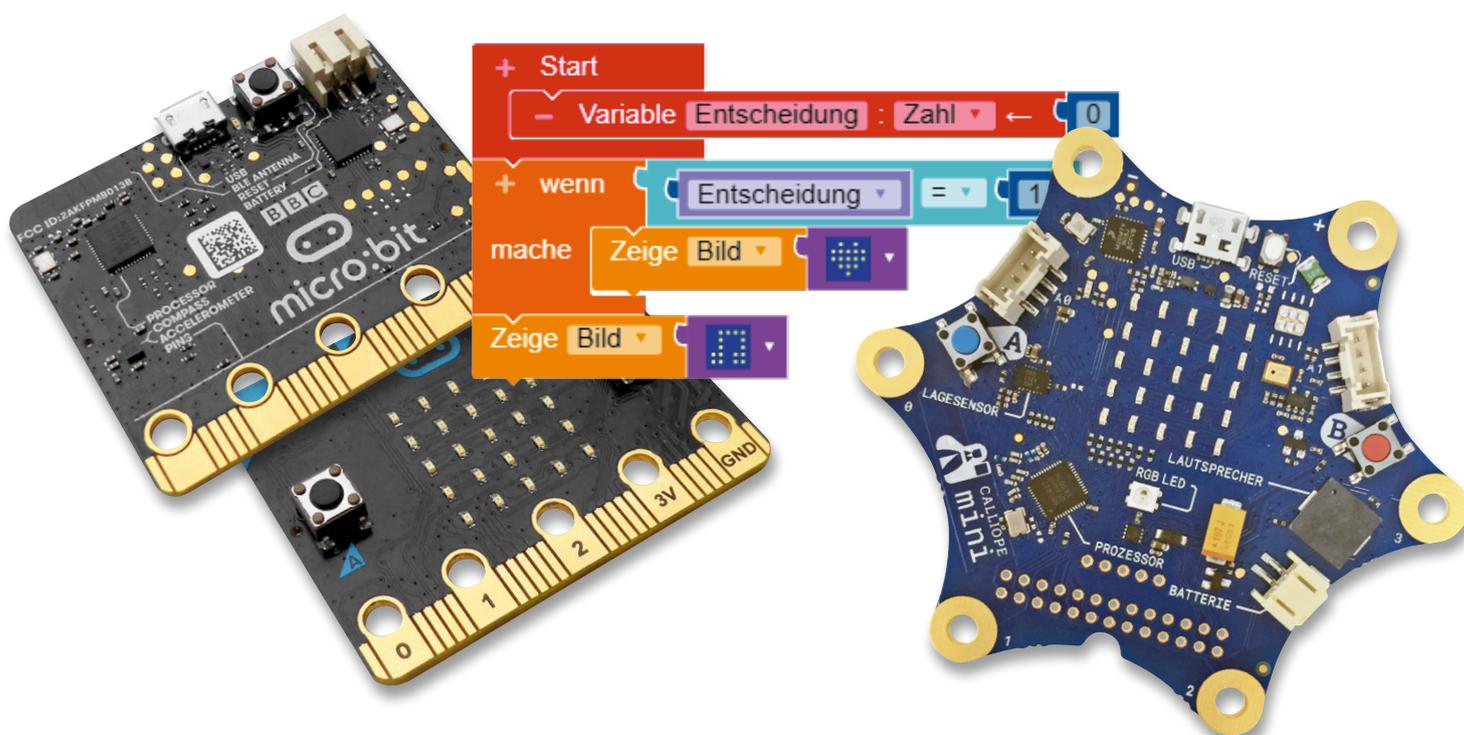


Einsteigen ganz einfach

Lern-Mikrocomputer BBC micro:bit und Calliope mini

BBC micro:bit und Calliope mini sind kleine, recht komplett ausgestattete Mikrocomputer-Plattformen, die, ergänzt durch auch für jüngere Schüler leicht beherrschbare-Code-Editoren, speziell für den Einsatz im Bildungsbereich konzipiert sind. Sie erlauben den ad-hoc-Einstieg in die Computerprogrammierung, machen es Lehrkräften einfach, Schülern das Programmieren nahezubringen. Eine umfangreiche Unterstützung renommierter Organisationen liefert dabei reichlich Rückenwind. Inzwischen sind die Plattformen wegen der reichhaltigen Ausstattung auch für den programmierenden Elektroniker interessant – wir betrachten Technik, Programmier- und Erweiterungsmöglichkeiten.



Hauptziel Bildung

Viele Organisationen, im Englischen „Foundations“ genannt, sind schon seit Jahren engagiert, wenn es darum geht, den technischen Nachwuchs an die moderne und zukünftig alles beherrschende Welt der Computer und Roboter heranzuführen. Schon zu den Zeiten der ersten 8-Bit-Mikrocontroller gab es dazu Ansätze, man denke hier nur einmal an Beispiele wie den NDR-Computer auf Z80-Basis oder BASICCODE aus den 1980er-Jahren.

