut auch weglassen.

### 9.3.4 Erweiterungen

Bei der Fernbedienung wird der Ton nur einmal beim Auslösen des Alarms abgespielt, danach herrscht wieder Ruhe. Wenn der Ton dauerhaft ertönen soll, musst du ihn in die »dauerhaft«-Schleife einbauen. Schaffst du das?

Das Festlegen der Gruppe entspricht der Sicherheit deiner Alarmanlage – je mehr Ziffern, desto besser: Wenn du nur die Gruppe 0 für deine beiden Calliope-Boards wählst, könnte jemand rein zufällig – oder auch absichtlich – eine Fernbedienung mit der gleichen Gruppe bauen und damit deine Fernbedienung entschärfen, ohne dass du es willst. Die »1234« ist auch nicht so toll, aber bei so etwas wie »35437« kann es schon etwas länger dauern, bis ein Einbrecher herausfindet, mit welcher Gruppe er seinen Calliope programmieren muss, um deine Alarmanlage zu entschärfen.

Was zu einem weiteren möglichen Projekt führt: Lass einen Freund einen Calliope mini mit einer bestimmten (kleinen Zahl) als Gruppe als Alarmanlage programmieren – schaffst du es sie zu knacken, das heißt die richtige Gruppe herauszufinden?

#### Tipps für das erfolgreiche Knacken

Für die Nummer der Gruppe kannst du einen Platzhalter anlegen. In dieses kannst du zuerst die Zahl »0« speichern und ausprobieren, ob du (mit der Gruppennummer 0) die Alarmanlage deines Freundes ausschalten kannst. Falls es nicht geklappt hat, zählst du den Platzhalter um 1 hoch und probierst es erneut. Wenn du das ganze einprogrammierst, testet dein Calliope mini automatisch alle Gruppennummern durch. Kannst du so die Alarmanlage knacken, also eine Alarmanlage mit dem Calliope ausschalten, deren Gruppe du gar nicht kennst?

Viel Spaß beim Bau deiner Alarmanlage und der Fernbedienung dafür!

## 9.4 Hau den Lukas

Ein altes Jahrmarktspiel ist »Hau den Lukas«, wo man versucht, mit möglichst viel Kraft mit einem Hammer auf einen Sensor zu schlagen, wobei die Schlagstärke gemessen und durch eine Reihe von Leuchten an einer Stange angezeigt wird. Da man die Calliope minis gut vernetzen kann, kannst du dich auch direkt mit deinen Freunden messen und schauen, wer das Spiel gewinnt.

Allerdings gibt es leider keinen verwendbaren Kraftsensor auf dem Calliope mini – oder auch zum Glück, denn wenn man das Spiel wie auf der Kirmes mit einem Hammer spielen würde, würde das deinem Calliope mini auf Dauer nicht gut tun. Was es gibt, ist ein Beschleunigungssensor; damit kann man etwas Ähnliches erreichen.

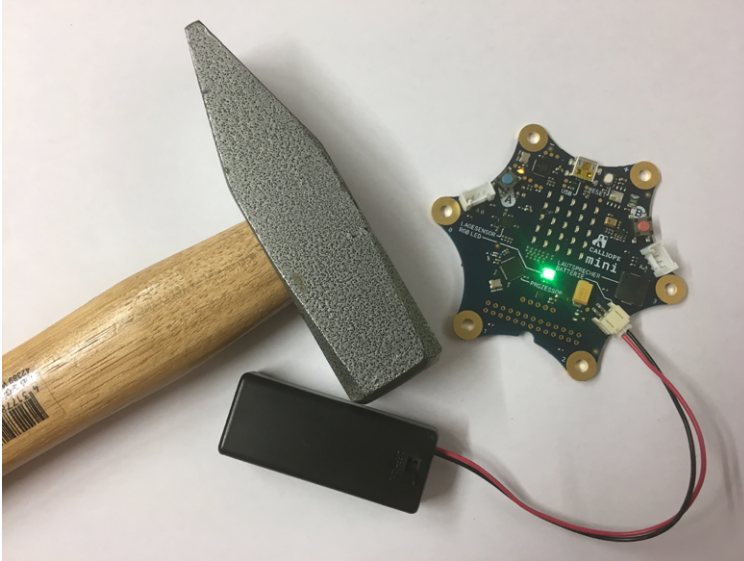


Abb. 9–20 Hau den Lukas

## 9.4.1 Grundprinzip

Für das »Hau den Lukas«-Spiel auf dem Calliope mini wirst du den Beschleunigungssensor nutzen. Der Spielablauf soll so aussehen:

- ▶ Ein Spieler startet das Spiel durch Knopfdruck.
- ▶ Für eine bestimmte Zeit – angezeigt durch die RGB-LED – soll jeder Spieler seinen Calliope mini möglichst wild bewegen.
- ▶ Nach Ablauf der Zeit wird die gemessene Beschleunigung jedes Spielers auf seinem Calliope mini als Zahlenwert angezeigt und danach an die Mitspieler gesendet.
- ▶ Der Spieler mit dem höchsten Beschleunigungswert hat gewonnen (bei ihm leuchtet die RGB-LED grün). Die anderen Spieler haben verloren (bei ihnen leuchtet die RGB-LED rot).
- ▶ Danach kann per Knopfdruck die nächste Runde gestartet werden.

## 9.4.2 Die Programmierung

Schau dir den Spielablauf etwas genauer an: Zum einen muss also jeder Spieler wissen, wie lange das Spiel dauert. Außerdem muss irgendwo festgelegt werden, ob ein Spiel gerade gestartet wurde. Die Kommunikation selber funktioniert so: Wird der Knopf gedrückt, muss eine Nachricht an alle Spieler gesendet werden,

und nach Ablauf des Spiels noch einmal. Danach kann ausgewertet werden, wer gewonnen hat. Jeder Spieler kann dabei das Spiel kontrollieren, also starten.

```

beim Start
  zeige Zeichenfolge " Hau den Lukas "
  setze Bewegungsmesser auf 8g
  ändere startSpiel auf 0
  ändere SpielDauer auf 1000
  
```

Abb. 9–21 Der Calliope startet das Spiel »Hau den Lukas«

Beim Starten des Calliope mini wird der Titel des Spiels genannt (damit du weißt, dass das richtige Programm auf dem Calliope mini läuft), danach wird der Bewegungsmesser auf einen möglichst großen Wert gestellt (»8g« bedeutet achtfache Erdbeschleunigung). Wenn er zum Beispiel auf 1g stehen würde, würde er maximal die Erdanziehung messen können. Das heißt, die effektivste Methode, das Spiel zu gewinnen, wäre, den Calliope mini einfach richtig herum hinzulegen – und das wäre etwas langweilig.

Danach werden zwei neue Platzhalter eingeführt und Werte darin gespeichert: Zum einen der Platzhalter »startSpiel«, um intern speichern zu können, ob das Spiel begonnen hat (dazu wird dann der Platzhalter auf »1« geändert). Zum anderen der Platzhalter »SpielDauer«, das bedeutet die Zeitdauer, wie lange man den Calliope mini während eines Spiels bewegen sollte. Dies sind hier 1000 ms, also eine Sekunde (das m steht für milli, 1000 Millisekunden sind eine Sekunde). Dieser Wert wird im Programm selber nicht verändert. Trotzdem ist es sinnvoll, einen Platzhalter zu verwenden. So kannst du jederzeit an einer Stelle diese Zeit abändern, falls die eine Sekunde zu kurz oder lang ist.

```

wenn Knopf A gedrückt
  wenn startSpiel = 0
  dann
    ändere startSpiel auf 1
    sende Nummer 0
    ändere Gewonnen auf 3
    ändere momentaneStaerke auf 0
    ändere Zeitstempel auf Laufzeit (ms) + SpielDauer
    setze LED-Farbe auf Gelb
  
```

Abb. 9–22 Starten des Spiels

Das Spiel soll beginnen, sobald ein Spieler Knopf »A« drückt. Hierzu wählst du die »wenn Knopf A gedrückt«-Funktion aus. Dabei soll das Spiel nur dann gestartet werden, wenn es nicht eh schon läuft. Dafür ist im Inneren der Funktion noch ein »wenn-dann«-Befehl eingebaut, der prüft, ob das Spiel noch nicht gestartet wurde (also »startSpiel« = 0 ist), ansonsten passiert durch einen Knopfdruck gar nichts. Ist das Spiel noch nicht gestartet, werden die weiteren Befehle ausgeführt:

- ▶ Zunächst wird »startSpiel« auf 1 gesetzt, um zu speichern, dass das Spiel gestartet wurde.
- ▶ Danach wird eine 0 an die anderen Calliope minis gesendet, damit auch sie wissen, dass das Spiel beginnt. Die 0 benutzt du am besten, da du nach dem Ende eines Spiels auch die Beschleunigungswerte sendest. Da diese Werte nicht 0 sein werden, kann der Calliope so später ganz einfach unterscheiden, ob nun das Signal zum Starten des Spiels gesendet wurde oder halt Beschleunigungswerte empfangen wurden.
- ▶ Anschließend werden drei weitere Platzhalter eingeführt: »Gewonnen« ist dafür da, nachher festzulegen, ob man das Spiel gewonnen hat oder nicht. Welchen Zahlenwert du genau für welchen Zustand festlegst, ist im Grunde egal. Die hier gespeicherte 3 steht hier für ein »Man weiß es noch nicht«. Wenn du lieber eine andere Zahl dafür verwenden willst, kannst du das natürlich auch machen. Dann musst du die 3 allerdings später überall da, wo sie in diesem Programm noch einmal beim Platzhalter »Gewonnen« auftaucht, durch deine andere Zahl ersetzen.
- ▶ In »momentaneStaerke« werden für jede Runde nach und nach die Beschleunigungswerte während des Spiels aufsummiert, für jedes neue Spiel wird wieder bei 0 begonnen.
- ▶ Der »Zeitstempel« gibt schließlich den Zeitpunkt an, an dem das Spiel beendet wird, was der momentane Zeitpunkt plus Spieldauer ist.
- ▶ Abschließend wird die RGB-LED auf »Gelb« gesetzt, um zu zeigen, dass das Spiel begonnen hat und geschüttelt werden soll.

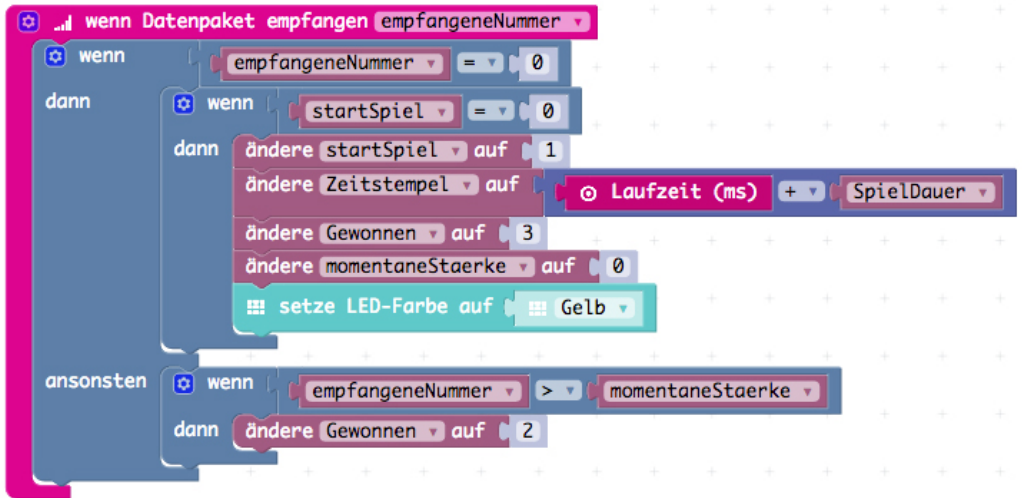


Abb. 9–23 Empfangen von Werten

## Laufzeit

Die Laufzeit ist die interne Uhr des Calliope mini. Sobald der Calliope gestartet wird, fängt die Uhr an zu zählen. Jede Tausendstel Sekunde (kurz ms) wird der in »Laufzeit« gespeicherte Wert um 1 hochgezählt. Damit weiß man genau, wie viel Zeit seit Start des Calliope mini vergangen ist. Eine Umrechnung in eine normale Uhrzeit wäre also nur möglich, wenn man die genaue Uhrzeit kennt, zu der der Calliope mini gestartet wurde.

Beim Empfangen der Datenpakete müssen zwei Fälle unterschieden werden: Zum einen gibt es eine empfangene 0. Sie zeigt an, dass jemand anderes ein Spiel gestartet hat. In diesem Fall werden dieselben Sachen ausgeführt wie beim Knopfdruck, um das Spiel zu starten. Zum anderen soll nach dem Beenden eines Spiels die »momentaneStaerke« von allen Calliope minis an alle anderen gesendet werden – damit reicht hier eine einfache Fallunterscheidung mittels des Befehls »wenn ... dann ... ansonsten« aus.

Wenn du also Werte größer als 0 empfangst, wird der »ansonsten«-Teil ausgeführt und der empfangene Wert wird einfach mit dem eigenen gespeicherten Stärkewert (im Platzhalter »momentaneStaerke) verglichen. Sollte der empfangene Wert größer als dein eigener sein, hast du bzw. dein Calliope mini leider verloren, was im »Gewonnen«-Platzhalter der 2 entspricht. Hast du und ein Mitspieler genau den gleichen Wert erreicht, haben beide gewonnen: In beiden Fällen ist der vom Mitgewinner empfangene Wert nicht größer als der eigene, damit bleiben beide Spieler im Zustand »Gewonnen«.

Auch wenn dein Programm noch nicht ganz fertig ist, probier einmal aus, ob der Spielstart schon funktioniert. Dazu brauchst du nur zwei Calliope minis.

Wenn auf einem Calliope mini Knopf »A« gedrückt wird, sollten beide das Spiel starten, was du am Aufleuchten der RGB-LED in Gelb erkennen kannst.

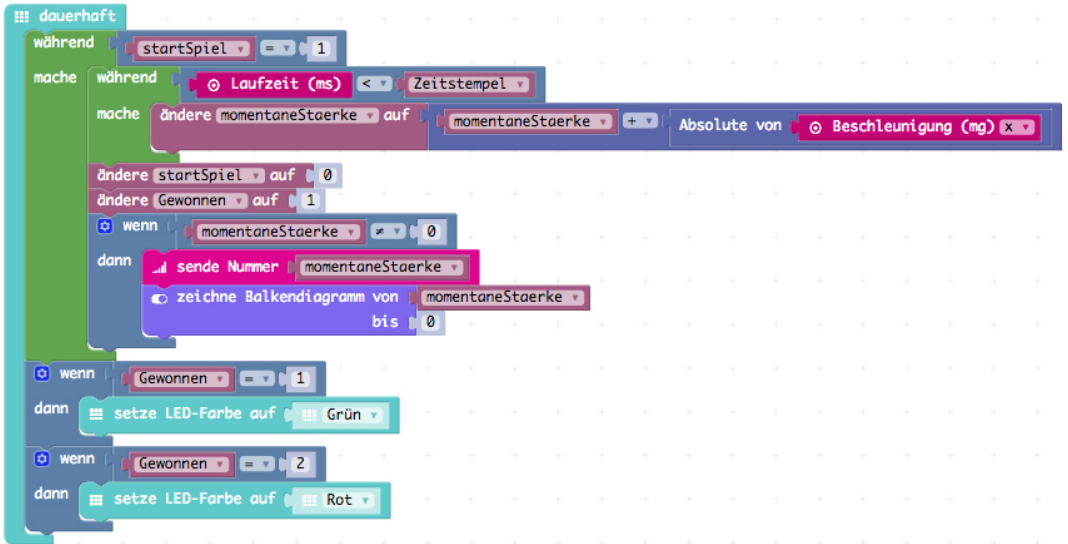


Abb. 9–24 Was Calliope ständig machen muss

Es fehlt nun noch das Aufsummieren der Beschleunigungswerte und die Anzeige, ob du gewonnen hast oder nicht. Dazu baust du in der »dauerhaft«-Schleife des Calliope mini zunächst eine Fallunterscheidung ein, ob das Spiel gerade läuft oder nicht (»startSpiel« muss dafür also 1 sein). Ist dies der Fall, wird in einer zweiten Schleife während des laufenden Spiels (also solange die aktuelle Zeit noch kleiner als der beim Start des Spiels festgelegte Wert im Platzhalter »Zeitstempel« ist) ständig die Beschleunigung gemessen und zu »momentaneStaerke« hinzuaddiert. Dabei ist es wichtig, den Absolutwert der Beschleunigung zu nehmen, damit sich die Beschleunigungen beim Hin- und Herbewegen in entgegengesetzte Richtungen nicht aufheben (also zu 0 addieren).

Ist der Zeitpunkt des Spielendes erreicht, wird »startSpiel« wieder auf nicht laufend (0) zurückgesetzt, die erreichte »momentaneStaerke« kurz angezeigt und an die anderen Calliope minis gesendet. Um auf Nummer Sicher zu gehen, wird das hier nur gemacht, wenn die Stärke tatsächlich nicht 0 ist, um nicht aus Versehen ein neues Spiel zu starten. Der Platzhalter »Gewonnen« wird außerdem auf 1 gesetzt. Solange keine weiteren Daten empfangen werden, die dem widersprechen, hast du das Spiel also gewonnen.

Nach dieser Fallunterscheidung wird noch der Platzhalter »Gewonnen« ausgewertet:

- ▶ Ist er 3 (unentschieden), wird gar nichts gemacht (um genau zu sein: Da das nur während des Spiels passiert, wo das Programm in der »während«-Schleife ist, wird dieser Fall eh nicht vorkommen.).
- ▶ Bei einer 1 hast du das Spiel gewonnen und die LED leuchtet grün.
- ▶ Bei einer 2 wurde das Spiel verloren und die LED leuchtet rot.

Und das war auch schon alles. Der Trick bei der Auswertung, wer gewonnen hat, besteht einfach darin, dass man zunächst einmal davon ausgehen kann, selber gewonnen zu haben, solange nicht ein Wert eintrifft, der größer als der eigene ist. In diesem Fall wird nachträglich auf die rote LED umgeschaltet, was aber so schnell geht, dass du es meist nicht sehen kannst. Damit ist die Anzahl der Mitspieler beliebig und könnte sich auch während eines Spiels noch ändern, wenn ein Spieler zum Beispiel ausfällt.

### 9.4.3 Verbesserte Punkteanzeige

Wie du vielleicht merkst, ist die Zahl für die eigene Stärke nicht wirklich gut zu lesen, da sie viel zu lang ist. Eine Alternative ist, die LEDs auf dem LED-Display des Calliope mini einzeln zu verwenden, sodass jede LED einer bestimmten Anzahl »Punkte« entspricht.

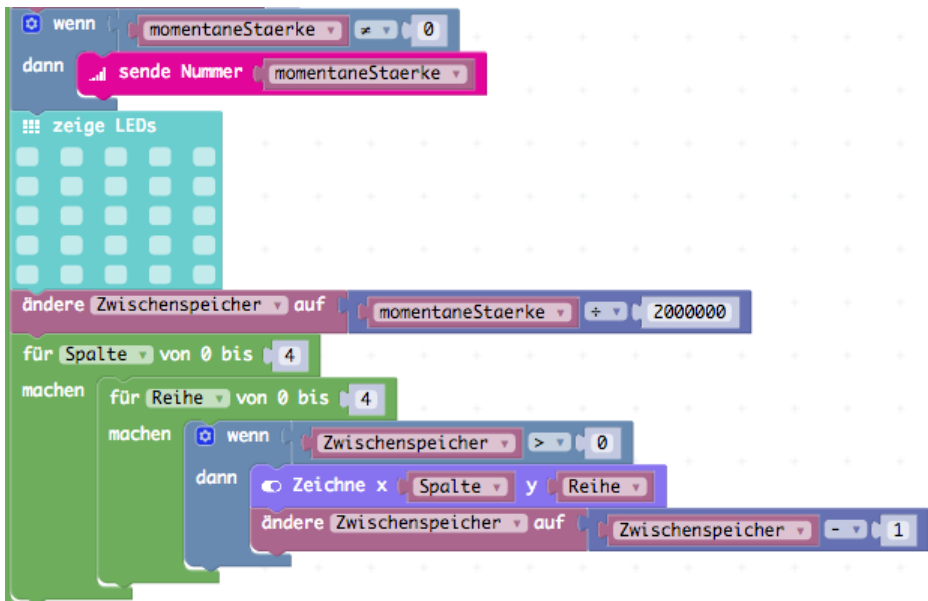


Abb. 9–25 Matrixdisplay als Punkteanzeige

Dafür kannst du den »zeige Nummer«-Block nach dem Senden des Platzhalters »momentaneStaerke« durch eine Anzeige per LED-Display ersetzen. Zunächst speicherst du den Wert der Stärke in einem neuen Platzhalter zwischen (deswegen auch »Zwischenspeicher« genannt). Da die Stärke sehr große Werte annimmt, kannst du den Wert für die »momentaneStaerke« durch einen bestimmten Wert teilen, was hier beim Ändern des Zwischenspeichers auf »momentaneStaerke/2000000« geschieht: Leuchten nach dem Schütteln gar keine LEDs auf, sollte er verkleinert werden. Sind danach alle an, sollte der Wert 2000000 noch einmal vergrößert werden – oder du bist einfach nur besonders gut.

Danach folgt jeweils eine Schleife für die Reihen und Spalten des Displays. Mit der ersten, äußeren Schleife werden alle Spalten des LED-Displays durchlaufen. Dabei wird für jede Spalte danach in einer weiteren Schleife auch jede Reihe durchlaufen. Solange der Zwischenspeicher noch größer als 0 ist, wird eine LED angeschaltet (mit der »Zeichne«-Funktion an der passenden Stelle für Spalte und Reihe), danach der Wert im Zwischenspeicher um 1 verringert. Irgendwann ist er soweit reduziert, dass keine weiteren LEDs mehr aufleuchten – oder es leuchten bereits alle LEDs.

## 9.4.4 Was man noch so machen könnte

Hier verwendest du ja nur die x-Komponente der Beschleunigung. Eine mögliche Änderung wäre es, alle drei Komponenten (x, y und z) auszuwerten – entweder als Addition zur momentanen Stärke oder als Subtraktion (wenn du die Spieler bestrafen willst, wenn sie den Calliope mini in einer unerlaubten Richtung bewegen). Du könntest natürlich auch die Rotation benutzen, sodass der Calliope mini möglichst schnell gedreht werden muss, um das Spiel zu gewinnen.

## 9.5 Das Roboterballett

Beim Roboterballett schlüpfen du und deine Freunde in die Rolle von Robotern, und ihr erhaltet auf euren Calliope minis Bewegungsanweisungen von einem Choreografen, nach denen ihr dann im Gleichklang tanzen könnt.

Für das Roboterballett brauchst du also eine Gruppe Kinder, die die Rolle der Roboter übernehmen, sowie ein weiteres Kind oder einen Erwachsenen für die Rolle des Choreografen, der die Anweisungen über seinen Calliope mini an die Roboter sendet. Jedes der Roboterkinder hält dabei seinen Calliope mini in der Hand und führt exakt die Bewegungen aus, die es auf dem Calliope angezeigt bekommt. Dabei ist der Mikrocontroller des Choreografen der Sender für alle anderen Empfänger-Calliope der Roboterkinder

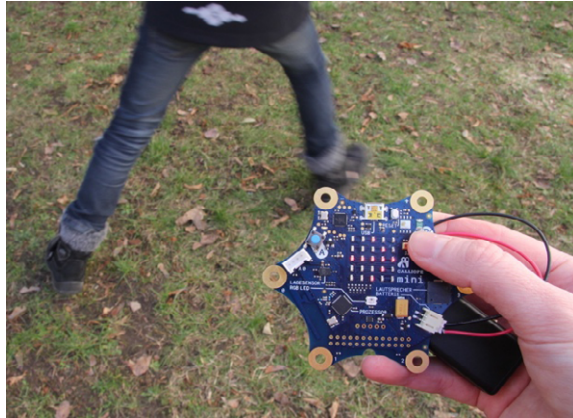


Abb. 9–26 Der Roboter soll einen Schritt nach rechts gehen.

Für die Programmierung des Roboterballetts solltest du dich bereits mit Logikverknüpfungen und dem Umgang mit Platzhaltern auskennen. Es ist also von Vorteil, wenn du bereits Programmiererfahrung besitzt und ein paar Projekte aus diesem Buch schon ausprobiert hast.

### 9.5.1 Bewegungsanweisungen senden

Zu Beginn deines Projektes musst du dir überlegen, aus welchen Grundbewegungen dein Ballett besteht. Ich entscheide mich für die Bewegungen vorwärts, rückwärts, rechts, links und hochspringen. Um diese Bewegungsanweisungen an das Roboterballett zu übermitteln, lege ich folgende Display-Anzeigen fest:

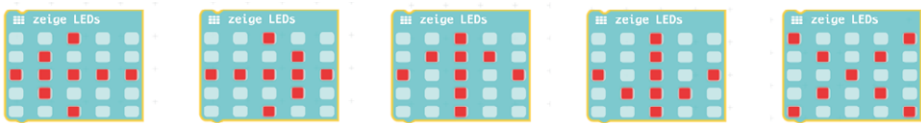


Abb. 9–27 Display-Anzeigen für das Roboterballett: links, rechts, vorwärts, rückwärts, hochspringen

Diese fünf Anzeigen sollen auf den Calliope-Displays der Roboter angezeigt werden, wenn der Choreograf mit seinem Calliope mini verschiedene Eingaben ausführt:

- ▶ links, wenn Knopf A gedrückt wird
- ▶ rechts, wenn Knopf B gedrückt wird

- ▶ vorwärts, wenn beide Knöpfe A+B gleichzeitig gedrückt werden
- ▶ rückwärts, wenn das Display nach unten zeigt, also umgedreht wurde
- ▶ hochspringen, wenn geschüttelt wird

Die Empfänger-Calliopes der RoboterKinder müssen per Funk eine Nachricht erhalten, welche Eingabe der Sender-Calliopes durchgeführt hat. Um den Wert dieser 5 Eingaben an den Empfänger-Board übermitteln zu können, benötigst du einen Platzhalter. Lege einen neuen Platzhalter an und nenne ihn »richtung«. Als nächsten Schritt ordnest du den 5 Bewegungen die Zahlenwerte aus der folgenden Tabelle zu. Das nächste Bild zeigt dir, wie du die Befehle im Editor-Skript zusammenstellen musst.

Bedeutung	Eingabe	Wert von »richtung«
Links	Wenn A gedrückt	0
Rechts	Wenn B gedrückt	1
Vorwärts	Wenn A+B gedrückt	2
Rückwärts	Wenn Display nach unten	3
Hochspringen	Wenn geschüttelt	4

Tab. 9–3 Zuordnung von Eingaben zum Zahlenwert des Platzhalters »richtung«

Wann immer eine dieser Eingaben erfolgt, muss der Wert des Platzhalters »richtung« auf den entsprechenden Wert aus der Tabelle geändert werden. Du brauchst insgesamt fünf Eingabe-Klammern und 5 »ändere richtung auf«-Blöcke. Dein Programmcode für das Versenden sollte jetzt ungefähr so aussehen:

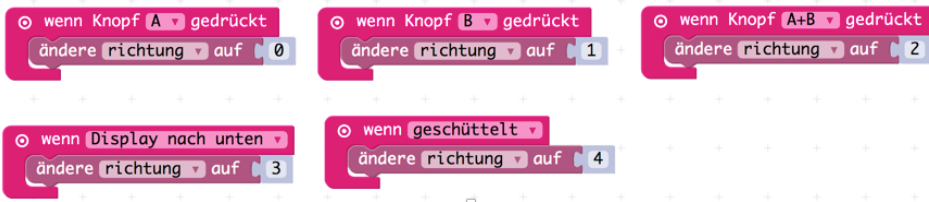


Abb. 9–28 Alle Programmierbefehle für den Choreografen

Der Sender-Calliope soll den Wert dieses Platzhalters an alle anderen Calliope-Empfänger senden, nachdem er den Zahlenwert geändert hat. Den notwendigen Befehl findest du in der Befehlsgruppe »Funk«. Wähle den »sende Nummer«-Block und füge anstatt der 0 einen »richtung«-Platzhalter ein.



Abb. 9–29 »richtung«-Platzhalter verändern und senden

Durch Auswertung dieser Nummer und der entsprechenden Anzeige auf den Empfänger-Boards werden die Roboterkinder wissen, wie sie sich bewegen sollen.

### Hinweis

Die Befehle für die Empfänger fügst du im gleichen Projekt ein wie die oberen Befehle für den sendenden Choreografen. Du siehst später, wie du bestimmen kannst, welcher Calliope mini der Choreograf ist und welche Boards die Roboterkinder sind.

## 9.5.2 Bewegungsanweisungen empfangen und auswerten

Die gesendete Nummer muss anschließend ausgewertet werden. Jeder Calliope mini kann darauf reagieren, wenn er ein Datenpaket empfangen hat. Der dazugehörige Befehl im Editor erwartet eine »receivedNumber«. Dies ist Englisch und bedeutet »empfangene Nummer«. Wähl diesen Block aus.

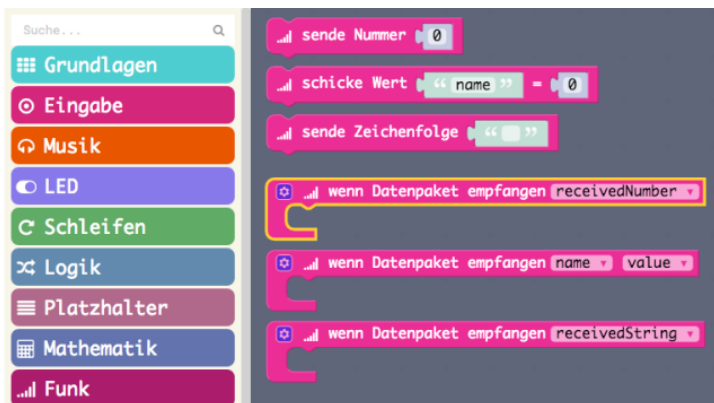


Abb. 9–30 Befehlsgruppe »Funk«

Zum Anzeigen der jeweiligen Display-Anzeige ist nun noch eine Logik-Abfrage notwendig, sodass die Empfänger-Boards wissen, welches der fünf Symbole angezeigt werden soll. Um zudem nachfolgend aufeinander immer wieder neue Befehle auf dem Display anzeigen zu können, ist es wichtig, dass das Display vor der Anzeige eines neuen Befehls immer wieder gelöscht wird. Dafür kannst du vor jeder Display-Anzeige noch den Befehl »Bildschirminhalt löschen« hinzufügen.

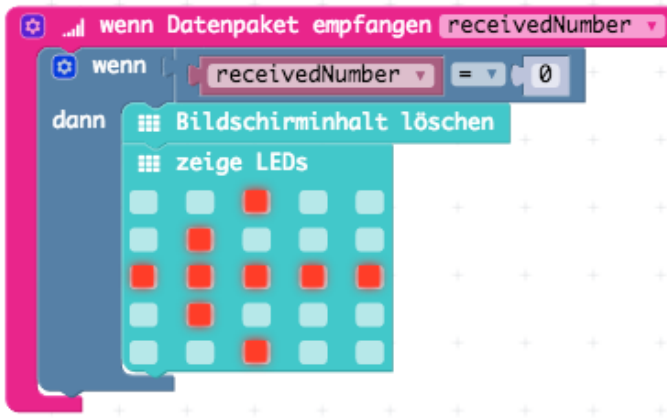


Abb. 9–31 Auswertung der empfangenen Nummer

Wiederhole diese Abfrage nun für die weiteren vier möglichen Fälle (rechts, vorwärts, zurück und hochspringen). Du kannst mit der rechten Maustaste auf das Wort »wenn« klicken und den Menüpunkt »Duplizieren« auswählen. Den neuen »wenn dann«-Block kannst du unten anfügen und die empfangene Nummer und die angezeigten LEDs verändern. Achte darauf, dass du keine Nummer doppelt hast.

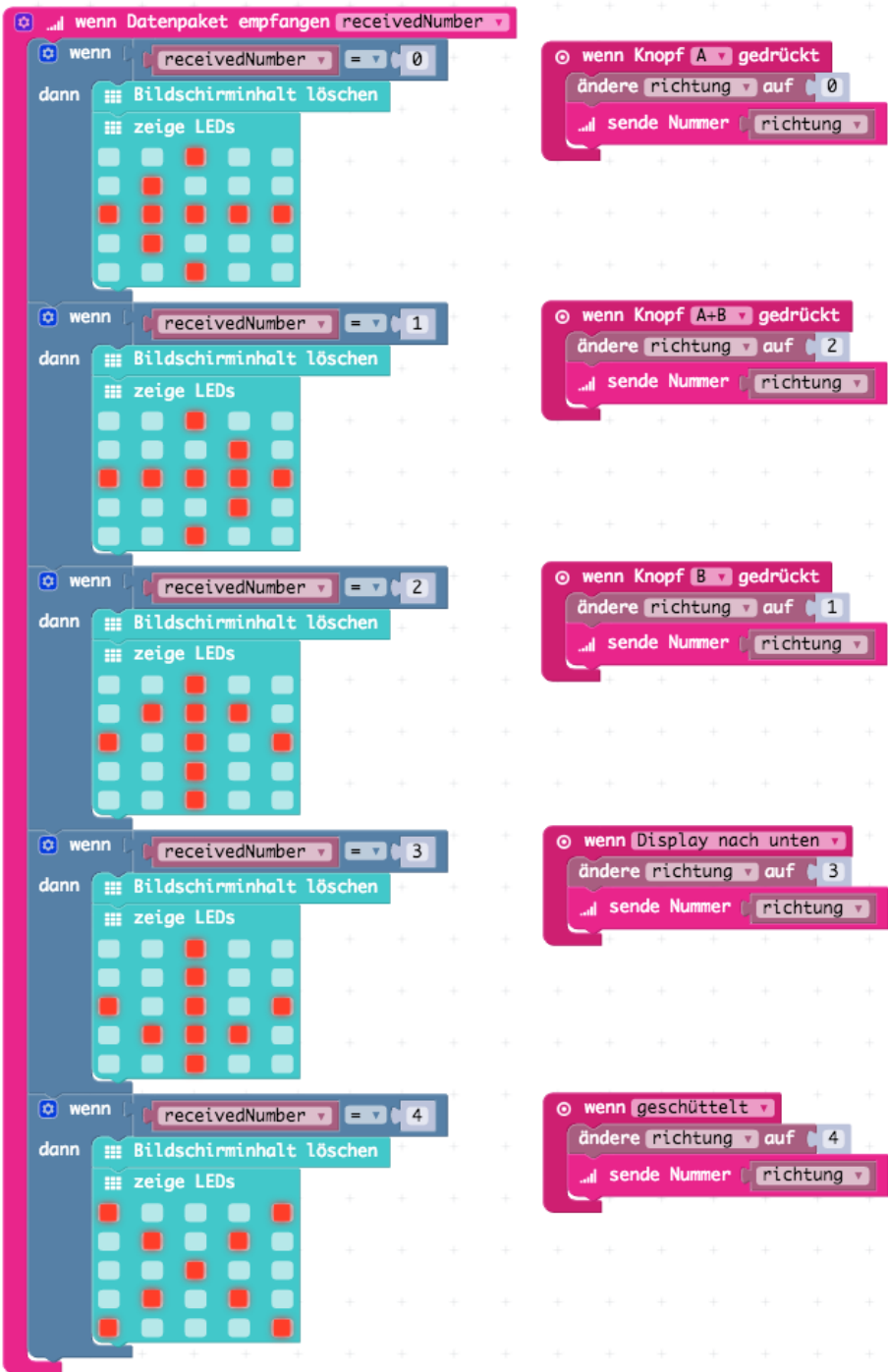


Abb. 9–32 Auswertung der empfangenen Nummer

Ihr könnt das Programm im Simulator ausprobieren oder gleich schon auf eure Calliope minis herunterladen. Jeder Calliope mini mit diesem Programm wird die Befehle ausführen können.

Wenn ihr das Roboterballett mit euren Calliope minis ausprobiert, werdet ihr bald feststellen, dass ihr alle die Choreografen sein könnt. Das bedeutet: Wenn ein Kind (zum Beispiel Maxi) den Knopf B drückt, erscheint der Pfeil nach rechts auf den beiden anderen Displays. Wenn nun aber eins der anderen Kinder danach vielleicht auf Knopf A drückt, dann erscheint auch auf Maxis Display ein Pfeil nach links.

Wenn ihr wollt, dass eine bestimmte Zeitlang nur Maxi der Choreograf sein soll, dann müsst ihr euer Programm noch ein bisschen verändern.

### 9.5.3 Einen einzigen Sender-Calliope bestimmen

Du benötigst für den folgenden Schritt noch einen Platzhalter. Erstell einen neuen Platzhalter und nenne ihn »empfangen«. Solange du diesen Platzhalter nicht veränderst, liegt sein Wert bei 0. Ein Platzhalter kann aber auch die Werte »wahr« oder »falsch« speichern. Solange du ihn nicht veränderst, ist ein Platzhalter »falsch«.

Stell dir vor, du als Roboterkind hältst deinen Calliope mini in der Hand und machst ihn an. Es wird nichts angezeigt, weil du noch kein Datenpaket empfangen hast. Also kannst du deinen Calliope mini auch die ganze Zeit fragen: »Hast du etwas empfangen?« – »*Nein*, habe ich nicht.« oder anders gesagt: »Calliope mini, du hast etwas empfangen!« – »*falsch*, habe ich nicht.« Nun führt ein anderes Kind (Maxi) eine Eingabe durch, zum Beispiel Knopf A drücken, und du fragst deinen Calliope mini wieder: »Hast du etwas empfangen?« – »Ja, stimmt, habe ich, es ist *wahr*.« Dein Calliope mini weiß nun bis zum nächsten Neustart, dass es bereits (früher) etwas empfangen hat.

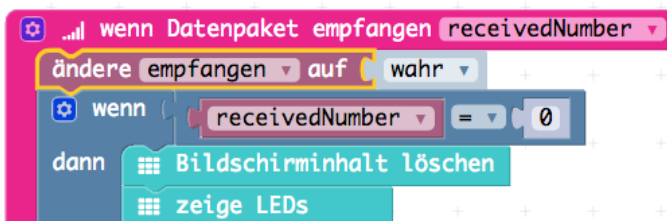


Abb. 9–33 Der Calliope mini soll sich merken, dass er ein Datenpaket empfangen hat.

Maxis Calliope mini hat aber noch kein Datenpaket empfangen, sondern nur eine Nummer gesendet. Der Platzhalter »empfangen« wurde auf Maxis Board noch nicht verändert, er ist immer noch bei 0 oder falsch.

Füge nun in den Eingabe-Blöcken eine Abfrage hinzu, die prüft, ob der Platzhalter »empfangen« noch falsch ist. Wenn dieser Platzhalter noch falsch ist, bedeu-

tet es, dass dieser Calliope mini (seit dem letzten Neustart) noch kein Datenpaket empfangen hat. Nur dann soll er den »richtung«-Platzhalter als Nummer an die anderen Calliope minis senden. Du findest die dafür notwendigen Befehle in der Befehlsgruppe »Logik« und benötigst außerdem den »empfangen«-Platzhalter.



Abb. 9–34 Nur senden, wenn noch nichts empfangen wurde.

Verändere alle Eingabeblocke in dieser Form. Jeder »sende Nummer richtung«-Block sollte von einem »wenn dann«-Block umschlossen werden, der überprüft, ob der Wert des Platzhalters »empfangen« falsch ist. Damit steht fest, ob es sich bei dem Calliope mini um den einzigen Sender oder einen der vielen Empfänger handelt.

Maxis Calliope mini hat noch kein Datenpaket empfangen und darf weiter-senden. Die anderen Calliope der RoboterKinder können nicht mehr senden, weil sie nach dem ersten Datenpaket ihren Platzhalter »empfangen« schon auf »wahr« verändert haben.

Wie bestimmt ihr nun, wer der nächste Choreograf sein soll, der den Robo-terKindern Befehle senden darf? Der Calliope mini, der als Erstes etwas sendet, bleibt bis zum nächsten Neustart der Sender. Wenn ihr nach einem Durchgang alle auf den weißen Reset-Knopf auf euren Calliope minis drückt, kommt ihr wie-der zurück zum Anfang. Noch niemand hat ein Datenpaket empfangen. Sobald ein Kind zum Beispiel auf den Knopf A drückt oder den Calliope mini schüttelt, werden bei allen anderen Calliope minis die entsprechenden Anzeigen aufleuch-ten. Dieses Kind darf die anderen dann steuern und Befehle senden, die anderen können nur noch Befehle empfangen.

Wenn du alle Schritte ausgeführt hast, sieht dein Programm so aus wie auf dem nächsten Bild, und ihr könnt mit dem Roboterballett beginnen, wenn alle Mitspieler dieses Programm auf ihre Calliope-Boards heruntergeladen haben.

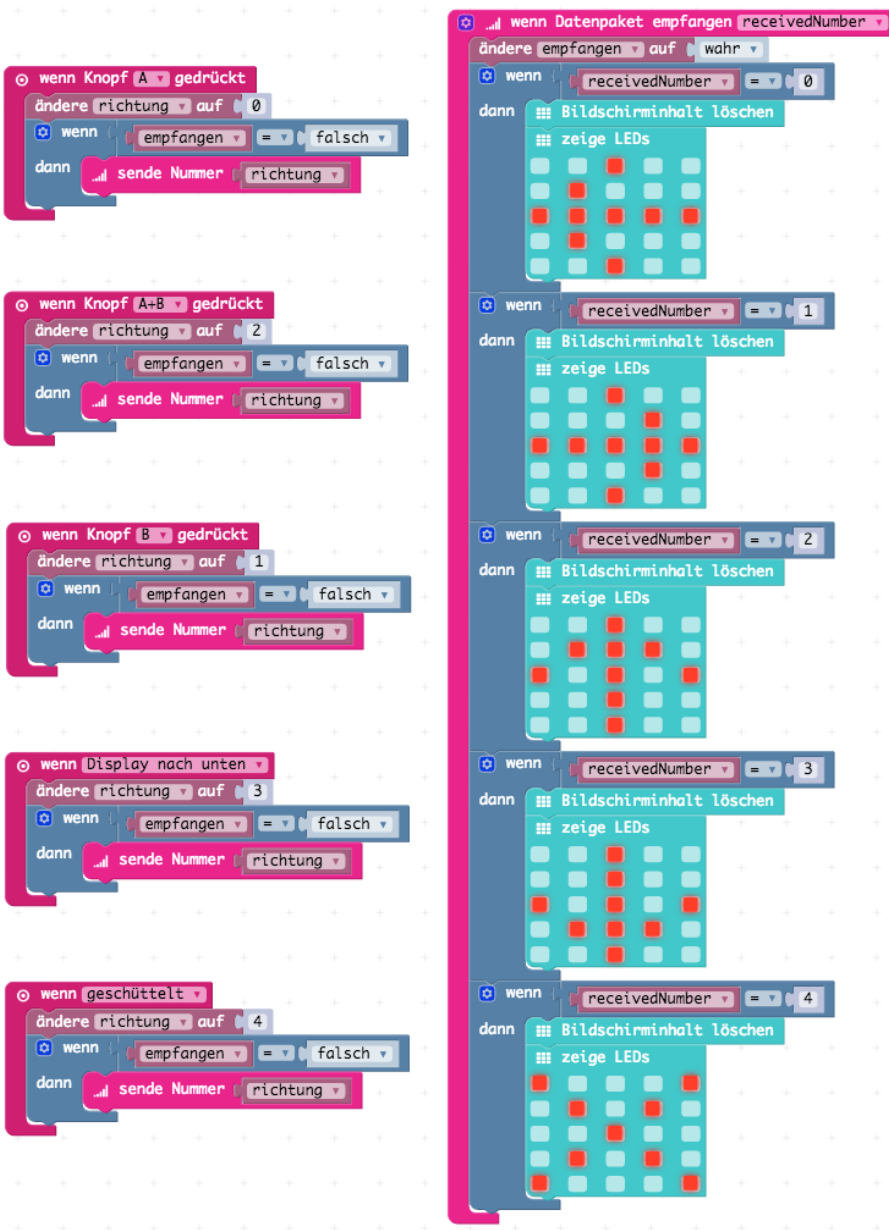


Abb. 9–35 Das fertige Programm für sendende und empfangende Calliope minis

## 9.5.4 Erweiterungen

Für die Ballettaufführung können sich die RoboterKinder noch effektvolle Roboterkostüme bauen. Mit Alufolie, Papiertüten und Papprollen lassen sich im Handumdrehen Helme und Armbänder zaubern – eurer Kreativität sind hier keine Grenzen gesetzt.