Rechnersysteme wurden immer komplexer, die Programmier-Hochsprachen ebenfalls – für die breite Masse der Anwender und für möglichen Techniker-Nachwuchs zumindest nicht auf Anhieb beherrschbar. Zwar tauchten mit dem Aufkommen der Mikrocontroller vom Schlage der AVR und PIC dazu passende Coding-Plattformen auf, die es auch Einsteigern ein-

facher machen sollten, ihre eigenen Lösungen zu programmieren, hier sei nur stellvertretend BASCOM genannt, aber das physische Programmieren, Programmiergeräte und die selbst aufzubauende Peripherie blieben das klassische Feld kundiger Elektroniker. Gerade Letzteres war auch eine Hürde gerade für den Bildungsbereich, der immer wieder nach einsatzfertiger Hardware verlangte.

Entscheidende Durchbrüche wurden, nach vielen proprietären Einplatinensystemen, mit der Konzeption der Arduino-Plattform und später mit dem Raspberry Pi erreicht. Beide wurden begleitet von leicht beherrschbaren Programmierplattformen, die auch das physische Programmieren des Mikrocontrollers automatisierten, sodass man sich nur auf die Aufgabe der Programmerstellung konzentrieren musste. Insbesondere die Arduino-IDE war und ist hier eine genial einfache Grundlage, inzwischen erheblich breiter aufgestellt, so auch für andere Systeme wie ARM-Plattformen oder die heute so beliebte ESP-Plattform. Nicht umsonst haben auch andere die gängige Struktur der Arduino-IDE aufgenommen. Und es gibt inzwischen zahlreiche weitere Editoren, die das Programmieren



noch einfacher machen wie Scratch, Ardublockly, den Atom-Editor usw. Fortgeschrittene arbeiten dann vielfach heute mit MicroPython, dafür sind zahlreiche Editoren und Versionen verfügbar.

Um allerdings (zunächst) tatsächlich völlig ohne Peripherie, Löten, Anschluss von Sensoren, Anzeigen oder Bedienelementen auszukommen, sind sowohl Arduino als auch Raspberry Pi nicht wirklich geeignet. Sie sind reine Rechnerplattformen ohne und mit Betriebssystemen und zahlreichen Schnittstellen. Um tatsächlich per Plug-and-Play einsteigen zu können, bedarf es einer dazu spezialisierten Plattform, die alles an Bord hat, was man für erste Schritte benötigt: Bedienelemente, Anzeigen, Sensoren, eine einfach realisierbare Stromversorgung und eine ganz einfach nutzbare Schnittstelle für die Programmierung. Erst für die weitergehende Arbeit sind dann zusätzliche Schnittstellen, passende Peripherie etc. gefragt. Mit der genannten Ausstattung ist eine solche Plattform fähig, auf Anhieb und ohne weitere Vorbereitungen in einem Klassenverband eingesetzt werden zu können – gerade auch in der angestrebten Altersklasse ab der dritten Schulklasse, also ab 9 Jahren.

Außerdem müssen sowohl Hardware als auch die Programmierplattform Lehrer und Eltern mit eher geringer Programmier-Affinität, d. h., ohne Vorkenntnisse, ansprechen können. Und die Programmierung musste ohne aufwendige geräteabhängige Installation einer IDE überall möglich sein, d. h. web- und browserbasiert.

Seelenverwandt

Genau diesen Forderungen entsprechen die beiden Lernsysteme „BBC micro:bit“ und das deutsche Pendant „Calliope mini“. Ein Blick in die technischen Daten zeigt auffällige technische Parallelen beider Systeme. Das ist kein Zufall. Beide Systeme wurden 2014/2015 parallel unter aktiver Beteiligung von Lehrkräften entwickelt, wobei der BBC micro:bit die technische Grundlage darstellt.

Der BBC micro:bit wurde tatsächlich von der BBC mit Unterstützung der Lancaster University entwickelt, seinen Namen erhielt er in Anlehnung an ein ähnliches Projekt der 1980er-Jahre, den zusammen mit Acorn entwickelten Bildungscomputer BBC Micro. Später hat man für die weitere Entwicklung, Verteilung und Vermarktung die Micro:bit Educational Foundation gegründet, eine gemeinnützige Bildungstiftung. Inzwischen wurden über eine Million dieser Mikrocomputer an Schulen des Vereinigten Königreiches verteilt, und seit einigen Jahren ist die Plattform

auch frei verkäuflich zu einem vergleichsweise für eine Komplettplattform geringen Preis. Die zentrale Anlaufstelle für alle Informationen zum BBC micro:bit findet man in vielen Sprachen unter [\[1\]](#).

Den BBC micro:bit gibt es als einzelnes Board, als Starter-Set mit Batteriekasten, Batterien und Booklet sowie als Bestandteil zahlreicher Experimentier-Sets, z. B. mit Tinker-Kit-Sensoren oder als Roboter-Kit.

In [Bild 1](#) ist eine Übersicht über die Bestückung und Anschlussbeschaltung zu sehen.