Ihr könnt euch auch noch andere Anweisungen ausdenken, solange alle tanzenden Mitspieler wissen, welche Bedeutung die Anzeigen auf dem Display haben. Eine leuchtende LED in der rechten oberen Ecke kann bedeuten, dass alle Roboter ihren rechten Arm heben müssen.

### Ballett aus mehreren Gruppen

Unter Verwendung des Programmierblocks »setze Gruppe« kannst du dein Ballett auch in zwei oder mehrere Gruppen unterteilen. Je nachdem, wie viele Gruppen du verwendest, brauchst du dann auch dieselbe Anzahl Choreografen, die über je eine Gruppe Regie führen.

Let's dance!

## 9.6 Eckstein, Eckstein ... alles muss versteckt sein!

Mit diesem Calliope-Projekt erfindest du das Versteckspiel neu! Wie ein Geheimagent benutzt du deinen Calliope mini, um das Versteck eines anderen Calliope-Mitspielers aufzuspüren. Dein Calliope-Board hilft dir durch seine Funkfähigkeit und die LED-Anzeige dabei, die versteckten Calliope minis aufzuspüren. Die LED-Anzeige zeigt dir genau an, wie nah du am Versteck bist. Kurz bevor du das Versteck gefunden hast, verrät der Calliope deines Mitspielers mit einem Ton aus dem Lautsprecher sein Versteck.



Abb. 9-36 Wer findet den versteckten Calliope mini?

Du fragst dich sicher, wie das funktionieren kann. Genau wie beim echten Versteckspiel, wo dein Mitspieler manchmal durch ein Geräusch auf sich aufmerksam macht, kann auch der Calliope mini auf sich aufmerksam machen. Allerdings nicht durch ein Geräusch, sondern durch ein Funksignal. Der suchende Calliope hört sich nach diesem Funksignal um und stellt fest, wie weit der versteckte Calliope entfernt ist. Je näher du mit deinem Calliope mini am Versteck des anderen bist, desto stärker wird das Funksignal.



Abb. 9-37 Das Funksignal wird stärker, je näher die Calliope minis einander sind

Wie du festgestellt hast, brauchst du zur Umsetzung für das Suchspiel zwei Calliope-Boards. Eins, das versteckt wird und als Sender arbeitet, und eins, das als »Sucher« zum Empfangen benutzt werden soll. Diesmal musst du also zwei unterschiedliche Programme bauen: für jeden der beiden Calliope minis ein anderes. Da du nun ungefähr weißt, wie das Versteckspiel funktioniert, kann es mit der Programmierung losgehen.

## 9.6.1 Das Programm für den versteckten Calliope

Du beginnst mit dem Programm für den versteckten Calliope mini. Als Erstes musst du über den Programmstart deinem Calliope einige Dinge mitteilen. Zum Beispiel müssen beide Calliope mini wissen, auf welcher Funkfrequenz sie sprechen. Das kannst du dir ähnlich wie einem Radiosender vorstellen. Diese Sender heißen beim Calliope »Gruppe«. Den Programmbefehl »setze Gruppe« findest du im Editor in der Rubrik »Funk« und dort wiederum unter »...Mehr«. Du wählst hier ganz einfach die Zahl 0 als Gruppe. Der Programmstart könnte wie auf dem folgenden Bild aussehen:



Abb. 9–38 Beim Starten wird eine Gruppe gesetzt

Damit du auch in kleineren Räumen suchen kannst und der Calliope mini nicht zu schnell gefunden wird, solltest du die Senderstärke etwas niedriger einstellen. Dein Calliope-Board funkt dann etwas »leiser« und das suchende Calliope-Board hat es später nicht zu einfach, den versteckten Calliope mini zu finden. Hierzu wählst du ebenfalls aus der Rubrik »Funk« und anschließend »...Mehr« den Programmierbefehl »setze Übertragungsstärke«. Da der größte wählbare Wert hier 7 ist, wählst du ungefähr die Mitte. Die Zahl 3 ist hier ein guter Anfang.



Abb. 9–39 Die Übertragungsstärke wird gesetzt

Die Startvorbereitungen hast du somit abgeschlossen, jetzt kommt der spannende Teil. Du musst dafür sorgen, dass dein Calliope mini auf sich aufmerksam macht. Dazu musst du es dauerhaft funken lassen. Dabei ist erst einmal egal, was dein

Calliope-Board per Funk überträgt, es soll später nur vom suchenden Calliope mini gehört werden. Dein Programm muss also um den Programmierbefehl »dauerhaft« und zum Beispiel »sende Zeichenfolge« erweitert werden.



Abb. 9-40 Fortwährend wird eine Zeichenfolge gesendet

Damit ist das Programm des versteckten Calliope-Spielers eigentlich schon fertig. Wenn du es auf deinen Calliope lädst, wirst du aber schnell feststellen, dass man nichts sehen kann. Das ist auch in Ordnung, aber unheimlich langweilig. Deshalb kannst du dein Programm so verändern, dass dein Calliope mini dir auf eine spannende Art und Weise mitteilt, dass er gerade auch wirklich funkt. Die beste Möglichkeit bietet dir dazu die LED-Anzeige. Du musst dazu nichts weiter tun, als dein Programm um drei Programmbefehle zu erweitern.

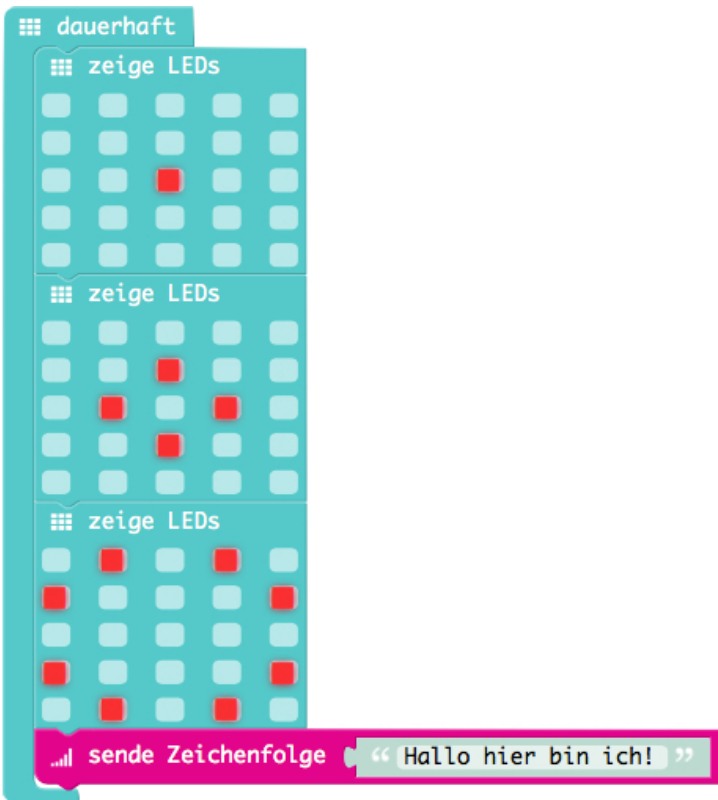


Abb. 9-41 Das Calliope-Display kommt hinzu

Jetzt spielt dein Calliope eine kleine Filmsequenz ab, die dir anzeigt, dass dein Calliope-Programm läuft und »Hallo hier bin ich!« sendet. Das Programm ist fürs Erste fertig. Es geht weiter mit dem zweiten Calliope mini, der als Sucher eingesetzt wird.

## 9.6.2 Das Programm für den suchenden Calliope

Auch hier beginnst du im Programmstart mit dem Programmbefehl »setze Gruppe«. Im Gegensatz zum versteckten Calliope wird der Programmbefehl »setze Übertragungsstärke« hier nicht benötigt. Dein erster Befehl ist also ganz einfach und sieht wie folgt aus.



Abb. 9–42 Der erste Befehl für den suchenden Calliope mini

Als Nächstes soll der suchende Calliope Ausschau nach dem versteckten Calliope halten. Hierzu fügst du dem Programmbefehl »wenn Datenpaket empfangen« den Platzhalter »receivedString« hinzu. Dieser Programmbefehl wird immer dann ausgeführt, wenn dein Calliope mini die Nachricht »Hallo hier bin ich!« vom versteckten Calliope empfängt.

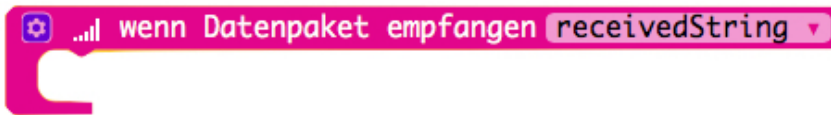


Abb. 9–43 Wenn der suchende Calliope eine Nachricht empfängt

Wie du in der Einleitung des Projekts gelernt hast, soll dein Calliope mini prüfen, wie stark das empfangene Funksignal des versteckten Calliope mini ist. Hierzu musst du am Programmbefehl »wenn Datenpaket empfangen receivedString« noch eine Spezialeinstellung vornehmen. Dazu klickst du auf das kleine blaue Kästchen mit dem Zahnrad.



Abb. 9–44 Das kleine Zahnrad links oben

Anschließend öffnet sich ein Kasten mit weiteren Programmierbefehlen. Dort findest du den Programmierbefehl »signal«. Diesen Befehl schiebst auf der rechten Seite zum Programmierbefehl »Packet«.

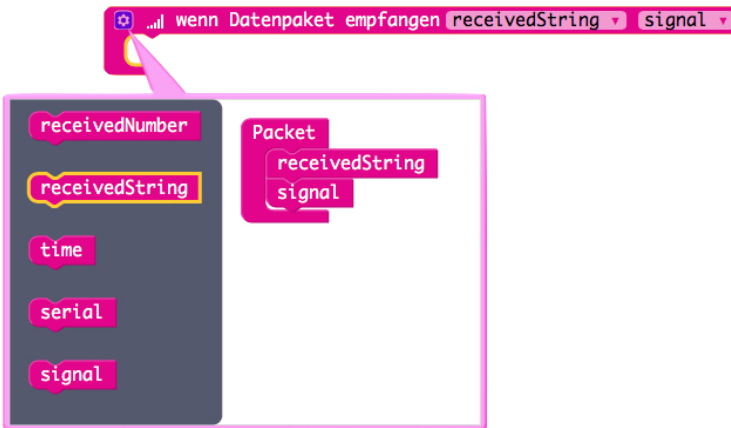


Abb. 9–45 Der Programmierbefehl »Packet« wird ergänzt

Dein suchender Calliope mini ist nun in der Lage, den versteckten Calliope zu hören. Außerdem verrät dir das Programm nun schon, wie weit der versteckte Calliope mini noch entfernt ist. Damit du die Entfernung auch ablesen kannst, fügst du deinem Programm den Programmierbefehl »zeige Nummer signal« hinzu.

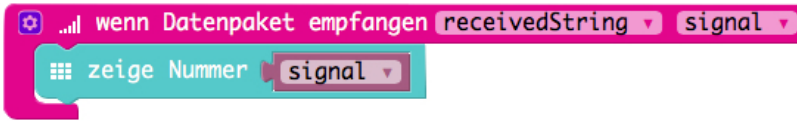


Abb. 9–46 Der Befehl »zeige Nummer signal« wird eingefügt

Du kannst das Programm nun testen, indem du es wie gewohnt auf den suchenden Calliope lädst. Schalte nun den versteckten Calliope mini ein. Dann beobachte, was der suchende Calliope macht. Wenn das Programm startet, siehst du, wie auf der LED-Anzeige eine Zahl erscheint. Bewege den suchenden Calliope nun in die Nähe des versteckten Calliope. Vor der angezeigten Zahl siehst du ein »-«, das können wir erst einmal ignorieren. Du wirst feststellen, dass die Zahl immer kleiner wird, je näher der suchende Calliope dem versteckten Calliope-Board kommt. Das »-« vor der Zahl kannst du durch einen weiteren Programmierbefehl entfernen. Hierzu erweiterst du das Programm durch den Programmierbefehl »Absolute von«. Du findest den Programmierbefehl in der Gruppe »Mathematik« und dort unter »...Mehr«.

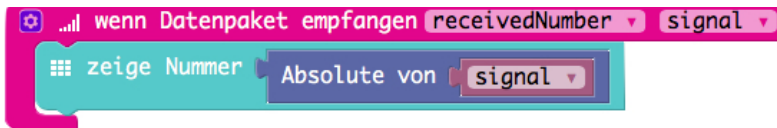


Abb. 9–47 »Absolute von signal« wird angezeigt

Jetzt weißt du, dass die Signalstärke sich verändert, wenn du dich dem versteckten Calliope näherst. Eigentlich bist du mit deinem Programm fertig, aber nach ein paar Versuchen bemerkst du, dass die Zahlen in recht großen Abständen über die LED-Anzeige laufen. Das dauert nicht nur lange, sondern ist auch nicht schön. Aus diesem Grund kannst du dein Programm so verändern, dass die Signalstärke durch unterschiedlich große Kreise angezeigt wird. Ein großer Kreis bedeutet, dass dein Calliope sehr weit vom versteckten Calliope entfernt ist. Ein Punkt hingegen bedeutet, dass dein Calliope das Versteck gefunden hat. In der Mitte gibt es noch einen mittelgroßen Kreis, der dir sagt, dass du schon ganz nah bist.

Zuerst betrachtest du den Fall, in dem die Signalstärke größer als 65 ist. Eine Zahl, die größer oder gleich 65 ist, bedeutet in deinem Programm, dass die beiden Calliope-Boards noch sehr weit voneinander entfernt sind. Da ein großer Kreis genau dies anzeigen soll, erweiterst du dein Programm also um einen »wenn dann«-Programmierbefehl. Anschließend zeichnest du über »zeige LEDs« den großen Kreis. Auf dem Bild siehst du, wie dein Programm jetzt aussehen soll.

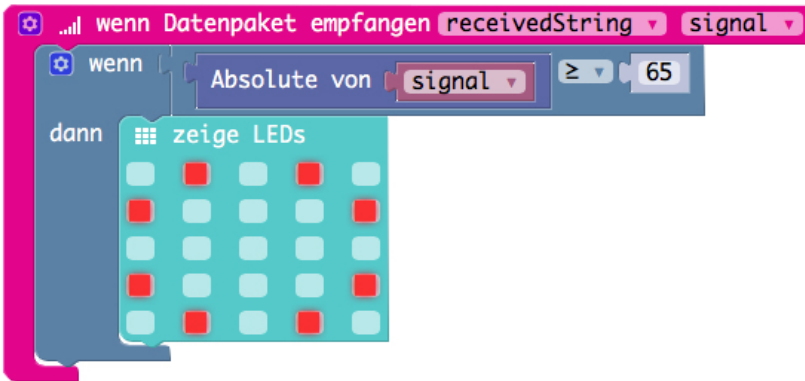


Abb. 9-48 Wenn Signal schwach, dann großer Kreis

Der nächste Fall sagt deinem Calliope mini, was er anzeigen soll, wenn es ganz nah am Versteck ist. »Ganz nah« bedeutet soviel wie »die Signalstärke ist kleiner als 55«. Auch hier verwendest du einen »wenn dann«-Programmierbefehl, der genau dies überprüft. Mit dem Programmierbefehl »zeige LEDs« kannst du dann den Punkt zeichnen.

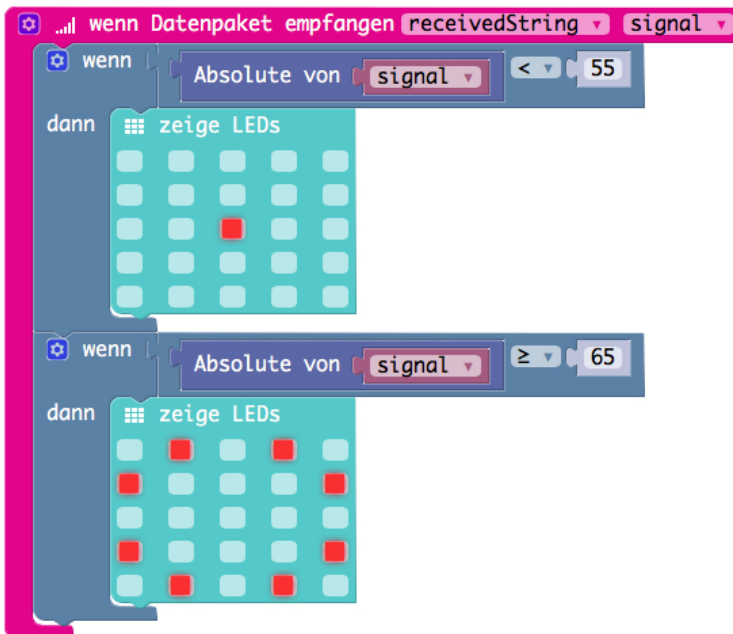


Abb. 9-49 Wenn ganz nah, dann nur Punkt anzeigen

Jetzt fehlt nur noch der Teil, der den mittelgroßen Kreis anzeigt. Da dein Calliope bereits weiß, wann es sehr nah ist und wann es weit weg ist, musst du dein Programm noch so ergänzen, dass dein Calliope auch weiß, wann es das Versteck

noch nicht gefunden hat, aber auch nicht so weit entfernt ist. Der gesuchte Programmierbefehl muss also genau die Signalstärken dazwischen überprüfen. Auch hierzu verwendest du natürlich am besten einen »wenn dann«-Programmierbefehl. Allerdings musst du ein wenig mehr überprüfen als in den beiden anderen Fällen. Denn zum einen muss die gefundene Signalstärke größer oder gleich 55 sein und zum anderen muss die Signalstärke kleiner oder gleich 65 sein. Du musst also die beiden vorherigen Fälle miteinander kombinieren. In der Gruppe »Logik« findest du einen Programmierbefehl »und«, der dir bei dieser Kombination hilft. Dein Programm ist nun fertig.

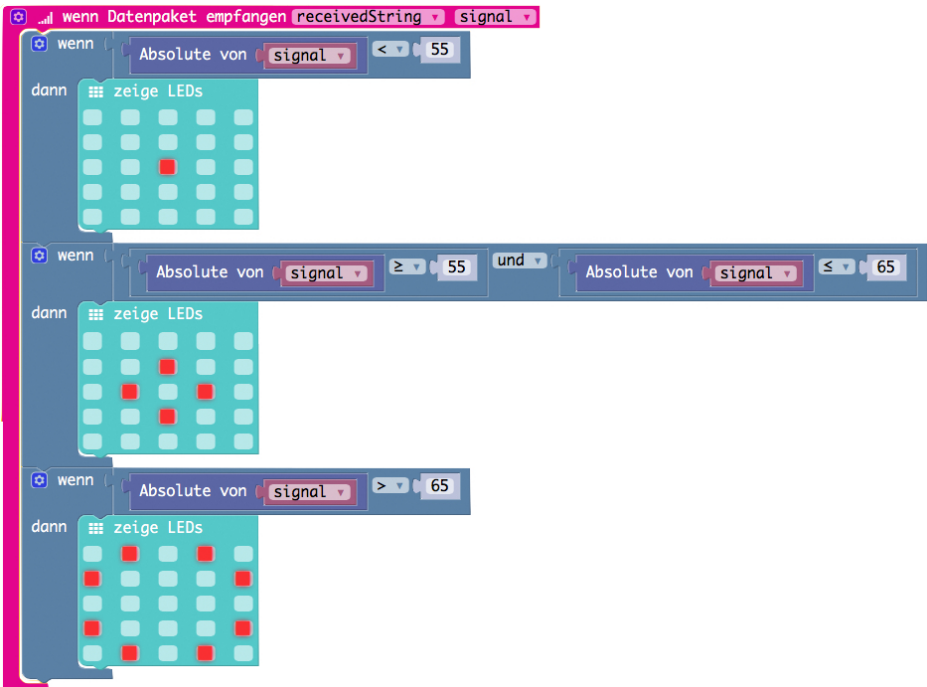


Abb. 9–50 Das Programm für den suchenden Calliope

Damit du den versteckten Calliope noch besser findest, kannst du das Programm des suchenden Calliope und des versteckten Calliope noch etwas verbessern, indem der versteckte Calliope mini einen Ton ausgibt, sobald der suchende Calliope sehr nah ist.

Im Programm des suchenden Calliope mini teilst du dem versteckten Calliope mini durch eine kurze Nachricht einfach mit, dass du sehr nah am Versteck bist. Dazu sendest du zum Beispiel eine Nummer.