Der Calliope mini, benannt nach der Zeus-Tochter und Schutzgöttin der Wissenschaft, Kalliope, basiert auf der gleichen Prozessorplattform, ist zum BBC micro:bit auch codekompatibel, aber noch einmal besser ausgestattet. Form und ausschließlich einseitige Bestückung sollen ihn besser handhabbar für Kinder, z. B. kurzschlussicherer machen. Das Projekt entstand als Sponsoren- und Crowdfunding-Projekt unter starker Beteiligung von Lehrkräften und des Bundeswirtschaftsministeriums, es wird heute unter dem Dach der gemeinnützigen Calliope gGmbH [\[2\]](#) betreut. Aufgrund der besseren Ausstattung ist der Calliope mini teurer als der BBC micro:bit, für den primär angesprochenen Bildungsbereich sind vergünstigte Angebote verfügbar.

Den Calliope mini gibt es bei [\[2\]](#) in einem praktischen Starter-Kit mit Batteriekasten, Batterien, USB-Kabel, einem kleinen Booklet und Aufklebern, die den Kindern das Identifizieren der Baugruppen auf der Platine erleichtern sollen. Auch ein passendes Kunststoffgehäuse und ein Starter-Kit mit externen Zusatzmodulen sind erhältlich. Zu bemerken ist noch, dass der Calliope mini zwei Grove-Anschlüsse besitzt, an die man die bekannten seeed-Grove-UART-/I²C-Module [\[3\]](#) direkt anstecken kann. Über eine 26-polige Erweiterungsschnittstelle sowie eine 6-polige Motor-Schnittstelle lassen sich zahlreiche eigene Erweiterungen anschließen.

Technische Daten BBC micro:bit/Calliope mini

| | BBC micro:bit | Calliope mini |
|-----------------------------|--|--|
| Prozessor | Nordic nRF51822 | Nordic nRF51822 |
| Core | ARM Cortex-M0 32 Bit | ARM Cortex-M0 32 Bit |
| Takt | 16 MHz | 16 MHz |
| Flash-ROM | 256 kB | 256 kB |
| RAM | 16 kB | 16 kB |
| USB | USB micro, 1.1, Mass Storage | USB micro, 1.1, Mass Storage |
| Bluetooth | BT 4.1 BLE | BT4.0, BLE |
| Funk | Proprietär Nordic Gazell, 2,4 GHz | – |
| Taster | 2x User, 1x System | 2x User |
| Anzeige | 5x5-LED-Matrix, rot, dimmbar, 10 Stufen | 5x5-LED-Matrix, rot, 1x programmierbare RGB-LED (WS2812B) |
| Audio | – | Piezo-Lautsprecher/MEMS-Mikrofon |
| Motortreiber | – | DRV8837 |
| Sensoren | 6-Achsen-Beschleunigung/Kompass, Temperatur (-25 bis +75 °C) | Beschleunigung/Kompass/Gyroskop (Bosch BMX055) |
| GPIO | 19x GPIO, 2x I ² C, 6x Display/Helligkeit, 2x Button, 3x PWM, UART, 6x Analog-Input (10 bit), SPI, 3x Touch-Input | 8–11 GPIO, PWM, 4x Analog-Input, SPI, I2C, 2x Grove (I ² C/Seriell) |
| Versorgungsspannung | 5 V (USB), 1,8–3,6 V (JST) | 5 V (USB), 3,3 V (JST) |
| Stromaufnahme | 120 mA (USB) max. | |
| Stromabgabe via GPIO | 90 mA max. | 100 mA max. |
| Abmessungen | 50 x 40 mm | 85 x 85 mm |