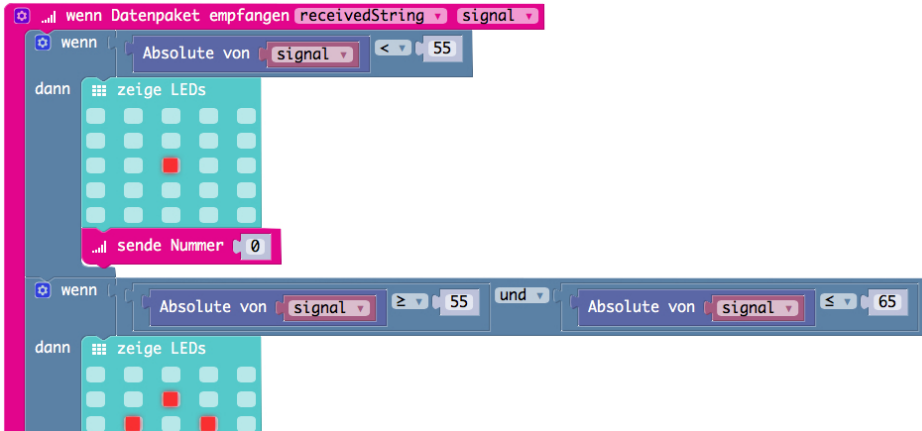


Abb. 9–51 Eine Nummer wird gesendet

Im Programm des versteckten Calliope wartest du so lange, bis der suchende Calliope die Nummer geschickt hat. Dies kannst du erreichen, indem du den Programmierbefehl »wenn Datenpaket empfangen« zum Programm des versteckten Calliope mini hinzufügst. Aber anstatt die Signalstärke zu überprüfen, musst du diesmal wirklich die empfangenen Daten auswerten. Da dein suchender Calliope mini die Zahl 0 versendet, sobald es sehr nah ist, muss im Programm des versteckten Calliope mini natürlich auch genau dies überprüft werden. Wenn also die empfangende Zahl 0 ist, dann soll dein verstecktes Calliope eine Note abspielen.



Abb. 9–52 Der versteckte Calliope mini spielt eine Note ab

### 9.6.3 Was ist noch möglich?

- ▶ Überleg doch einmal, was du an den Programmen verändern musst, damit unterschiedliche Töne bei unterschiedlichen Funksignalstärken ausgegeben werden.
- ▶ Du könntest das Spiel noch spannender machen, indem du das Programm des suchenden Calliope mini so veränderst, dass es den versteckten Calliope dazu bringt, einen Ton auszugeben, wenn du einen Knopf drückst.

Viel Erfolg beim Verstecken und beim Finden!

## 9.7 Theremin

Ein Theremin ist ein Musikinstrument, das den Abstand deiner Hände zu dem Instrument misst und abhängig davon die Frequenz und Lautstärke misst – also, welcher Ton gerade wie laut gespielt wird. Im Gegensatz zu anderen Musikinstrumenten ist damit ein gleichmäßiger Wandel des Tons möglich, da ja nicht eine Taste angeschlagen wird, sondern der Abstand deiner Hand zum Instrument beliebig geändert werden kann, was zu schön spukigen Klängen führt. In diesem Projekt wirst du etwas Ähnliches mit deinem Calliope mini bauen.

Leider ist der Lautsprecher des Calliope nicht besonders laut (wobei mancher, der das mit anhören muss, vielleicht anderer Meinung ist). Damit du das Theremin besser hören kannst, gibt es weiter unten noch zwei Erweiterungen: Du kannst deinen Calliope mini als Kopfhörer verwenden oder einen externen Lautsprecher anschließen.

Dafür bräuchtest du allerdings im ersten Fall einen weiteren Calliope mini, im zweiten Fall einen kleinen Lautsprecher (8 Ohm), etwas Draht und einen kleinen Widerstand (etwa 47 Ohm).

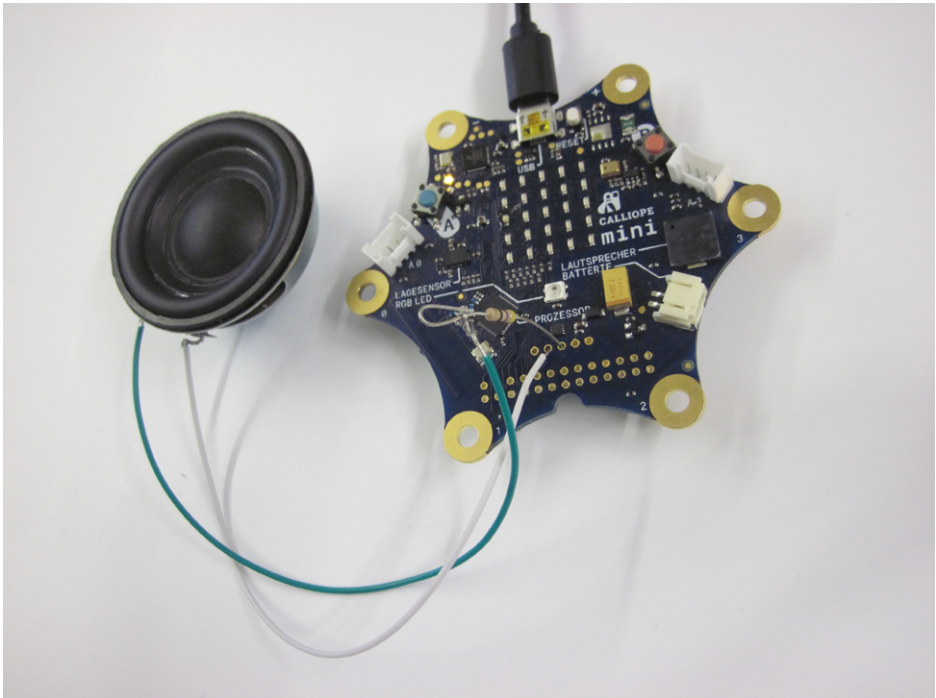


Abb. 9–53 Calliope mini mit Lautsprecher

## 9.7.1 Los geht's

Der Lichtsensor des Calliope mini ist ein Sensor, der sich prima als Grundlage für das Theremin eignet: Wenn du den Calliope mini entweder unter eine Lampe legst und mit der Hand abschattest oder einfach mit einer Taschenlampe unterschiedlich beleuchtest, wird ein ähnlicher Effekt wie bei einem richtigen Theremin erzielt.

Das Grundprogramm noch einfacher als die Alarmanlage mit dem Lichtsensor, die am Anfang des Buches gebaut wurde.



Abb. 9-54 Das Basisprogramm für das Theremin

Du benutzt den Lichtsensor, um damit einfach die Frequenz des Klingeltons festzulegen. Wenn du es ausprobierst, merkst du aber, dass es nicht richtig gut klingt. Leider stimmen die Zahlenwerte, die der Sensor liefert, nicht richtig zu dem, was man als unterschiedliche Töne haben will – da muss also noch etwas nachgebessert werden.

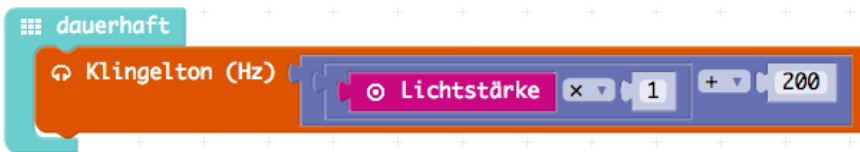


Abb. 9-55 Das Theremin mit etwas Mathematik

Dazu kannst du ein wenig Mathematik gebrauchen: Wenn der Sensor zu niedrige Werte liefert, kann man nur ein dumpfes Brummen hören (wenn überhaupt etwas zu hören ist). Um etwas schönere Töne zu erzielen, kannst du einfach einen Wert (hier 200) zum Sensorwert addieren. Hier merkst du, dass eine höhere Frequenz gleichbedeutend ist mit höheren Tönen. Zusätzlich könnte man auch die Sensorwerte mittels Multiplikation noch weiter vergrößern. In dem Beispiel multiplizierst du nur mit 1, was am Ergebnis nichts ändert und damit erst einmal nichts wirklich bringt. Du kannst aber einfach einmal an beiden Zahlenwerte so lange herumspielen, bis dein Lieblingstonbereich erreicht ist.

## 9.7.2 Ein Calliope mini als Kopfhörer

Am besten ist der Lautsprecher des Calliope mini natürlich zu hören, wenn er möglichst nah am Ohr ist. Da der Lichtsensor, mit dem du die Melodie erzeugst, aber direkt daneben ist, kann man ihn so nicht benutzen. Stattdessen kannst du, falls du ein zweites (oder gar drittes) Calliope-Board zur Verfügung hast, diesen Calliope mini als Kopfhörer verwenden.



Abb. 9–56 Musik-Sender

Ein Calliope mini arbeitet dabei als Sensor, der zunächst den Wert des Lichtsensors misst und entsprechend deinen mathematischen Berechnungen für die besten Töne an alle Calliope-Board einer bestimmten Gruppe (hier 42) sendet. Die Gruppe wird direkt beim Start des Calliope mini festgelegt.

Danach wird dauerhaft wie zuvor die Lichtstärke ausgelesen und mit der gleichen Mathematik wie zuvor angepasst. Der Wert wird allerdings nicht mehr genutzt, um den Klingelton direkt abzuspielen, sondern in einem Platzhalter namens »Frequenz« gespeichert. Danach wird der Klingelton mit dieser Frequenz auch noch auf dem Calliope mini selber abgespielt. Das schadet nicht, und du kannst einfach kontrollieren, ob das Programm läuft. Danach wird die Frequenz auch noch an alle anderen Calliope minis in derselben Gruppe gesendet.

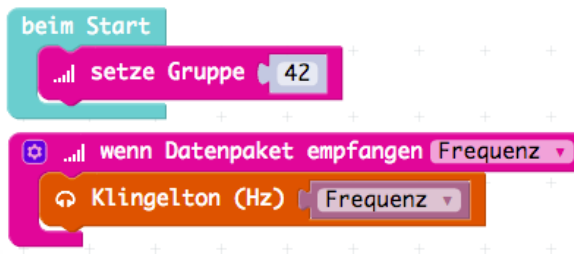


Abb. 9–57 Musik-Empfänger

Einen zweiten Calliope mini programmierst du als Empfänger mit einem weiteren Programm. Es kommt beim Starten in dieselbe Gruppe wie das sendende Theremin (hier die 42). Jedes Mal, wenn ein neuer Wert empfangen wird, wird dann der Klingelton auf den empfangenen Wert geändert. Damit kann man den einen Calliope an sein Ohr halten, während man den anderen zum Erzeugen der Melodie verwendet. Funktioniert das bei dir?

### 9.7.3 Es wird lauter – Anschluss eines externen Lautsprechers

Eine weitere Möglichkeit, den Calliope mini besser klingen zu lassen, ist der Anschluss eines externen Lautsprechers (wie auf dem Bild am Anfang des Kapitels). Dafür brauchst du natürlich erst einmal einen externen Lautsprecher, vielleicht hast du so etwas in der Bastelbox oder kannst jemand fragen, der so etwas hat. Es muss nichts Besonderes sein, irgendein Kleinlautspecher sollte funktionieren. Im Grunde kannst du auch genauso einen alten Kopfhörer anschließen, je nachdem, was du halt da hast. Dann brauchst du noch etwas Draht (einfache Büroklammern zum Beispiel bestehen nur aus Draht) und für den Lautsprecher wahrscheinlich noch einen kleinen Widerstand (auch hier kannst du wieder kreativ Sachen nutzen, die du gerade griffbereit hast: Mit einem Bleistift beispielsweise kann man prima Widerstände bauen.) Programmieren musst du in diesem Fall nichts, sondern nur den Lautsprecher anschließen.

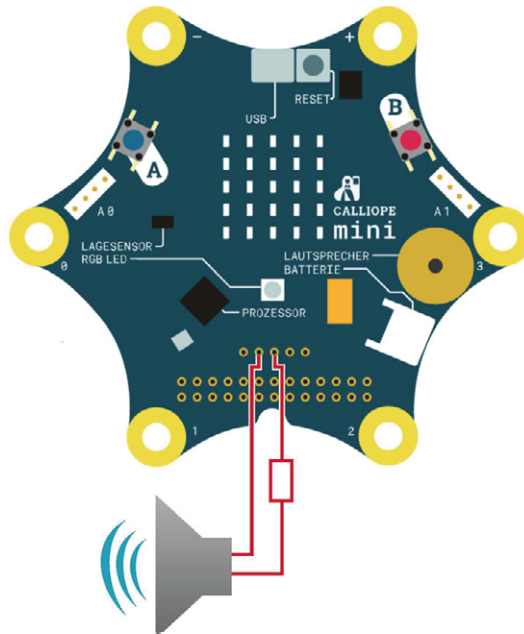


Abb. 9–58 Calliope Lautsprecherkontakte

Die Kontakte dazu verbergen sich in der Kontaktreihe für den Motoranschluss, wie auf dem Bild gezeigt. Die Anschlüsse auf dem Calliope mini sind leider schlecht zugänglich für einfache Basteleien und auch noch unbeschriftet. Als ersten Schritt kannst du – falls dein Lautsprecher nicht schon welche hat – Drähte an ihm befestigen. Hier wurde eine sogenannte Litze verwendet, ein Bündel aus dünnen Drähten mit einer Kunststoffisolierung außen herum. Die Isolierung wurde an den Enden entfernt (das geht normalerweise schon, indem du den Draht zwischen zwei Fingernägel klemmst und dann zum Ende abziehst (genauer wurde das bereits in Abschnitt 7.2.4 erklärt)). Das kannst du dir sparen, wenn du einen Draht ohne Isolierung verwendest (ein super angenehmer Draht dafür ist sogenannter Silberdraht aus dem Bastelladen oder aber die angesprochenen Büroklammern aus Draht).

Der Draht wird einfach um die Kontakte des Lautsprechers herumgewickelt und in die passenden Löcher im Calliope mini gesteckt. Auf der Rückseite kannst du die Drähte umbiegen und mit einem Streifen Klebeband auf den Calliope mini kleben. Du musst nur aufpassen, dass die blanken Drähte keine anderen metallischen Stellen berühren außer den beiden Löchern, daher solltest du sie auch am besten mit Klebeband festkleben.

Wenn beim Einstecken der Drähte beim Calliope mini die Lichter ausgehen (vor allem die kleine gelbe Power-LED), verbraucht der Lautsprecher zu viel Strom. Das kannst du reparieren, indem du einen Widerstand zwischen einen Draht des Lautsprechers und den Ausgang des Calliope mini einbaust: Je größer der Widerstand, desto geringer der Stromverbrauch, aber damit auch die Lautstärke. Du müsstest also durch Ausprobieren herausfinden, mit welchem Widerstand sich der Calliope mini nicht mehr abstellt, der Lautsprecher aber noch laut genug ist.

## Veränderbare Widerstände

Am leichtesten lässt sich der Widerstand einstellen, wenn du ein Potentiometer oder einen anderen veränderbaren Widerstand griffbereit hast. Diese Widerstände haben üblicherweise drei Kontakte. Zwischen zwei der Kontakte (meist die äußeren) befindet sich Material, durch das der Strom nicht so gut fließt. Der dritte Kontakt ist ein Schleifkontakt auf diesem Material. Damit befindet sich zwischen diesem Kontakt und einem der äußeren je nach Einstellung unterschiedlich viel Material, der Widerstand ändert sich dementsprechend.

## 9.7.4 Erweiterungen und Änderungen

Natürlich kannst du auch beide Varianten kombinieren, sodass mehrere Calliope minis mit Lautsprechern per Funk angesteuert werden. Auch könntest du mit zwei Calliope-Boards deinen eigenen Funkkopfhörer basteln. Schließlich ist der Lichtsensor nicht der einzige Sensor, der hierfür verwendet werden kann: Beschleunigung, Rotation und Magnetkraft wären weitere Möglichkeiten. Schaffst du es, das Programm so anzupassen, dass diese Sensoren anstelle des Lichtsensors genutzt werden?



# 10

## Jetzt kommt Bewegung ins Spiel ...

In diesem Kapitel erfährst du, wie du mit dem Calliope mini Motoren steuern kannst.

Die Einführung zu diesem Kapitel wurde von Mario Lukas geschrieben, ebenso wie das Projekt »Der Calliope-Tee-Roboter«. Das Projekt »Das lichtscheue Borstentier« ist von Lina Wassong.

## 10.1 Wie funktioniert ein Elektromotor?

Ganz gleich, ob elektrische Zahnbürste, Waschmaschine, CD-Player oder große Lokomotiven: In all diesen Geräten und Maschinen befindet sich ein Elektromotor. Du fragst dich nun sicher, wie ein solcher Elektromotor funktioniert. Alle Elektromotoren funktionieren nach dem gleichen Prinzip. Dabei wird Elektrizität in eine mechanische Bewegung umgewandelt. Vielleicht bist du jetzt ein wenig verwirrt und findest, dass alles sehr kompliziert klingt. Keine Angst, so kompliziert ist es wirklich nicht.

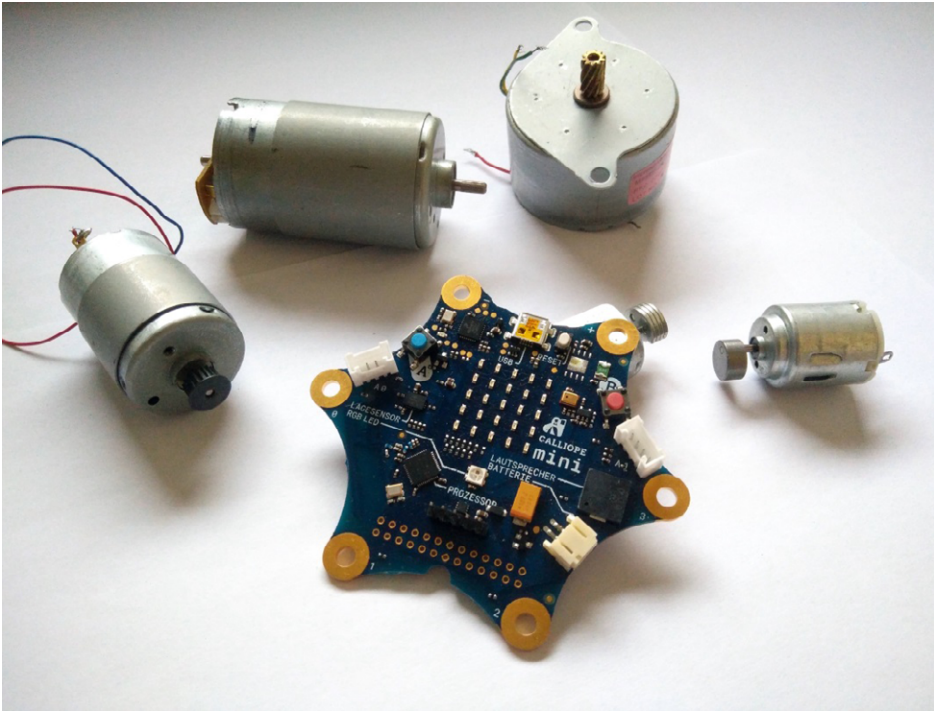


Abb. 10–1 Motoren werden vom Calliope mini gesteuert

Eigentlich brauchst du nur drei Zutaten, um einen funktionierenden Elektromotor zu bauen:

- ▶ eine Batterie
- ▶ einen Hufeisenmagneten
- ▶ eine Drahtspule

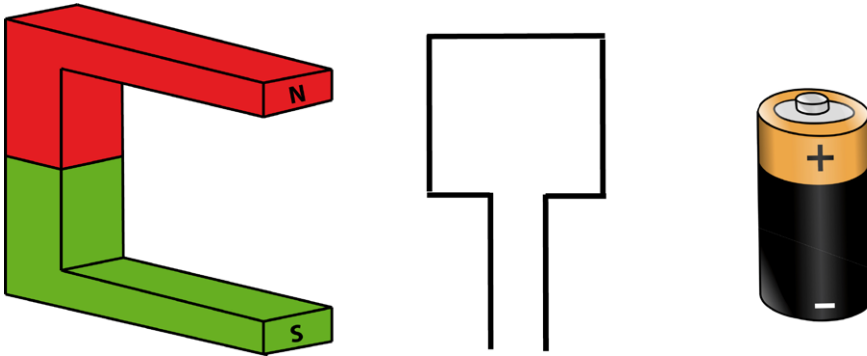


Abb. 10–2 Die Zutaten für einen Motor

Eine Drahtspule kann ein einfaches Stück Draht sein, das in eine spezielle Form gewickelt ist. Das Spannende passiert, wenn du eine Batterie an die Enden der Drahtspule anschließt. Die Spule wird dann nämlich auch zu einem Magneten. Genau genommen spricht man dann von einem Elektromagneten. Genau wie ein normaler Magnet besitzt auch der Elektromagnet einen Nord- und einen Südpol. Je nachdem, wie herum du Plus und Minus der Batterie anschließt, ändert sich die Position von Nord- und Südpol.

Positionierst du die Spule nun in der Mitte des Hufeisenmagneten und schließt die Batterie an, dreht sich die Spule. Was ist passiert? Eine wichtige Eigenschaft von Magneten ist, dass sich die unterschiedlichen Pole (Nord- und Südpol) gegenseitig anziehen. Gleiche Pole hingegen stoßen sich ab. Diese Regel gilt gleichermaßen für den Elektromagneten und den Hufeisenmagneten. Wie du ja weißt, besitzt die Spule mit angeschlossener Batterie einen Nord- und einen Südpol. Je nachdem, wie die Spule nun zum Hufeisenmagneten ausgerichtet ist, ziehen sich die Pole der beiden Magnete an oder stoßen sich ab – es entsteht also eine Bewegung. Damit ist der einfache Elektromotor fast fertig. Der Elektromotor kann bisher jedoch nur eine halbe Umdrehung schaffen. Er bleibt an der Position stehen, wo sich die Pole der beiden Magneten nur noch anziehen. Wie oben bereits erwähnt, kann man die Position der Pole des Elektromagneten ganz einfach verändern, indem man die Batterie andersherum anschließt. Wenn das geschehen ist, dreht die Spule sich eine halbe Umdrehung weiter. Damit man jedoch nicht ständig die Batterie umdrehen muss, gibt es in einem richtigen Elektromotor einen sogenannten Kommutator. Der Kommutator unterbricht automatisch die Stromverbindung und sorgt dafür, dass der Strom in einer anderen Richtung durch die Spule fließt.

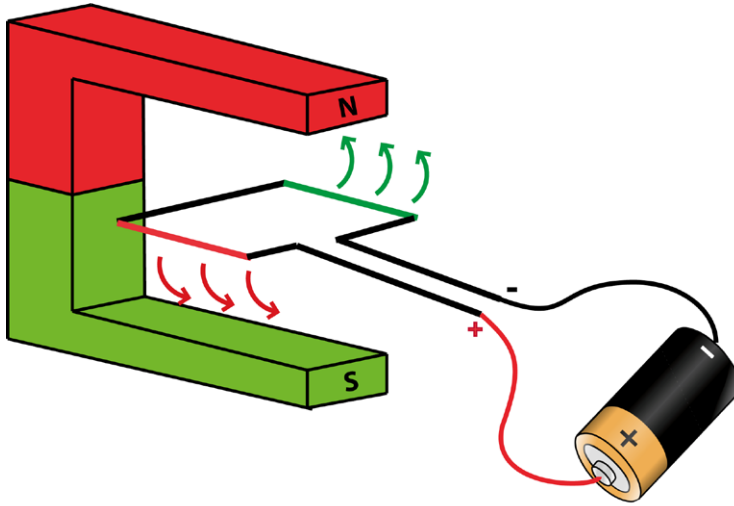


Abb. 10–3 Der Kommutator unterbricht die Stromverbindung

## 10.1.1 Wie funktioniert ein Servomotor?

Wie ein Elektromotor funktioniert, hast du gerade gelernt. Ein Servomotor ist eine spezielle Form des Elektromotors. Ein Servomotor besitzt zusätzlich noch Zahnräder, die man *Getriebe* nennt, und eine Elektronik. Die Elektronik sorgt dafür, dass sich der Servomotor nicht wie ein normaler Elektromotor andauernd dreht, sondern nur an vorher festgelegten Positionen. Leider kann ein Servomotor deshalb aber auch nur die Hälfte einer ganzen Umdrehung schaffen. Das Tolle ist jedoch, dass du die genaue Position auf dieser halben Umdrehung selbst bestimmen kannst.

Wie du auf der Abbildung sehen kannst, hat ein Servomotor nicht nur zwei Anschlüsse für Plus und Minus, sondern noch einen dritten. Genau über diesen Anschluss wird dem Servomotor die gewünschte Position mitgeteilt, also die Elektronik bedient.

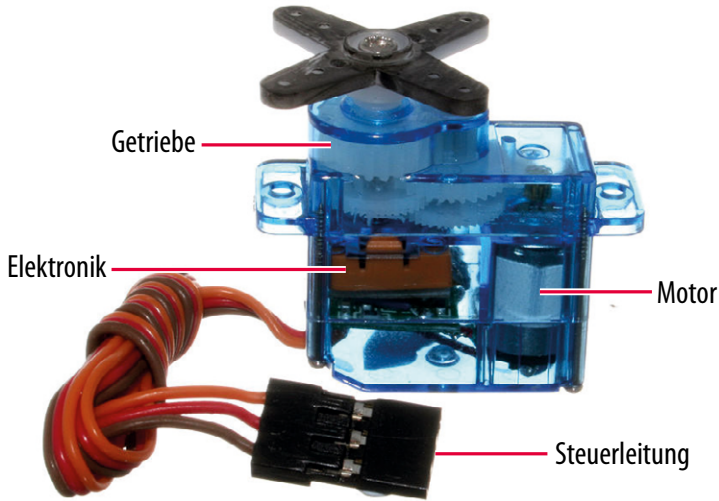


Abb. 10-4 Der Servomotor

## 10.1.2 Motoren mit dem Calliope mini ansteuern

Natürlich kannst du Motoren mit deinem Calliope ansteuern, genau genommen sogar bis zu zwei Motoren gleichzeitig. Jedoch musst du darauf achten, wie die Motoren am Calliope angeschlossen werden. Auf dem nächsten Bild kannst du sehen, wie ein einzelner Motor am Calliope mini angeschlossen wird.

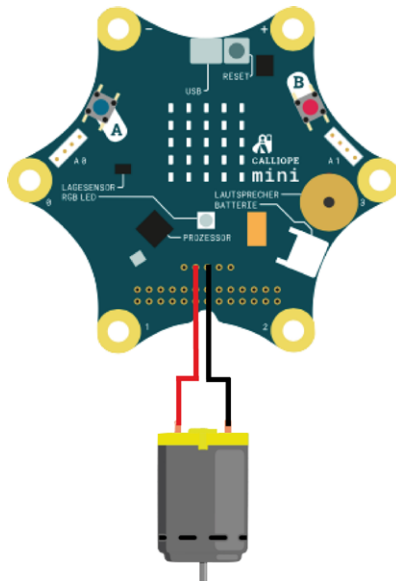


Abb. 10-5 Anschluss eines einzelnen Motors an den Calliope mini

Damit der Motor sich dreht, musst du deinem Calliope mini natürlich über ein Programm sagen, was er machen soll.

## Motoren am Calliope anschließen

Du kannst Motoren nicht an jedem beliebigen Pin deines Calliope anschließen. Zum Anschließen von Motoren sind an deinem Calliope die Löcher unten vorgesehen. Dort kannst du die Kabel für die Motoren direkt anlöten. Praktischer ist es jedoch, wenn du dort sogenannte Stiftleisten anlötet. Am besten fragst du einen Erwachsenen, ob er dir dabei hilft.

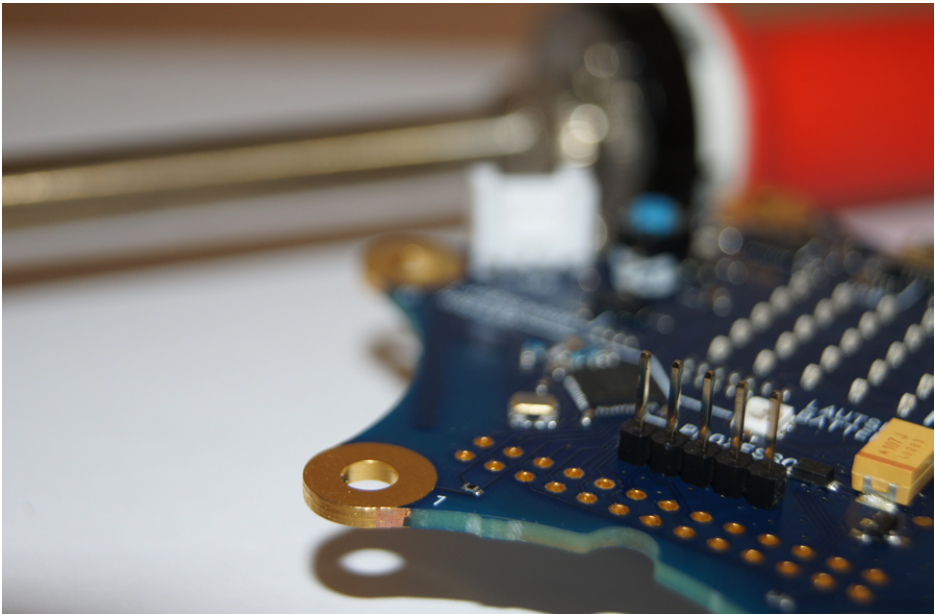


Abb. 10–6 Hier siehst du die Stiftleisten des Calliope mini in Nahaufnahme

Das einfachste Programm, um einen Motor anzusteuern, besteht aus einem »beim Start«-Programmierbefehl, gefolgt von einem »Motor an mit 50«-Befehl. Die Zahl 50 legt dabei fest, wie schnell sich der Motor drehen soll. Wenn du den Motor nach einer Weile wieder stoppen möchtest, benötigst du einen weiteren Programmierbefehl für Motoren, nämlich »Motor auslaufen«. Über das Auswahlfeld beim Programmierbefehl »Motor auslaufen« kannst du zusätzlich alternativ zwischen »Motor Pause« und »Motor schlafen« wählen.

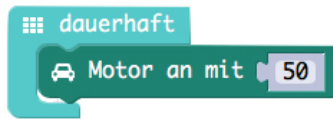


Abb. 10-7 Motor an

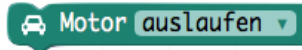


Abb. 10-8 Motor aus

## Motorgeschwindigkeit und Richtung

Du kannst eine Motorgeschwindigkeit zwischen 0 und 100 wählen. Wenn du ein Minuszeichen vor die Zahl schreibst, läuft dein Motor in die andere Richtung. Der Richtungswechsel funktioniert aber nur, wenn ein Motor angeschlossen ist.

Falls du planst, einen Roboter auf Rädern zu bauen, benötigst du zum Lenken zwei Motoren. Um zwei Motoren am Calliope mini zu betreiben, musst du die Motoren wie auf dem folgenden Bild an dein Calliope-Board anschließen.

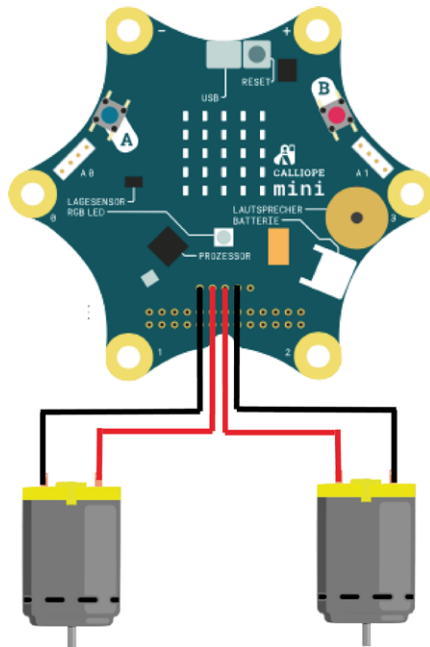


Abb. 10-9 So werden zwei Motoren an deinen Calliope mini angeschlossen

Dann kannst du mit dem Programmierbefehl »Motor A mit 50« den Motor A zum Drehen bringen.



Abb. 10–10 Motor A erhält einen Befehl

Über das Auswahlfeld kannst du entscheiden, ob nur Motor A, nur Motor B oder Motor A und B gleichzeitig drehen sollen.

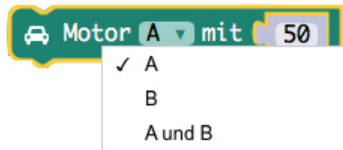


Abb. 10–11 Einer oder beide Motoren

- ▶ Versuche jetzt, ein kleines Programm zu bauen, das den Motor für einen Moment laufen lässt, stoppt und anschließend wieder laufen lässt.
- ▶ Jetzt kannst du dir für den Sommer, wenn es heiß ist, einen Ventilator mit verschiedenen Geschwindigkeitsstufen bauen.
- ▶ Was musst du am Programm verändern, damit sich die Geschwindigkeit des Motors verändert?
- ▶ Beobachte doch einmal, was passiert, wenn du den Programmierbefehl »Motor an mit 50« auf »Motor an mit -50« veränderst.

### 10.1.3 Servomotoren mit dem Calliope ansteuern

Ein Servomotor lässt sich viel leichter an deinem Calliope anschließen als Motoren. Denn du kannst wie gewohnt die großen Kontakte an deinem Calliope benutzen. Hierzu schließt du das schwarze Kabel des Servomotors an den »-« Kontakt an. Das rote Kabel musst du an »+« anschließen. Nun musst du den Servomotor nur noch an einen Pin des Calliope anschließen. Du kannst dazu zwischen Pin 1 und 2 wählen. In der Abbildung wurde das gelbe Signalkabel des Servomotors an Pin 1 angeschlossen.

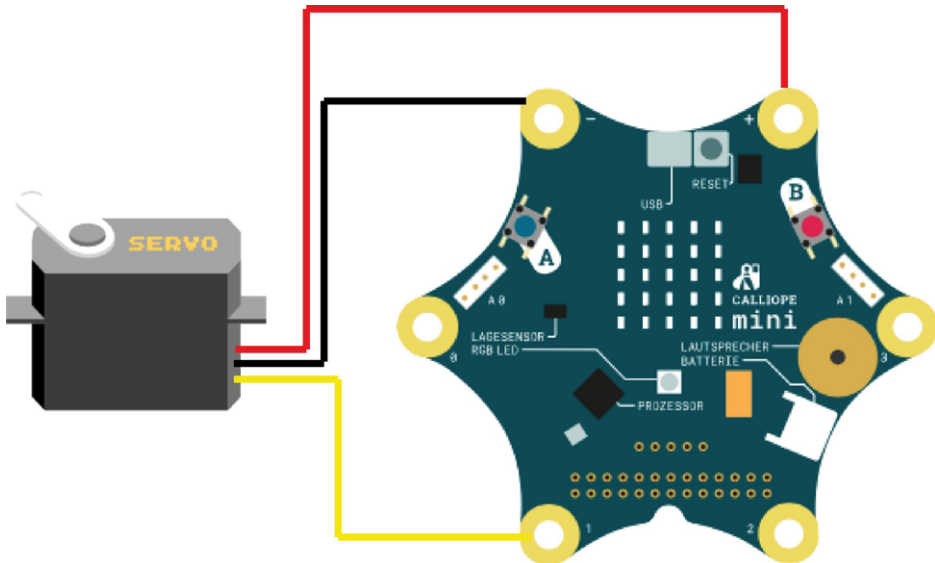


Abb. 10–12 So schließt du einen Servomotor an den Calliope mini an

Mit einem einfachen Programm kannst du deinen Servomotor nun in Bewegung setzen. Hierzu ziehst du den Programmierbefehl »beim Start« auf die Programmierfläche und ergänzst dein Programm anschließend um den Programmierbefehl »schreibe Servo an Pin P1 auf 180«. Die Programmierbefehle für den Servomotor verstecken sich in der Gruppe »Fortgeschritten« der Programmierbefehle und dort wiederum in der Gruppe »Pins«. Fertig. Dieser Programmierbefehl sorgt dafür, dass dein Servomotor eine halbe Umdrehung macht.



Abb. 10–13 Der Servo am Calliope mini macht eine halbe Umdrehung

- ▶ Wie kannst du das Programm so abändern, dass der Servomotor sich wieder an die Anfangsposition zurückbewegt?
- ▶ Versuch doch, deinen Servomotor einmal »winkeln« zu lassen, indem du das Programm so abänderst, dass sich der Servomotor-Arm ständig hin und her bewegt.
- ▶ Wie musst du dein Programm anpassen, damit zwei Servomotoren gleichzeitig »winkeln«?

## 10.2 Der Calliope-Tee-Roboter

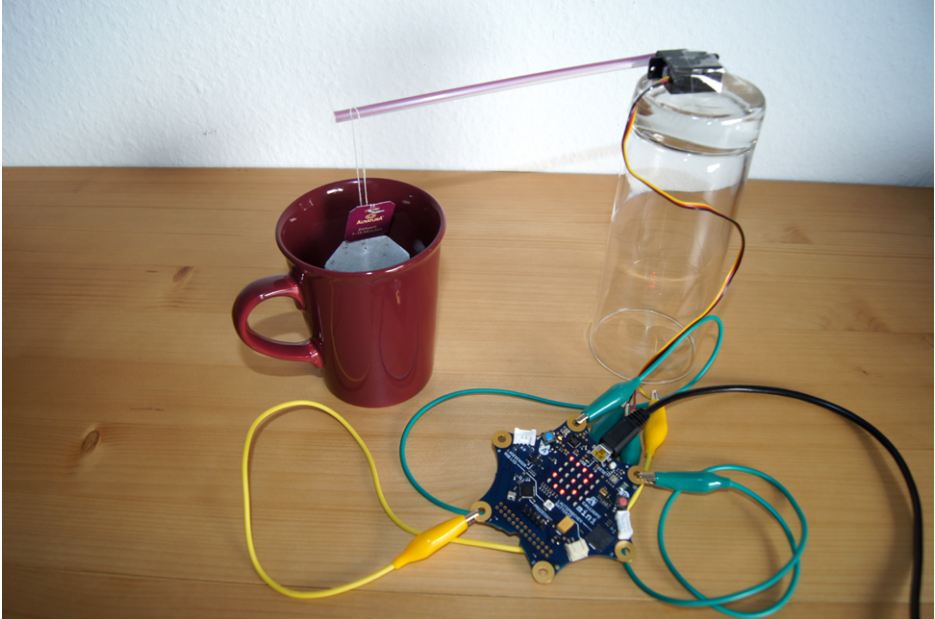


Abb. 10–14 Der Calliope-Tee-Roboter im Einsatz

Du hast bereits gelernt, wie dein Calliope mini einen Servomotor steuern kann. Jetzt ist es an der Zeit, das erste richtige Projekt mit einem Servomotor zu basteln. Sicher hast du dir schon einmal einen Tee zubereitet. Du weißt, dass man dazu eine Tasse mit heißem Wasser benötigt, in die man einen Teebeutel legt. Danach wartest du ein paar Minuten und nimmst den Teebeutel wieder aus dem heißen Wasser. Dabei musst du ständig die Zeit im Auge behalten, damit du den Teebeutel genau zum richtigen Zeitpunkt aus der Tasse ziehen kannst. Genau diese Aufgabe kannst du bequem von deinem Calliope erledigen lassen. Du benötigst dazu nur wenige Dinge:

- ▶ einen Servomotor
- ▶ ein großes und ein kleines Gefäß (Gläser oder Tassen)
- ▶ einen Trinkhalm
- ▶ etwas Klebeband
- ▶ eine Schere
- ▶ 3 Krokoklemmen

Bevor du mit der Programmierung des Calliope minis beginnen kannst, musst du noch ein wenig basteln. Du beginnst damit, den Servomotor vorzubereiten. Knipse hierzu mit einer Zange eine Seite des Servo-Arms ab.



Abb. 10–15 Ein Servo-Arm wird entfernt

Als Nächstes knipst du mit der Zange den Stecker am Servomotor-Kabel ab. Spreize jetzt die Kabel auseinander. Anschließend kannst du die Kabelenden mit einer Zange entmanteln. Die Kabel sind sehr dünn, deshalb musst du ganz vorsichtig vorgehen.

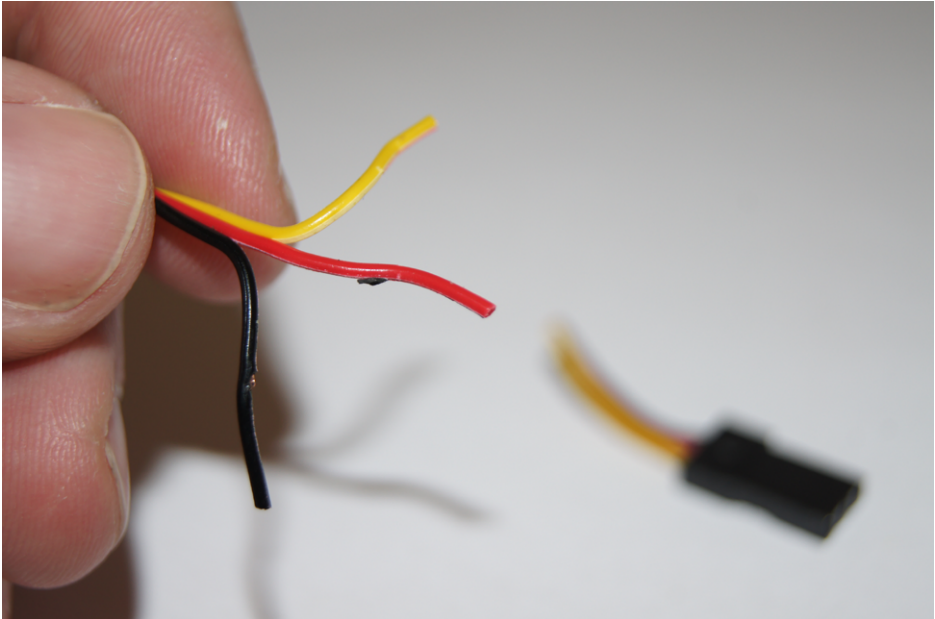


Abb. 10–16 Die gespreitzten Kabel sind bereit zum Entmanteln.



Abb. 10–17 Die Kabelenden werden entmantelt.

Den schwierigen Teil hast du zum Glück geschafft. Als Nächstes klebst du den Servomotor mit Klebeband so auf das größere Gefäß – hier eine Tasse –, dass der Servo-Arm nach oben zeigt.