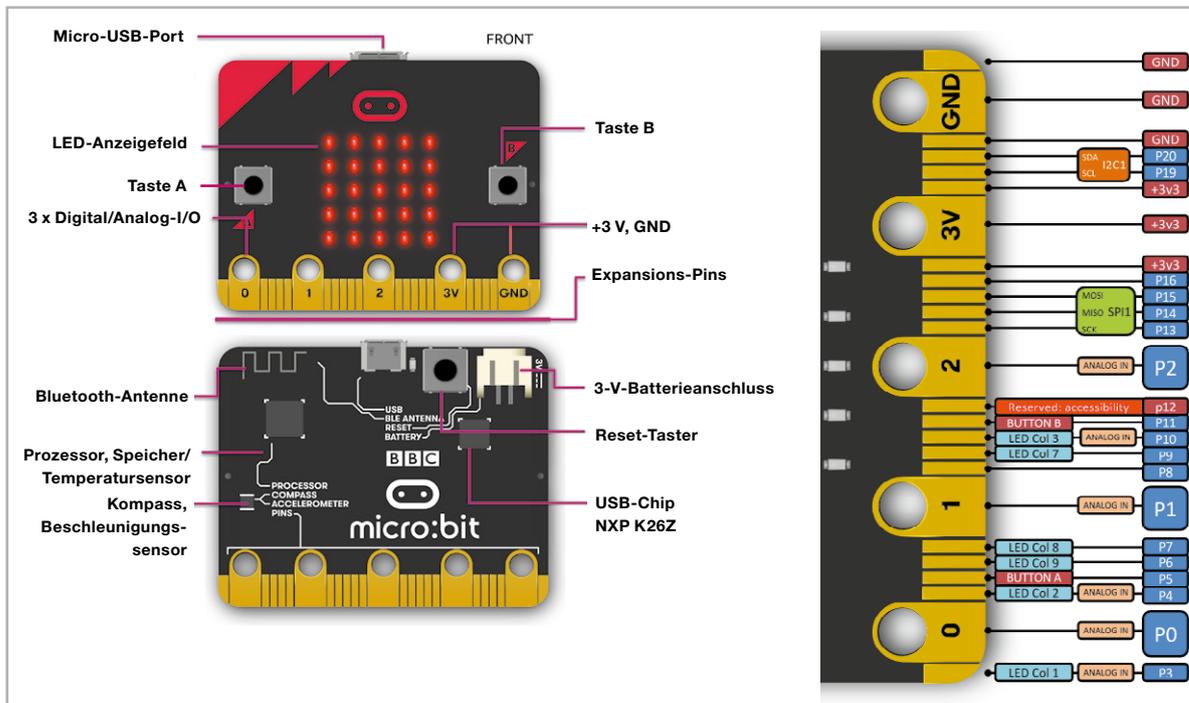


Bild 1: Aufbau und Anschlussbelegung des BBC micro:bit. Grafik: micro:bit developer community

Bild 2 zeigt die Übersicht über den Calliope mini und dessen Anschlussbeschaltung.

Beide Plattformen sind ab Werk bereits mit ersten Applikationen programmiert, die mitgelieferten Booklets machen deren Nutzung sofort ohne Ballast spielerisch möglich.

Einfach starten

Beide Systeme sind für Kinder bzw. absolute Einsteiger konzipiert, entsprechend gut unterstützt werden sie über ihre Internet-Plattformen. Beide bieten ein sehr einfaches Tutorial für den Einstieg, das erste

Kennenlernen und das Erkunden der Plattform über die bereits geladenen Programme.

Die Systeme sind entweder per USB-Kabel oder Bluetooth mit dem Computer, einem Smartphone oder Tablet zu verbinden – dann kann man sofort die verschiedenen, zur Verfügung stehenden Entwicklungsumgebungen nutzen. Auch diese gleichen sich bzw. überschneiden sich sogar. Hauptsächlich kommen hier Code-Editoren, die auf Programmierblöcken oder grafischen Programmiersprachen basieren, zum Einsatz, aber auch „erwachsene“ Programmiersprachen wie MicroPython oder JavaScript. Es sind auch zahlreiche bekannte Editoren einsetzbar, die etwa Scratch der Scratch Foundation oder C++ benutzen, für den Einstieg sollte man sich jedoch an die offiziellen Editoren der Anbieter halten, denn diese bieten auch die besonders einfache Verbindung zum Lernsystem.

Das Herunterladen aus den meisten Editoren ist überhaupt kein Problem: mit dem Befehl „Herunterladen“ wird das erstellte und per Simulation getestete Programm automatisch kompiliert, als .hex-File gespeichert und kann