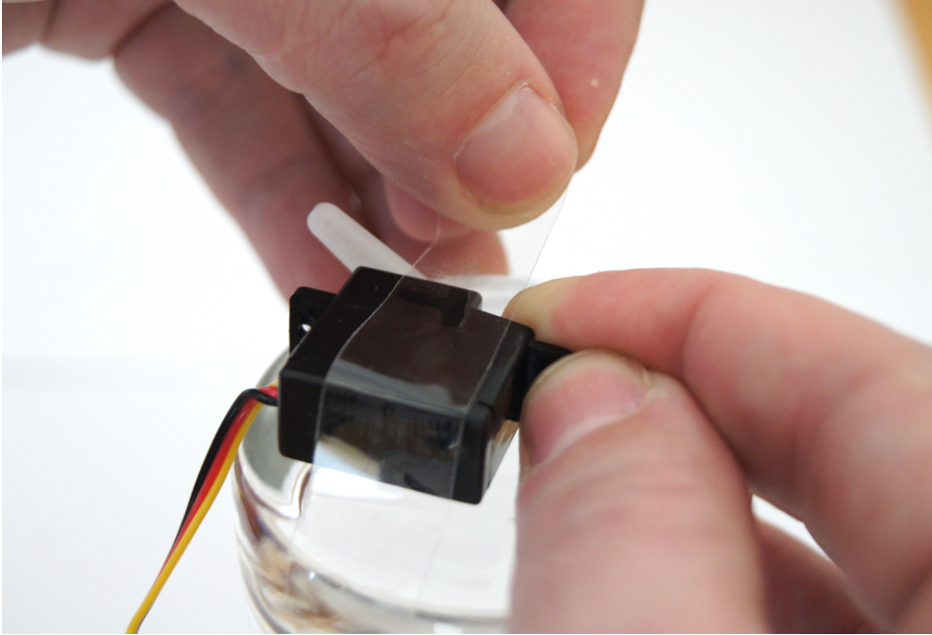


Abb. 10–18 Der Servo wird an der Tasse befestigt.

Schneide jetzt die Knickstelle an dem Trinkhalm ab. Damit du den Teebeutel später besser befestigen kannst, schneidest du den Trinkhalm an einem Ende ca. 1 cm weit ein. Dann klebst du den Trinkhalm mit dem anderen Ende an den Servo-Arm.

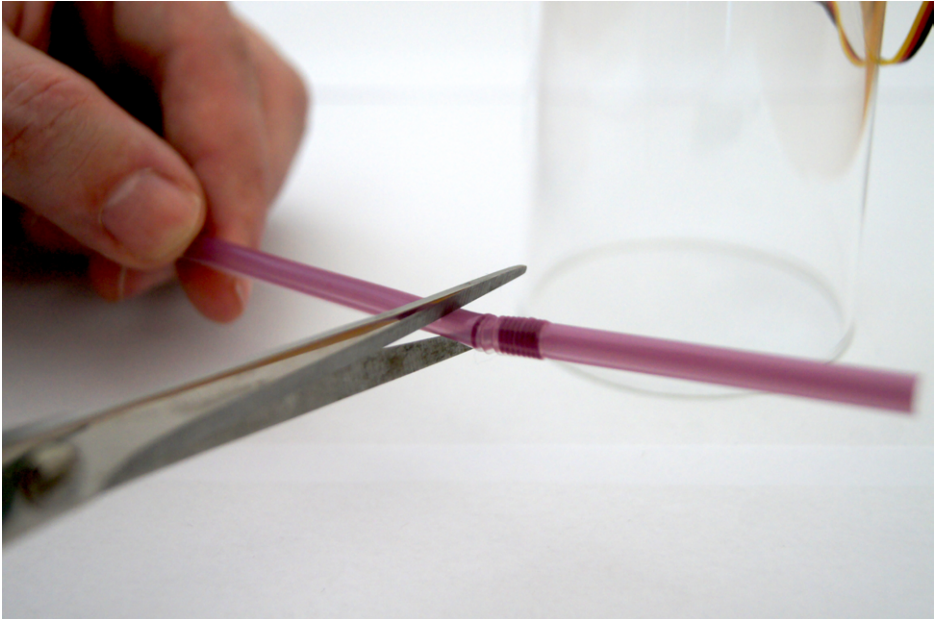


Abb. 10–19 Der Strohhalm wird präpariert.



Abb. 10–20 Der Strohhalm wird ca. einen Zentimeter eingeschnitten, ...

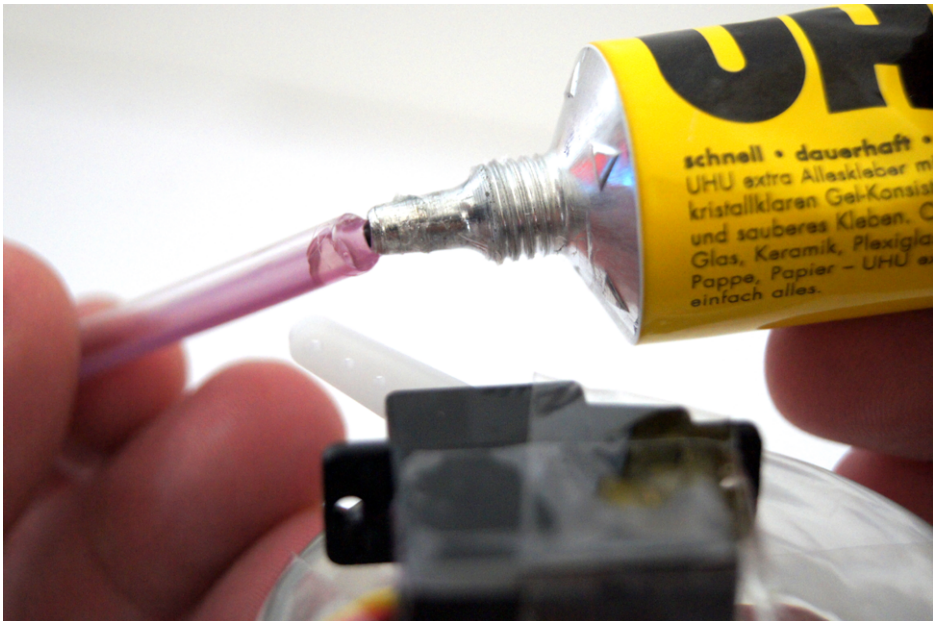


Abb. 10–21 ... mit Kleber versehen ...

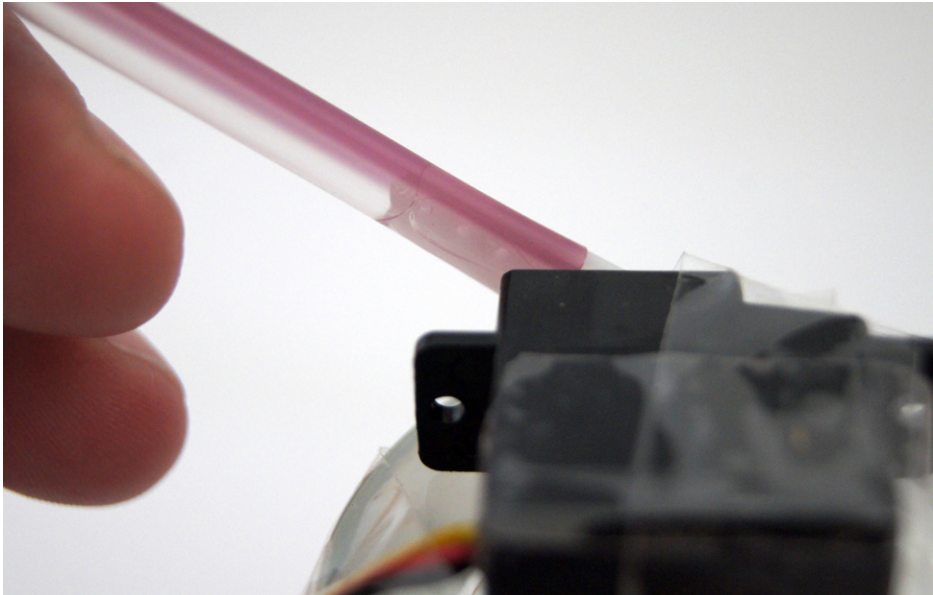


Abb. 10–22 ... und schließlich am Servo-Arm befestigt.

Um den Tee-Roboter fertigzustellen, musst du noch den Calliope mini am Servomotor anschließen. Nimm die Krokodilklemmen zur Hand und verbinde jedes Kabel des Servomotors mit einer Krokodilklemme. Die anderen Enden werden

am Calliope mini angeschlossen. Das gelbe Servomotor-Kabel muss an den Pin1 des Calliope-Boards geklemmt werden. Das schwarze Kabel musst du an »-« am Calliope mini anschließen. Es bleibt noch das rote Kabel, das du an »+« an deinem Calliope anschließt.

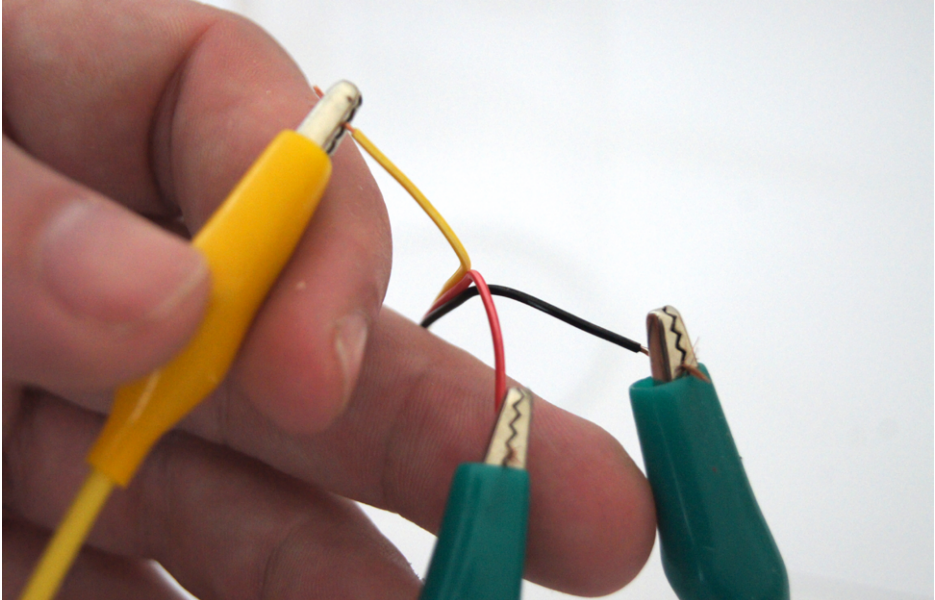


Abb. 10–23 Beachte die Kabelfarben beim Anschließen

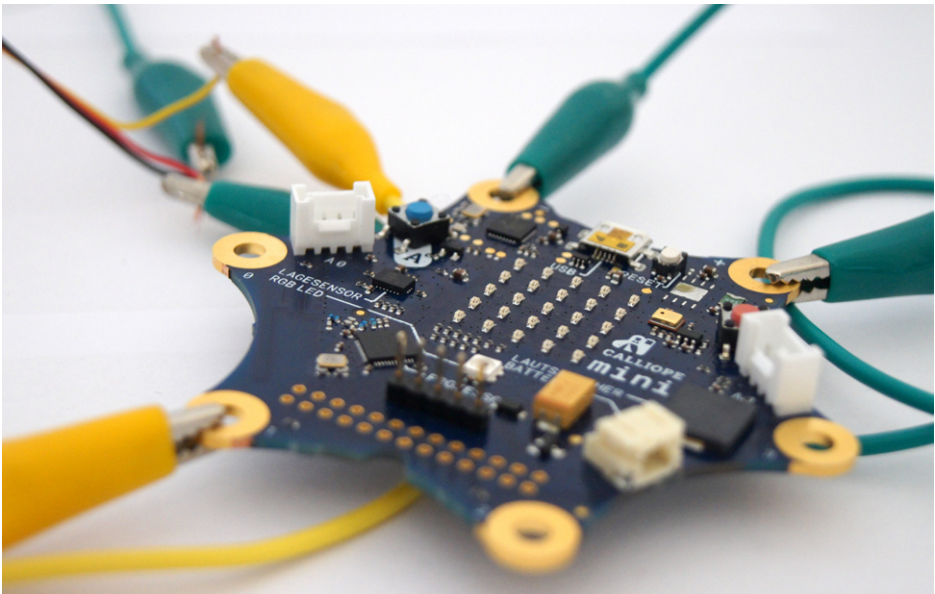


Abb. 10–24 Und so werden die Krokodilsklemmen an den Calliope mini angeschlossen.

Am eingeschnittenen Ende des Trinkhalms kannst du nun einen Teebeutel befestigen. Darunter stellst du eine Tasse. Der Aufbau des Tee-Roboters ist jetzt fertig.



Abb. 10–25 Der fertige Aufbau

Gut, du hast alles vorbereitet. Wie immer musst du deinem Calliope mini nun noch in Form eines Programms sagen, was er tun soll. Da du den genauen Startzeitpunkt selber festlegen möchtest, beginnst du mit dem Programmierbefehl »wenn Knopf A gedrückt«.

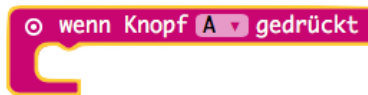


Abb. 10–26 Die Programmierung geht los

Nachdem der Knopf A gedrückt wurde, soll dein Calliope also den Servomotor bewegen. Da du den Teebeutel zu Beginn des Programms in die Tasse absenken möchtest, beginnst du mit dem Programmierbefehl »schreibe Servo an Pin P1 auf 0«. Am Ende möchtest du, dass der Teebeutel wieder aus der Tasse gezogen wird. Hierzu verwendest du den gleichen Programmierbefehl noch einmal, allerdings änderst du die Zahl der Servo-Position auf 30.

Allerdings änderst du die Zahl der Servo-Position. In Abb. 10–27 wird die Servo-Position 180 gewählt. Da die verwendeten Gläser unterschiedliche Höhen haben können, musst du hier mit der Servo-Position etwas experimentieren. Eine gute Startposition könnte zum Beispiel 30 sein.



Abb. 10–27 Der Teebeutel wird in die Tasse gesenkt und wieder herausgezogen.

Wenn du dieses Programm nun ausprobierst, wirst du feststellen, dass der Teebeutel sofort sehr schnell in die Tasse abgesenkt und auch sehr schnell wieder herausgezogen wird. Du brauchst also noch eine Pause zwischen den beiden Servomotor-Programmierbefehlen. Um eine Pause zu erzeugen, fügst du zwischen die Servomotor-Programmierbefehle einfach den Programmierbefehl »pausiere (ms) 1000« und änderst die Zahl 1000 auf 60000.

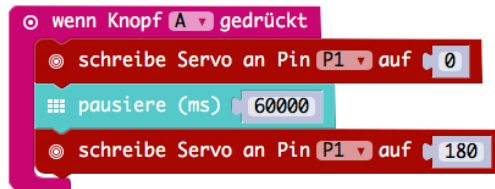


Abb. 10–28 Mit »pausiere...« legst du die Zeit fest, wie lange der Teebeutel abgesenkt bleiben soll.

## Etwas über Zeit

Mit dem Programmierbefehl »pausiere« kannst du die Zeit in ms (Millisekunden) angeben. Es gilt also: 1000 Millisekunden entsprechen einer Sekunde. Somit entsprechen 60000 Millisekunden einer Minute.

Wenn du das Programm nun ausprobierst, wirst du feststellen, dass dein Servomotor den Beutel in die Tasse absenkt, dein Calliope mini genau eine Minute wartet – und erst dann wird der Teebeutel vom Servomotor wieder aus der Tasse gezogen.

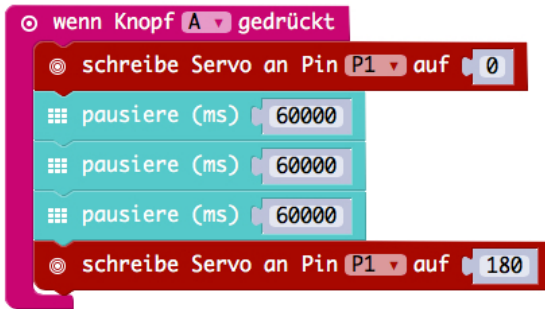


Abb. 10–29 Dreimal 6000 Millisekunden ergeben ...?

Du kannst nun die Zeit verlängern, indem du mehrere »pausiere (ms) 60000«-Programmierbefehle hintereinander einfügst. Das Programm in der Abbildung wartet genau drei Minuten, bis der Servomotor den Teebeutel aus der Tasse zieht.

## 10.3 Das lichtscheue Borstentier

Das Calliope-Borstentier ist ein sehr lichtscheuer Mikrocontroller. Immer wenn es zu hell wird, wacht das Borstentier auf und krabbelt so lange herum, bis es ein dunkleres Versteck gefunden hat. Erst dann kann das Calliope-Borstentier in Ruhe weiterschlafen.

Möchtest du auch gerne so ein Tierchen in deinem Zimmer herumkrabbeln lassen? Kein Problem! Alles, was du wissen musst, um dein eigenes Borstentier zu programmieren und zusammenzubauen, findest du auf den nächsten Seiten.

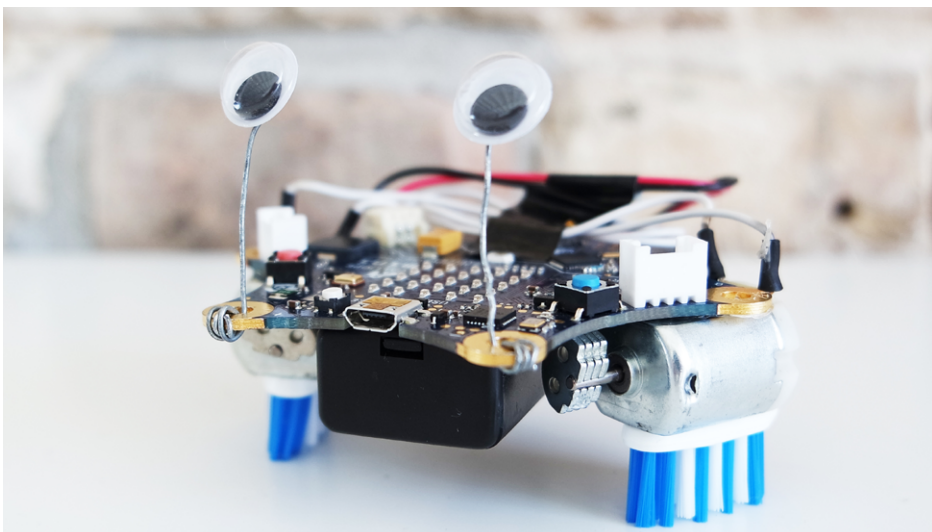


Abb. 10–30 Das lichtscheue Borstentier

### 10.3.1 Materialien und Werkzeuge

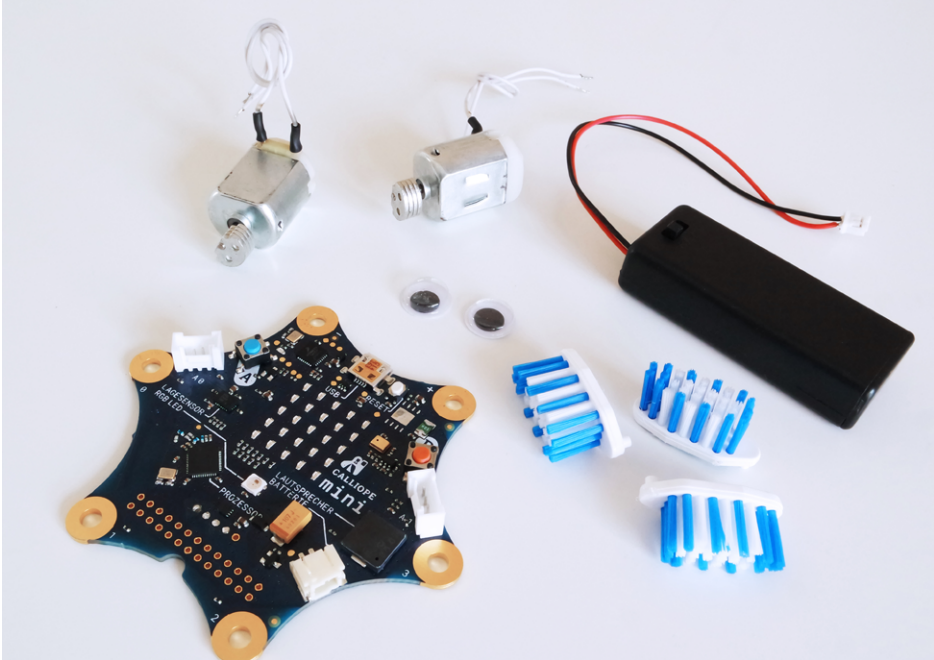


Abb. 10–31 Materialien für das Borstentier

Wie viele?	Was	Anmerkung
1	Calliope	
1	Battery-Pack	
2	Batterien	
2	Vibrationsmotor	4 cm
2	Zahnbürstenkopf	
1	LötKolben	
1	Lötzinn	
1	Zange	
1	Heißklebepistole	mit Patrone
2	Kugelaugen	

Tab. 10–1 Materialien für das Calliope-Borstentier

## 10.3.2 Wie funktioniert das Borstentier?

Immer wenn es dem Borstentier zu hell wird, sucht es sich einen dunkleren Platz. Um mit Helligkeit und Dunkelheit zu arbeiten, kennst bereits den Lichtsensor. Damit kannst du die Lichtwerte messen und in dein Programm einbinden. Wie kann sich aber dein Calliope durch den Raum bewegen? Es hat ja schließlich keine Beine ...

In diesem Kapitel hast du schon einiges über Motoren erfahren. Auch für das Borstentier werden wir Motoren verwenden, nämlich zwei Vibrationsmotoren. Diese Vibrationsmotoren werden mit dem Calliope mini verbunden und auf die Unterseite des Boards geklebt. Unter die Motoren werden nochmals zwei Zahnbürstenköpfe geklebt.

Und nun macht das Programm Folgendes: Immer wenn der Lichtsensor einen zu hohen Helligkeitswert misst, lässt der Calliope mini die Motoren vibrieren. Die Vibration überträgt sich auf Zahnbürstenköpfe und das Borstentier setzt sich in Bewegung. Sobald der Lichtsensor Dunkelheit misst, hören die Motoren auf zu vibrieren. Das Tierchen bleibt stehen und kann weiterschlafen.

## 10.4 Der Vibrationsmotor

Vibrationsmotoren werden im Alltag häufig als Signalgeber eingesetzt. Zum Beispiel im Smartphone, wenn neue Nachrichten oder Anrufe eintreffen. Oder für Feedback, wenn Eingaben über den sogenannten Touchscreen erfolgen. Und auch Lautsprecher haben einen Vibrationsmotor, der Schwingungen erzeugt, die wir hören können. In diesem Projekt verwenden wir einen etwas größeren und somit stärkeren Vibrationsmotor, damit dein Borstentier genug Kraft hat, herumzukrabbeln.

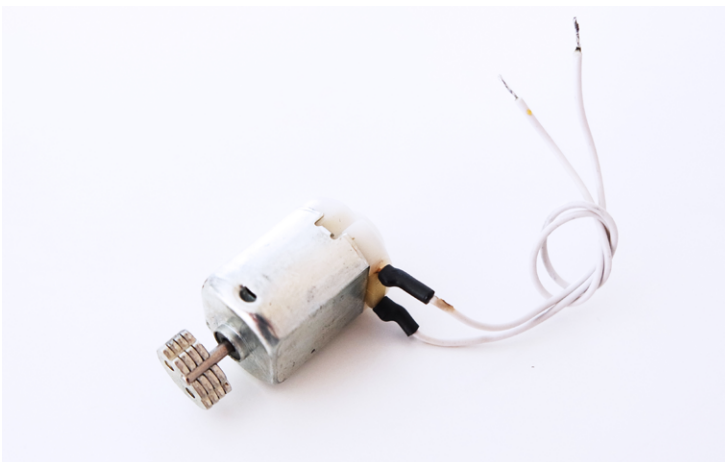


Abb. 10–32 Der Vibrationsmotor

## Wie funktioniert ein Vibrationsmotor?

Es gibt unterschiedliche Arten von Vibrationsmotoren, aber in ihrer Funktionsweise sind sie sich sehr ähnlich. Immer dann, wenn der Motor mit Strom versorgt wird, erzeugt er eine Unwucht. Von einer Unwucht spricht man, wenn sich ein Körper ungleichmäßig dreht oder schwingt. Durch diese ungleichmäßige Bewegung entsteht am Ende die Vibration.

Das klingt jetzt etwas kompliziert, deswegen schauen wir uns ein Beispiel an: Ein Rad am Auto ist rund und dreht sich normalerweise gleichmäßig. Das geschieht, weil das Gewicht des Rades überall gleichmäßig verteilt ist. Die obere Seite ist also nicht schwerer als die untere Seite usw. Außerdem sind die Schrauben, mit denen ein Rad an der Achse befestigt ist, gleichmäßig fest angezogen. Fährt das Auto, drehen sich alle vier Räder gleich. Ist aber ein Rad nicht richtig angebracht, kann sich das Gewicht nicht gleichmäßig verteilen und somit das Rad nicht gleichmäßig drehen. Das Rad fängt an zu »eiern« und gerät sozusagen »aus der Bahn«. Spätestens wenn das Lenkrad anfängt zu vibrieren, merkt auch der Fahrer die Unwucht und sollte unbedingt seine vier Räder überprüfen. Frag doch mal einen Autofahrer, ob er bereits Erfahrungen mit einer Unwucht beim Autofahren gemacht hat.

## 10.4.1 ... und jetzt zur Programmierung deines Borstentiers

Du verwendest zum Programmieren den Online-PXT-Editor, den du unter folgendem Link erreichst: <https://mini.pxt.io/> Solltest du zum Editor noch Fragen haben, findest du dazu in Abschnitt 2.2.1 eine Anleitung. Wenn du mit der Umgebung schon vertraut bist, umso besser. Legen wir gleich los:

### 10.4.2 Zu Beginn

Wie du sicherlich bereits weißt, braucht jedes Projekt einen guten Namen. Wie wäre es mit Borstentier-Programm? So kannst du es leicht von anderen Programmen unterscheiden. Schreib den Namen in die Box



»Wähle einen Namen ...«, die du unten links findest. Danach ziehst du den Grundlagenblock »dauerhaft« in dein Programmierfeld.

Abb. 10–33 Programm starten

### 10.4.3 Die Logik im Spiel

Immer wenn es hell ist, dann soll dein Borstentier krabbeln. Ansonsten soll es schlafen. Das klingt doch sehr nach einem »Logik«-Programmierbefehl, oder? Genau diesen Befehl findest du im »Logik«-Menü. Wähl ihn aus und schieb ihn in deinen »dauerhaft«-Block hinein.

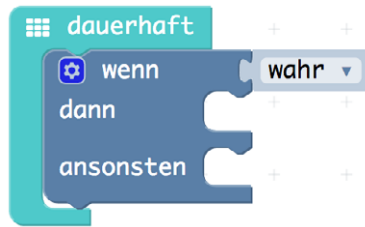


Abb. 10–34 »Logik«- Programmierbefehl einsetzen

### 10.4.4 Die Helligkeitswerte vergleichen

Im nächsten Schritt baust du einen »Vergleichs«-Programmierbefehl ein, um die gegebene Helligkeit mit deinem vorgegebenen Wert zu vergleichen. Auch diesen Programmierbefehl findest du im Menü »Logik«.

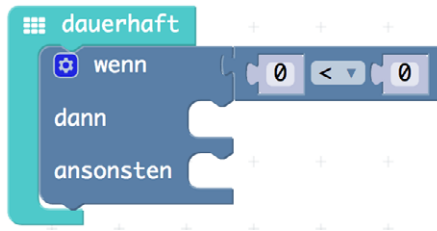


Abb. 10–35 Vergleich programmieren

Danach setzt du den »Lichtstärke«-Block ein, der den Lichtsensor die Helligkeitswerte auslesen lässt.

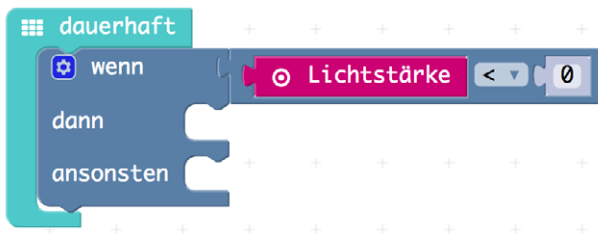


Abb. 10–36 Lichtstärke einsetzen

Zum Schluss musst du noch einen Wert zwischen 0 und 255 festlegen, der dein Borstentier zum Krabbeln bringt. In diesem Fall ist es der Wert 50. Immer wenn der Lichtsensor einen Wert misst, der größer als 50 ist, sucht sich das lichtscheue Tier einen dunkleren Ort. Findet es einen Platz mit einem Helligkeitswert unter 50, hört es auf zu krabbeln.

Hier kannst du gerne verschiedene Werte ausprobieren, bis du einen findest, der für dein Projekt genau richtig ist.



Abb. 10–37 Lichtstärke einsetzen

### 10.4.5 ... und jetzt zu den Motoren

Jetzt programmierst du die beiden Motoren. Hierfür setzt du den »Motor-A-mit-0«-Block hinter das »dann« im »Logik«-Programmierbefehl.

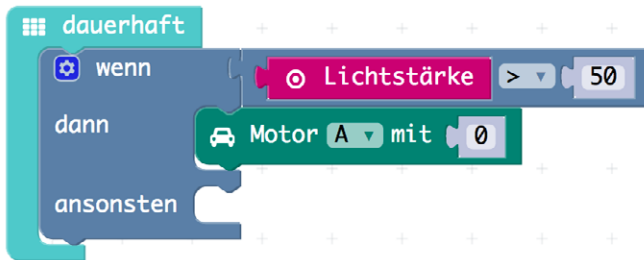


Abb. 10–38 »Motor A« einsetzen ...

Genau den gleichen »Motor«-Programmierbefehl setzt du noch einmal darunter. Mit einem Doppelklick auf das »A« öffnet sich ein Menü, aus dem du »B« wählst. Jetzt hast du insgesamt zwei Motoren, »Motor A« und »Motor B«, in dein Programm eingebaut.

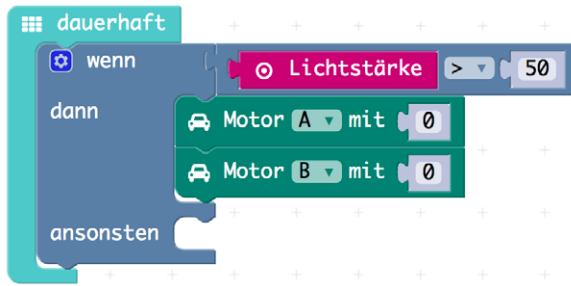


Abb. 10–39 ... und »Motor B«

Beide Motoren sind mit 0 Prozent Leistung vorprogrammiert, das heißt, dass sie einfach aus sind. Zeit, das zu ändern, schließlich sollen sie ja vibrieren. Je nachdem, wie stark sich dein Tierchen bewegen soll, trägst du hier eine Zahl zwischen 1 und 100 ein.

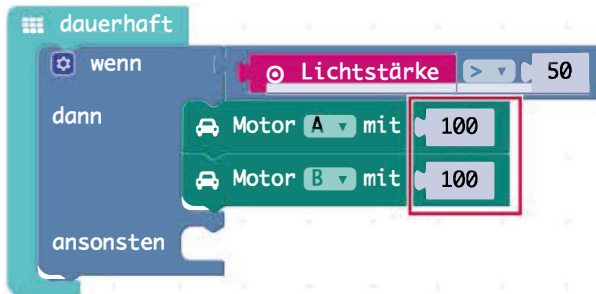


Abb. 10–40 Vibrationsstärke programmieren

## 10.4.6 Und wie schläft das Borstentier wieder ein?

Bisher fangen in deinem Programm beide Motoren an zu vibrieren, wenn es für das Borstentier zu hell wird. Wie können die Motoren nun ausgeschaltet werden, damit das Tier wieder einschlft? Zum Glück haben wir noch ein leeres »ansonsten« in unserem »Logik«-Programmierbefehl stehen. In dieses »ansonsten« baust du einfach zwei Motorblöcke ein und stellst den Wert auf 0 herunter. Vergiss nicht, den zweiten Motor in »Motor B« umzubenennen.

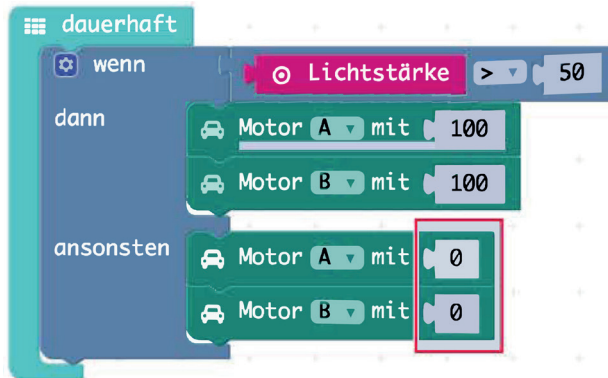


Abb. 10–41 Das fertige Borstentier-Programm

Wunderbar, dein lichtscheues Borstentier-Programm ist fertig! Jetzt speichert du das Programm wie gehabt auf deinem Computer ab und kopierst es auf deinen Calliope, der über ein USB-Kabel mit deinem Computer verbunden ist.

### 10.4.7 Das Borstentier basteln

Die Programmierarbeit ist getan. Aber jetzt musst du deinen Calliope mini noch irgendwie in ein Borstentier verwandeln, oder? In den nächsten Schritten erfährst du, wie leicht sich das Tier zusammenbauen lässt.

### 10.4.8 Anschließen der Motoren

Deine zwei Vibrationsmotoren haben genau zwei Kabel. Über das eine Kabel wird der Motor mit Strom versorgt und über das andere Kabel fließt der Strom wieder ab. Hierbei spielt es keine Rolle, welches Kabel des Motors an den Minus- bzw. Pluspol angeschlossen wird.

In der Abbildung kannst du die vier Anschlüsse für die beiden Motoren sehen, die du bereits programmiert hast. Die beiden äußeren Pins sind Minuspole und bei den beiden inneren Pins handelt es sich um Pluspole.

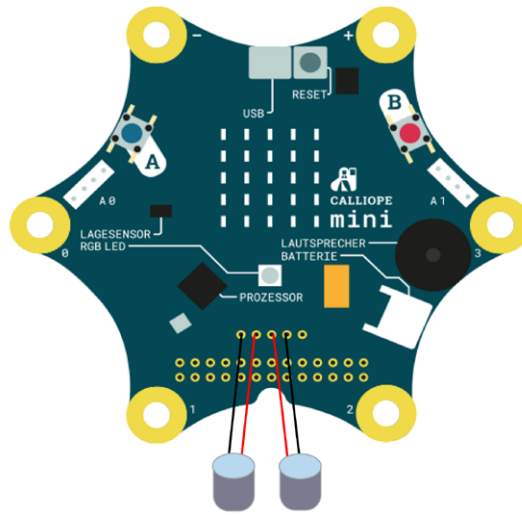


Abb. 10-42 Anschluss der Vibrationsmotoren

Da das ganze Tierchen nachher vibrieren wird, sollten die Kabel gut an dem Calliope-Board befestigt sein. Sie werden also festgelötet. Hierfür wird jeweils das Kabelende entmantelt. Das heißt, dass das Gummi um den Draht herum entfernt wird. Dann werden die Kabelenden in den jeweiligen Pin geschoben und mit einem LötKolben und etwas Lötzinn fixiert.

Damit der LötKolben das Lötzinn schmelzen kann, muss der LötKolben ganz heiß werden. Oft sind es Temperaturen über 300 ° Celsius. Da man sich sehr leicht daran verbrennen kann, solltest du einen Erwachsenen um Hilfe fragen. Bestimmt habt ihr bereits einen LötKolben bei euch Zuhause. Frag einfach einmal nach!

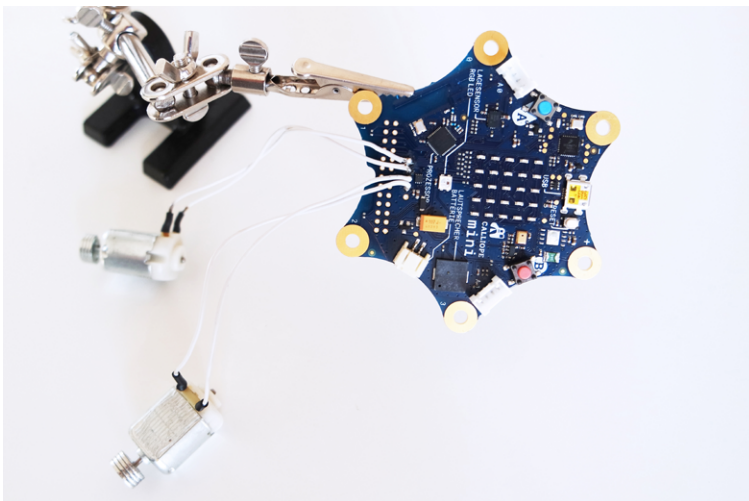


Abb. 10-43 Festlöten der Motorenkabel

## 10.4.9 Befestigen der Motoren und der Batteriebox

Jetzt sind die Motorenkabel an den Calliope mini gelötet, und du solltest die Motoren auf die Unterseite kleben, damit sie die Vibration gut übertragen. Hierfür eignet sich am besten eine Heißklebepistole. Vorsicht: Wie der Name bereits sagt, ist eine Heißklebepistole wirklich heiß!

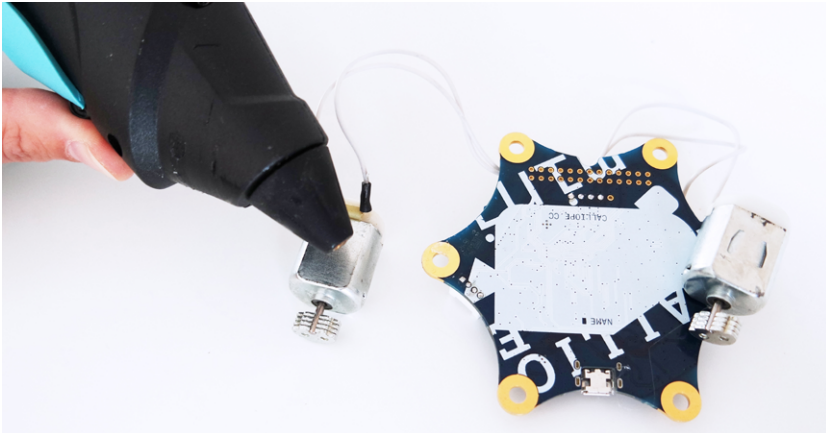


Abb. 10–44 Motoren auf den Calliope mini kleben

Befestige die Motoren, wie in der unteren Abbildung, am äußeren Rand des Boards. So ist zwischen den Motoren noch genügend Platz, um die Batteriehalterung anzubringen. Die Batteriebox wird mit doppelseitigem Klebeband auf das Calliope-Board geklebt, damit sie wieder abgenommen werden kann, um Batterien auszutauschen. Dein lichtempfindliches Borstentier braucht natürlich Strom, damit es funktioniert.

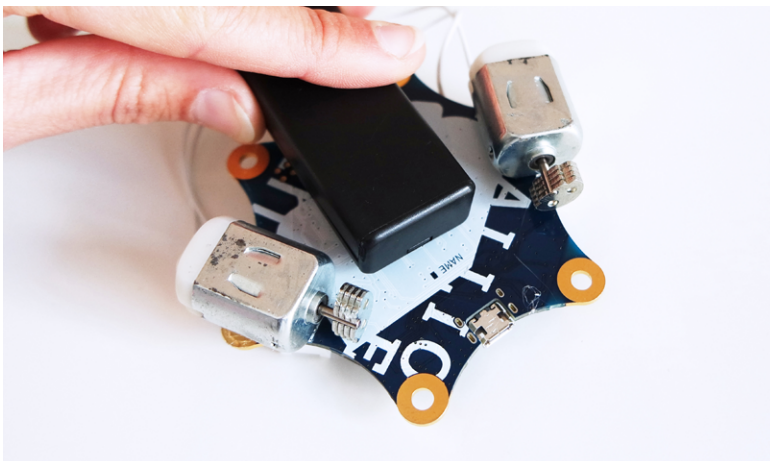


Abb. 10–45 Die Batteriebox wird aufgeklebt

## 10.4.10 Zahnbürstenköpfe anbringen

Für den nächsten Schritt fragst du wieder einen Erwachsenen um Hilfe. Mit einer scharfen Zange trennst du jetzt die Köpfe der zwei Zahnbürsten ab. Aber Achtung: Beim Abtrennen können die Zahnbürstenköpfe ganz schön schnell durch die Gegend fliegen!

### Halte nach Zahnbürstenköpfen ohne Stiel Ausschau:

In manchen Drogeriemärkten werden Zahnbürstenköpfe bereits ohne Bürstenstiel verkauft. Dann müssen sie nicht extra abgetrennt werden. Auch in diesem Projekt wurden Zahnbürstenköpfe ohne Stiel verwendet.

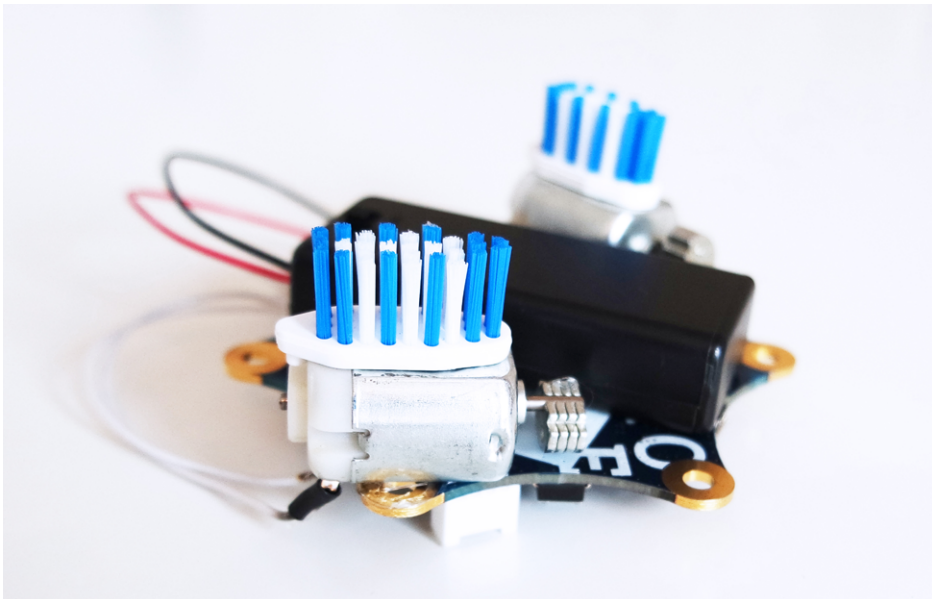


Abb. 10-46 Die aufgeklebten Zahnbürstenköpfe

Um die Bürsten auf die Unterseite der Motoren zu kleben, eignet sich wieder Heißkleber sehr gut. Trage jeweils etwas Kleber auf den Zahnbürstenkopf auf und drücke ihn anschließend so lange auf den Motor, bis der Kleber ausgetrocknet ist.

### 10.4.11 Den Calliope in ein Borstentier verwandeln

Na ja, in ein echtes Tier verwandeln kannst du deinen Calliope mini leider nicht. Aber in etwas Ähnliches – und dafür fehlt ihm noch etwas Wichtiges: die Kugelaugen. Bei diesem Borstentier wurde zuerst etwas Draht um die zwei vorderen Pins gewickelt und anschließend wurden die Kugelaugen aufgeklebt. Jetzt kann das Borstentier sehen, wo es überall entlang krabbelt.