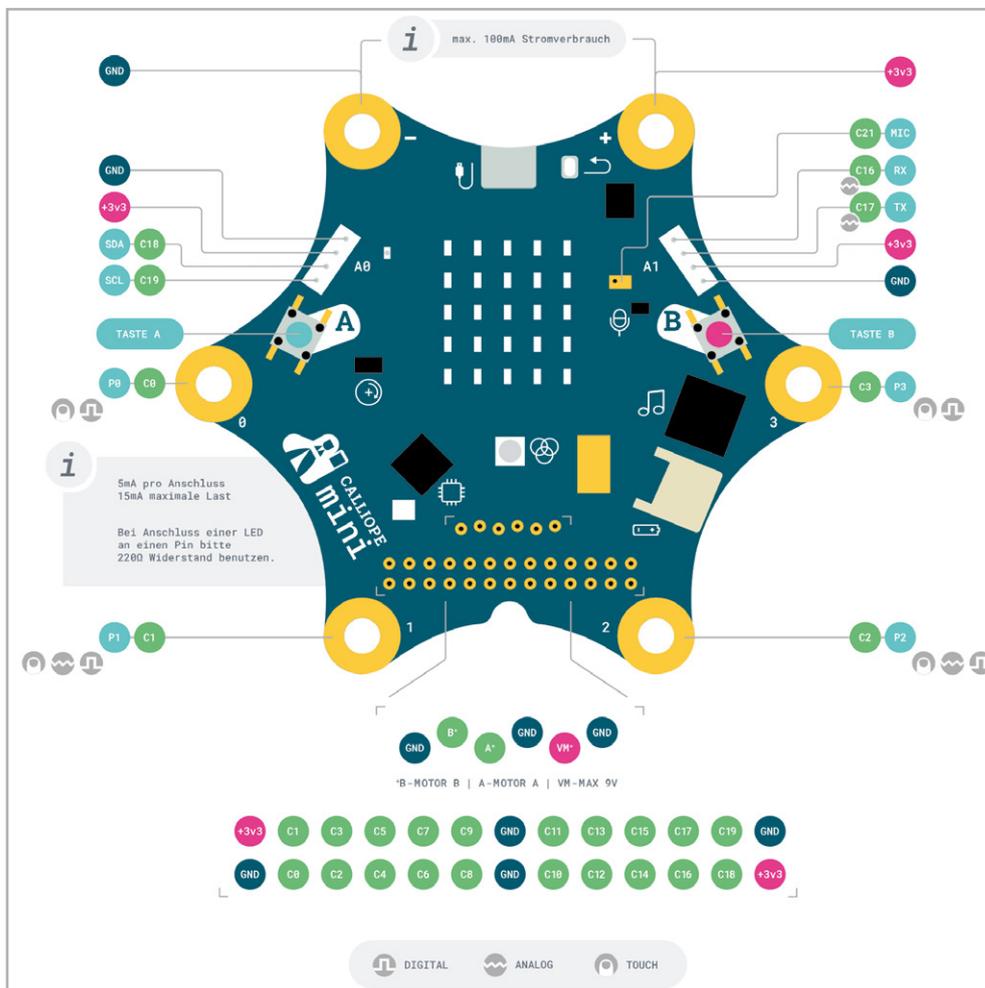


Bild 2: Aufbau und Anschlussbelegung des Calliope mini. Bild: calliope.cc



auf den angeschlossenen Mini-Computer, der auf dem jeweiligen Desktop als externes USB-Laufwerk erscheint, einfach per Drag & Drop verschoben werden.

Die Programmierung des BBC micro:bit

Für den BBC micro:bit wird primär der JavaScript-Block-Editor „MakeCode Editor“ von Microsoft (Bild 3) präferiert, dazu der micro:bit-Python-Editor (Bild 4). Unter den Empfehlungen ist auch der von Apple entwickelte Swift Playground (Bild 5), hier sind verschiedene Programmierparadigmen vereint und erlauben z. B. die schnelle Erstellung bewegter Abläufe in einer grafischen Umgebung. Der Swift Playground ermöglicht den ProgrammierEinstieg ganz ohne Vorkenntnisse, da hier stark textbeschreibend unterstützt wird. In der Edition für den BBC micro:bit ist das micro:bit book integriert, damit realisiert man mit Swift Playground sehr schnell erste Programmschritte, um z. B. die Ansteuerung von LEDs, die Auswertung von Pins und Sensoren kennenzulernen. Der Editor ist speziell für Apple-Geräte, insbesondere für neuere iPads und iPhones ausgelegt.

Der „MakeCode Editor“ dürfte aber wohl den meist genutzten Einstieg darstellen, ermöglicht er doch auf besonders einfache Weise, über den bewährten grafischen, viele Fehler von vornherein ausschließenden Blockaufbau, Programme zu erstellen. Es gibt eine große Anzahl an Beispielen, die stets mit einer Aufgabenstellung beginnen (Bild 6). Man kann genauso gut ein eigenes Projekt anlegen und sofort mithilfe der grafischen Blöcke seinen ersten Code erstellen. Im Beispiel in Bild 7 ist rechts ein kurzer, in Sekunden zusammengebauter Demo-Programmablauf zu sehen; es soll nach dem Start für fünf Sekunden ein Startsymbol erscheinen, das man sich selbst zusammenstellen kann, danach soll in einer kleinen Schleife vier Mal die Temperatur des internen Temperatursensors mit jeweils einer Sekunde Pause gezeigt werden. Das fertige Programm kann in der Ansicht des micro:bit links im Editor simuliert werden, in Bild 7 sind Sequenzen der Anzeige kombiniert. In gleicher Weise kann man auch die Eingabe auf dem micro:bit programmieren und simulieren, in Bild 8 ist die Anzeige eines Textes nach dem Drücken der Taste A auf dem Board hinzugefügt. Im Simulationsablauf kann man anhand des jeweils gelb umrandeten Programmierschritts genau verfol-

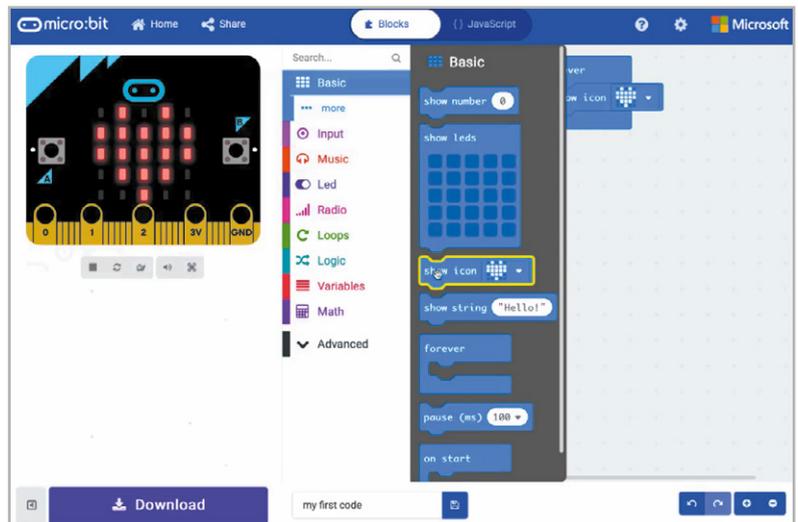


Bild 3: Blockorientierter Editor mit integriertem Simulator – der MakeCode Editor von Microsoft

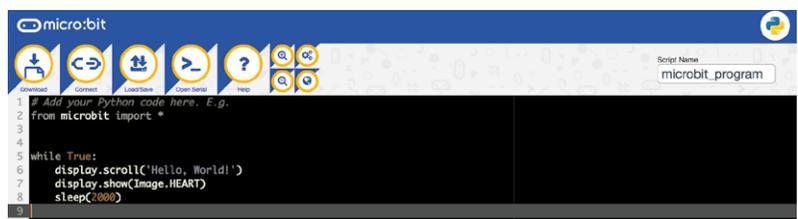


Bild 4: Der Einstieg in MicroPython gelingt mit dem speziell abgestimmten MicroPython-Editor.

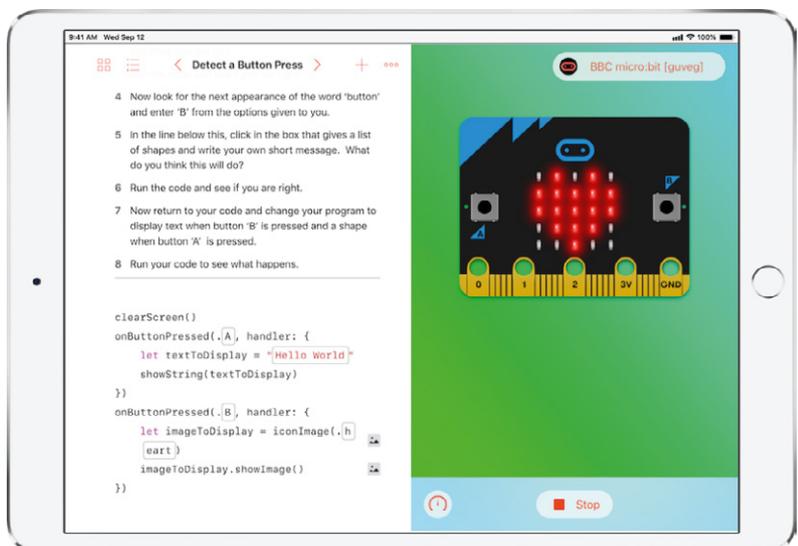


Bild 5: Der Swift Playground ermöglicht den ProgrammierEinstieg ganz ohne Vorkenntnisse.

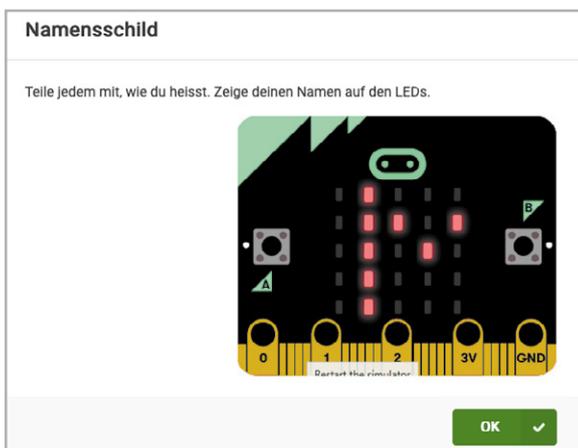


Bild 6: Die einzelnen Lektionen beginnen immer mit einer Aufgabenstellung, wie es Schüler gewohnt sind.



Bild 7: Ein einfaches Programm, hier für den Abruf der Temperatur des bordeigenen Temperatursensors, ist schnell erstellt und simuliert.

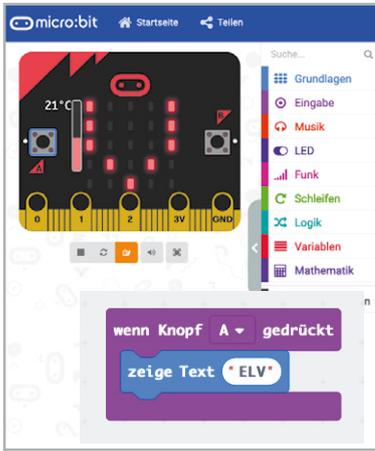


Bild 8: Hier wird auf eine Eingabe hin ein Text ausgegeben.

gen, was das Programm gerade tut. Und schließlich kann man in der JavaScript-Ansicht das Programm übersetzt sehen. Auch hier wird während der Simulation der jeweils aktive Programmschritt markiert (Bild 9).