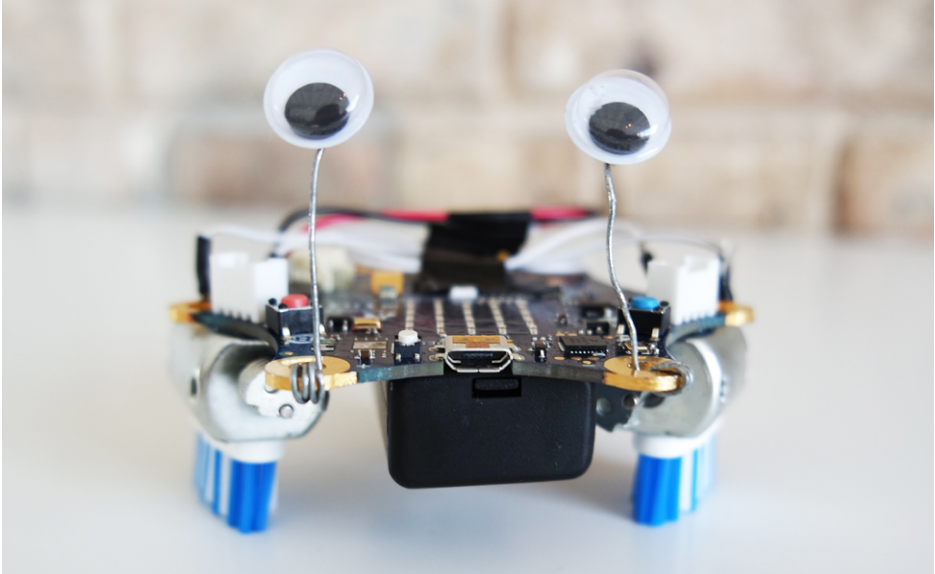


Abb. 10–47 Das fertige Borstentier

Natürlich kannst du dein Tierchen auch so programmieren, dass es anfängt zu krabbeln, sobald es zu dunkel wird. Oder du programmierst doch den Temperatursensor, und das Borstentier reagiert auf Wärme oder Kälte? Es könnte auch anfangen, zu blinken und zu piepsen, sobald es sich in Bewegung setzt ...

In diesem Sinne: Viel Spaß beim Basteln!

## 10.4.12 Was ist noch möglich?

### Es gibt weitere Programmiersprachen

Bevor du die letzten Bastelschritte am Borstentier vornimmst, nämlich die Kugelaugen anbringen, steht noch ein Blick in die Zukunft an, und zwar in deine Zukunft als Programmiererin oder Programmierer.

In diesem Buch hast du alle Projekte mit Hilfe der praktischen Programmiersprache durchgeführt, die aus Blöcken und Blockbefehlen besteht. In dem PXT-Editor, der für alle Projekte dieses Buches verwendet wurde, gibt es eine tolle Möglichkeit, das von dir erstellte Programm auch in einer anderen Programmiersprache darstellen zu lassen.

Eine weitere, weit verbreitete Programmiersprache ist beispielsweise JavaScript. Sie sieht ganz anders aus als das, was du bisher in dem Buch gesehen und gelernt. Javascript besteht aus Codezeile anstatt aus Blöcken.

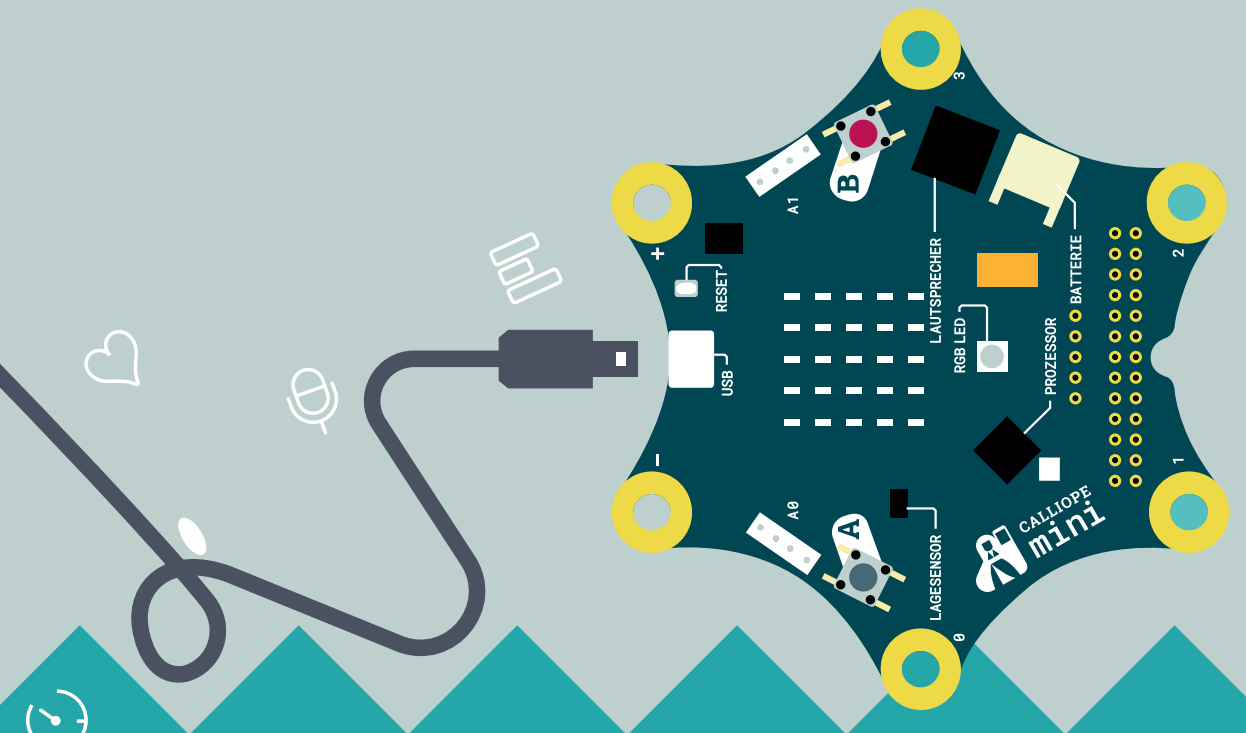
Im PXT-Editor hast du die Möglichkeit, dein Calliope-Programm, das du mit Blöcken erstellt hast, auch in Javascript-Code darzustellen. Du kannst das machen, indem du in der oberen Menü-Zeile rechts neben »Blöcke« den Button »Javascript« anklickst. Nun ändert sich im PXT-Editor die Darstellungsweise und Javascript-Programmiercode erscheint. Du kannst natürlich auch wieder zurück in den anderen Block-Code wechseln.

Spiel etwas damit rum und versuch zu verstehen, wie mit Javascript programmiert werden könnte.

Vieles wird dir dabei unklar bleiben, aber es ist eine tolle – und bequeme – Möglichkeit, zwei verschiedene Programmiermöglichkeiten miteinander zu vergleichen. Probier es mal aus!

- ▶ Versuche einmal, das Programm so zu ändern, dass ein Calliope mini dir durch ein akustisches Signal anzeigt, wie lange es noch läuft.
- ▶ Wie kannst du das Programm mit Hilfe von Schleifen so verändern, dass die Programmierbefehle »pausiere« und »zeige Nummer« nur einmal vorkommen?
- ▶ Du könntest das Programm so verändern, dass du über Knopf B die Zeit einstellen kannst.

Aber jetzt ist erst einmal Zeit für einen leckeren Tee ...



# Anhang

## A.1 Calliope mini, 1. Generation

Es gibt zwei Generationen von Calliope mini-Boards, und dieses Buch funktioniert mit beiden. Ein Board der 1. Generation erkennst du daran, dass das Calliope-Logo rechts unter dem B-Button liegt.

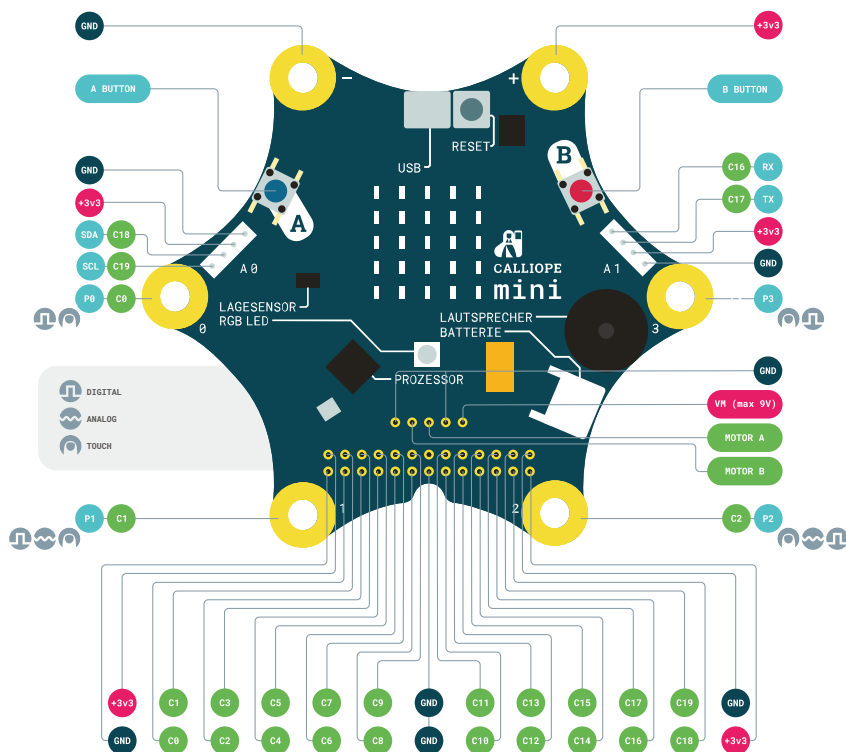


Abb. A-1 Der Calliope mini mit dem Logo rechts

Du kannst in diesem Schaubild alle Bauteile deines Calliope-Boards und die zahlreichen Anschlussmöglichkeiten sehen. Diese Informationen sind wichtig, wenn du weitere Elektronikbauteile – wie beispielsweise Motoren – an dein Calliope-Board anschließen möchtest.

Du kommst über folgenden Link zum passenden PXT-Editor: <https://mini.pxt.io/>.

## A.2 Calliope mini, 2. Generation

Diese Variante des Calliope minis unterscheidet sich nur in technischen Details von den Calliope minis der 1. Generation. Du erkennst dieses Calliope-mini-Board daran, dass das Calliope-Logo links unter dem A-Button angebracht wurde.

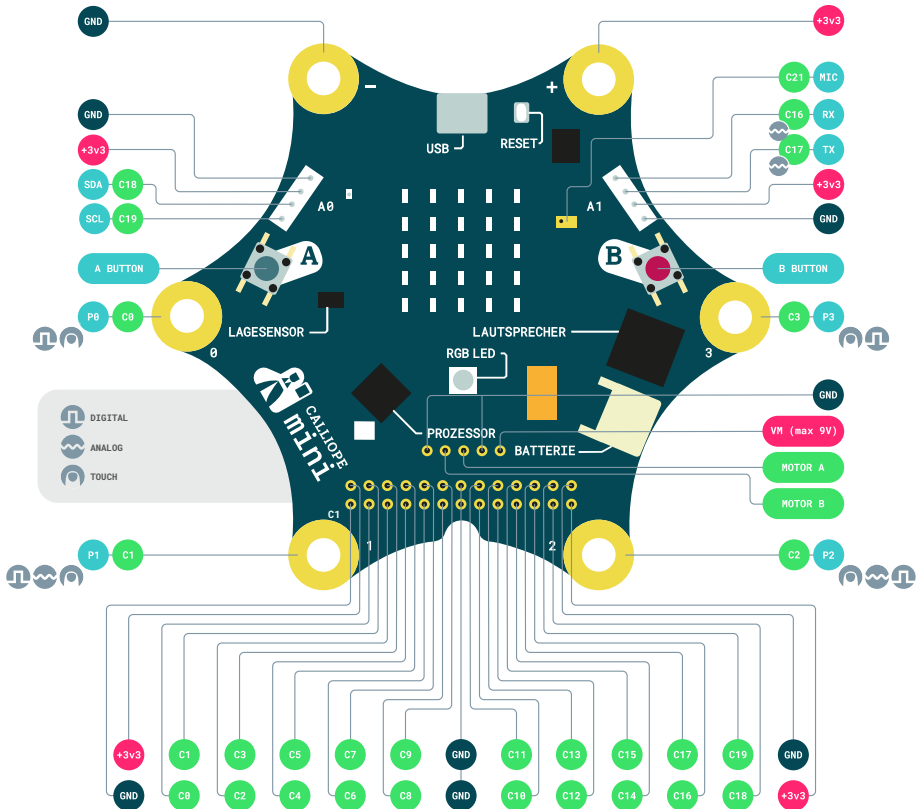


Abb. A-2 Der Calliope mini mit dem Firmenlogo links

Solltest du ein Calliope-Board mit dem Logo links besitzen, dann verwendest du folgenden Link, um zum PXT-Editor zu gelangen: <http://pxt.calliope.cc>.

Weitere Informationen dazu findest du auch auf der Calliope-Webseite unter <https://calliope.cc/editor>.

# Über die Autoren



**Nadine Bergner** studierte die Fächer Informatik, Mathematik und Physik für das Lehramt an Gymnasien und Gesamtschulen an der RWTH Aachen. Seit 2010 leitet sie das InfoSphere – Schülerlabor Informatik, wo 32 Informatikmodule für Kinder und Jugendliche der Klassenstufen 3 bis 13 angeboten werden. Insgesamt konnte sie dort bereits über 8.000 kleine und große Gäste für die Informatik begeistern.



**Patrick Franken** ist gelernter Fachinformatiker und Geschäftsführer der Firma AISLER B.V., einem auf das Prototyping spezialisierten Lieferanten für Platinen und elektrische Bauteile. Vor seiner Tätigkeit bei AISLER hat er über sieben Jahre als Software-Entwickler im Maschinenbau gearbeitet und dabei u.a. ein international gemeldetes Patent im Halbleiterbereich anerkannt bekommen.



**Julia Kleeberger** hat nach Promotion und einer achtjährigen Karriere in der industriellen Forschung das gemeinnützige Unternehmen »Junge Tüftler« gegründet ([www.junge-tueftler.de](http://www.junge-tueftler.de)). Es setzt sich dafür ein, Kindern einen spielerischen Zugang zu digitalen Inhalten zu eröffnen. Neben handwerklichen Fertigkeiten und logischem Denken werden auch Metakompetenzen des 21.Jh. geschult.



**Thimo Leonhardt** promovierte 2015 in der Fachdidaktik Informatik und unterstützt seit fast zehn Jahren Kinder, um diese für informatische und elektronische Themen zu begeistern. Seine Dissertation behandelte die Fragestellung nach der Konzeption eines begabungsfördernden Umfeldes für Mädchen in der Informatik.



**Mario Lukas** hat Informatik an der RWTH Aachen studiert. Er publizierte Artikel zu seiner Maker-Tätigkeit in diversen Fachmagazinen und ist Co-Autor des Buchs »Licht und Spaß«. Hauptsächlich beschäftigt er sich mit den Themen 3D-Scannen und 3D-Druck. Mehrfach konnte er bei nationalen und internationalen Wettbewerben mit seinen Kreationen gute Platzierungen belegen. Mario leitet zur Zeit das FabScanPi 3D-Scanner-Projekt und ist erster Vorsitzender des Vereins Freie Maker e.V.



**Mario Pesch** studiert Chemie und Geographie an der Westfälischen Wilhelms Universität Münster. Neben seinem Studium arbeitet er im Schülerlabor GI@School am Institut für Geoinformatik. Dort schafft er mit der senseBox ([www.senseBox.de](http://www.senseBox.de)) einen einfachen Zugang zu DIY-Messgeräten für Schülerinnen und Schüler und interessierten Bürgerinnen und Bürgern.



**Natalia Prost** arbeitet als Software-Entwicklerin und konzeptionelle Projektleiterin bei den Jungen Tüftlern. Außerdem führt sie mit [www.papiercomputer.de](http://www.papiercomputer.de) eigene Projekte durch, um Informatikwissen spielerisch zu vermitteln und zur kreativen Nutzung von Technologien im schulischen und privaten Kontext anzuregen.



**Jan Thar** kommt aus der Elektrotechnik/Informatik-Ecke und ist dann im FabLab Aachen ([hci.rwth-aachen.de/fablab](http://hci.rwth-aachen.de/fablab)) – ein guter Platz, um verrückte Dinge zu bauen –hängengeblieben. Einige der Basteleien sind bei Instructables zu finden ([www.instructables.com/member/janth](http://www.instructables.com/member/janth)). Jan ist häufig auf Maker-faires anzutreffen.



**Lina Wassong** verbrachte während ihres Studiums der Bekleidungstechnik an der HAW Hamburg einen Auslandsaufenthalt in San Francisco. Dieser inspirierte sie, sich mit der kreativen Seite von Elektronik, Software und Maschinen auseinanderzusetzen. Seitdem gestaltet Wassong, neben ihrer Arbeit als Autorin für Verlage wie O'Reilly, dpunkt und Make, interaktive Fashion-Designs und Accessoires. Mit den Mitteln von Elektronik und digitalen Fertigungsverfahren wie Lasercutting und 3D-Druck präsentiert sie die ästhetischen Seiten von Technik und deren vielfältige Einsatzbereiche.

# Index

## A

Achse 54  
 Alarmanlage 58  
   vernetzt 227  
 Alarmton 62  
 Antenne 220, 222  
 ASCII 145  
   Code 144  
 Ausführungsgeschwindigkeit 161

## B

Batterieanschluss 6  
 Bauteile 3  
 Befehle 13  
   ausgegraut 23  
   Farbe 21  
   Form  
 Befehlsgruppen 13  
 Beschichtung 41  
 Beschleunigung 51, 53  
   Sensor 50, 235  
 Betriebssystem 16  
 Bewegung 269  
   Sensor 6, 50  
 Block 13  
 Borstentier 285  
 BPM 204  
 Browser 16  
 Brummen 261  
 Buchstabengenerator 143

## C

Calliope mini 3  
 Cookie 12

## D

Datentyp 78  
 Dauerschleife 14, 21  
 Download 16  
 Drahtspule 268  
 Duplizieren 120

## E

Editor 11  
 Elektromotor 268  
 Entmanteln 171  
 Entscheidungen 150  
 Explorer 17

## F

Fallunterscheidung 239  
 Fang den Dot 208  
 Fehlermeldung 83  
 Fehlerquellen 25  
 Fernbedienung 231  
 Frequenz 261  
 Funken 220  
 Funkfernbedienung 228  
 Funksignal 251

## G

Getriebe 270  
 Gruppe 228

## H

Hau den Lukas 234  
 Helligkeit 60  
   Sensor 52  
 Herunterladen 16  
 hex-Datei 17  
 Hufeisenmagneten 268

## I

Importieren 28  
 Infrarotlicht 63

## J

JavaScript 297

## K

Klingelton 261  
 Knöpfe 5, 32  
 Kommutator 269  
 Kompass 50  
 Kopfhörer 262

Kreis 256  
 Krokodilklemmen 35, 281  
 Kugelaugen 296

## L

Lagesensor 6, 228  
 Laufschrift 20  
 Laufzeit 99, 238  
 Lautsprecher 5, 61, 260, 263  
 LED  
   abschalten 125  
   anschalten 125  
   Display 5, 15, 144  
 Leitfähigkeit 36  
 Lichtarten 63  
 Lichtsensor 5, 58, 261  
 Lichtwerte 287  
 Litze 171, 264  
 Logik 149, 153  
 Löschen 21  
 Lötten 293  
 Luftgitarre 196

## M

Magnet 269  
 Mapping 178, 187  
 Metronom 98  
 Mikrofon 50  
 Mikroprozessor 3  
 Mikro-USB-Anschluss 6  
 Millisekunde 24, 236  
 Morse-Alphabet 223  
 Motorgeschwindigkeit 273  
 Mülleimer 21  
 Musik 198

## N

Nicht-Block 150

## O

One to many 220  
 Operatoren 67

**P**

Pausiere 93  
 Pin-Belegung 168  
 Pixel  
   schalten 125  
   zeichnen 125  
 Platzhalter 78  
   anlegen 79  
 Pole 269  
 Potentiometer 264  
 Programmieren 3, 10  
 Programmieroberfläche  
 13  
 Programmiersprachen 11  
 PXT-Editor 11, 297

**R**

Radio 220  
 Rauschen 55  
 Reaktionsspiel 126  
 Reichweite 220  
 Reset-Knopf 5  
 RGB-LED 5, 66  
 Richtungswechsel 273  
 Roboterballett 241

**S**

Sanduhr 88  
 Schleifen 92  
   verschachtelte 94  
 Schütteln 119  
 Scratch-Editor. *Siehe* Editor  
 Sekunde 236  
 Sensoren 5, 50  
 Servomotor 270, 274  
 Silberdraht 264  
 Simulator 13  
 SOS 224  
 Speichern 27  
 Stiftleisten 272  
 Stromkreis 46  
 Stromleiter 42

**T**

Teebeutel 279  
 Tee-Roboter 276  
 Temperatursensor 50, 51  
 Theremin 260  
 Thermometer 65  
 Tischbasketball 166  
 Tischbillard 165  
 Töne 25

**U**

Übertragen 16  
 Ultraviolette Strahlung 63  
 Unsichtbares Licht 63  
 Unwucht 288

**V**

Variable 78  
 Ventilator 274  
 Verschachtelte Schleifen  
 94  
 Vibration 287  
   Motor 287

**W**

Während-mache-Befehl  
 156  
 Wahrheitswert 153  
 Wasserwaage 53  
 Wenn-Dann-Befehl 33  
 Widerstand 263, 264  
 Winkel 228  
 Winkelmessung 228  
 Wortschatzspiel 143  
 Würfel 106  
 Würfelglück 154  
 Würfeln 157

**X**

X-Achse 201

**Z**

Zahnbürstenkopf 295  
 Zahnräder 270  
 Zeichenfolge 23  
 Zeichnen (Pixel) 125  
 Zeit 127, 284  
 Zufallszahlen 106