Nach dem erfolgreichen Testlauf lädt man den fertigen, automatisch kompilierten Programmcode auf den micro:bit herunter und kann hier das Programm nutzen. Wie man übrigens sieht, läuft der Editor in der gewünschten Sprache, es gibt also nicht die Einstiegshürde der Fremdsprache, was der jungen Altersklasse der Schüler entgegenkommt.

Ganz ähnlich wie der „MakeCode Editor“ funktioniert die Programmiersprache NEPO der Programmierumgebung „Open Roberta Lab!“ [4] des Fraunhofer-Instituts für intelligente Analyse- und Informationssysteme (IAIS). Diese Open-Source-Programmierungsumgebung, die aus der Fraunhofer-Initiative „Roberta® – Lernen mit Robotern“ hervorging, ist quasi die deutsche Zentrale für derartige Lernsysteme, hier ist unter vielen anderen Systemen auch die grafische, blockbasierte Programmierungsumgebung für den micro:bit integriert. Bild 10 zeigt, wie „NEPO“ im Prinzip funktioniert. Wir haben wieder die Programmierblock-Auswahl links, stellen aus dieser unser Programm zusammen (hier werden übrigens Programmfehler sofort interaktiv angemerkt), simulieren rechts das Programm und laden es schließlich auf den Mini-Rechner. „Open Roberta Lab!“ wird ständig gepflegt und erweitert und ist wohl derzeit das mächtigste Block-Programmierungswerkzeug.

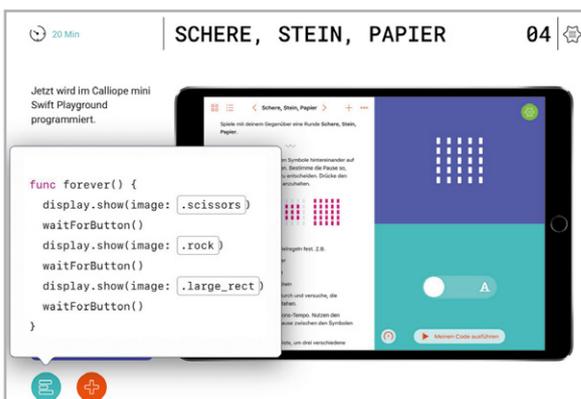


Bild 11: Auch für den Calliope Mini gibt es eine Swift-Playground-Version.

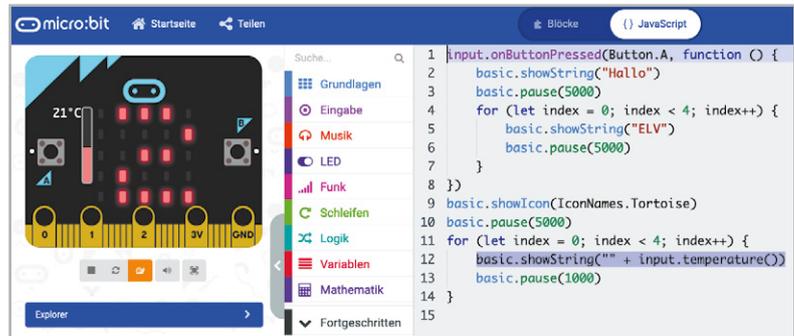


Bild 9: Alternativ zur Blockanzeige kann man Programme und Testläufe auch in JavaScript anzeigen lassen.

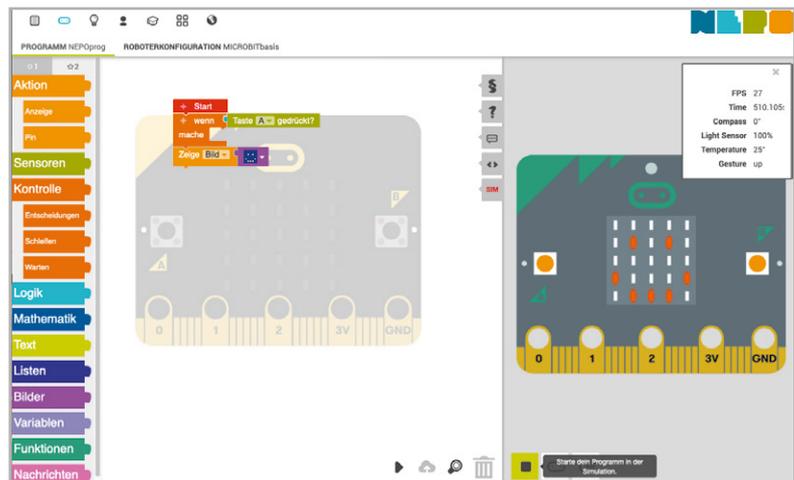


Bild 10: Der NEPO-Editor des Fraunhofer IAIS ist ebenfalls blockorientiert und enthält einen sehr großen Funktionsumfang.

Zum Schluss noch ein Wort zum bereits kurz erwähnten MicroPython-Editor. Mit der Unterstützung der zugehörigen Wiki-Seite [5] kann man sehr einfach in diese ungemein beliebte Programmiersprache einsteigen, die ja bekanntermaßen auch auf vielen Mikrorechnersystemen wie dem Raspberry Pi, Arduino oder den ESP-Systemen einsetzbar ist. Dies dürfte einer der einfachsten Einstiege in höhere Programmiersprachen sein.

Die Programmierung des Calliope mini

Auch der Calliope mini ist unter ähnlichen oder den gleichen Programmierungsumgebungen zu programmieren. Geht man von der zentralen Webseite [2] aus, trifft man auch hier den speziell für dieses System und für Mobilgeräte ausgerichteten Swift Playground (Bild 11). Ebenfalls für Mobilgeräte vorgesehen ist die Calliope-Mini-App (Bild 12), in der drei Edi-

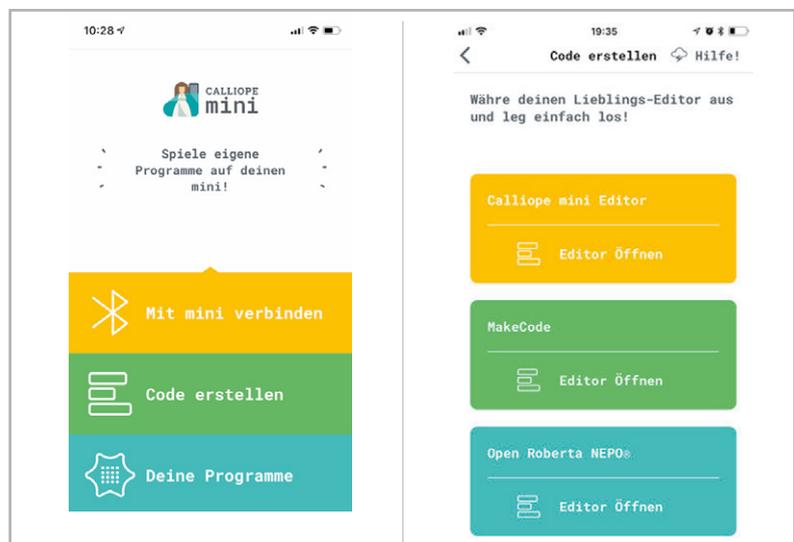


Bild 12: Für Mobilgeräte vorgesehen: die Calliope-Mini-App, in der drei Editoren verfügbar sind.

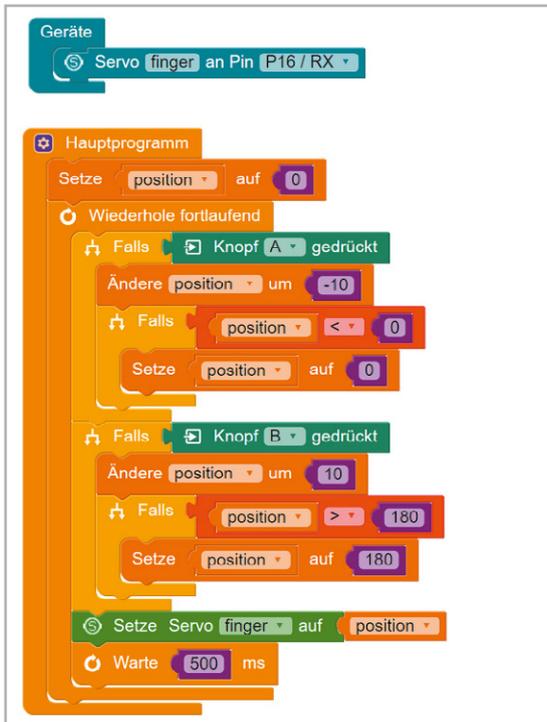


Bild 13: Eine weitere grafische und auch für komplexere Lösungen einsetzbare Programmierplattform für den Calliope mini ist „abozza!“ der Universität Osnabrück.

toren verfügbar sind. Das ist einmal der sehr einfache und für die allerersten Schritte einsetzbare „Calliope mini Editor“, der über ganz wenige Elemente erstes Verständnis für das Erstellen eines Programms weckt. Dann finden wir den bereits beim BBC micro:bit besprochenen MakeCode Editor genauso wie den ebenfalls bereits besprochenen „Open Roberta NEPO®-Editor“. Die beiden letztgenannten Editoren sind auch als Online-Plattform verfügbar und funktionieren ganz genauso wie beim Pendant für den BBC micro:bit. Eine weitere grafische Programmierplattform für den Calliope mini ist „abozza!“ der Universität Osnabrück [6], die unter jedem JavaScript-fähigen Browser und einer installierten Java-Run-time-Umgebung läuft. Auch sie basiert auf der Blockly-Grundlage, ist von der grafischen Logik her sehr didaktisch angelegt, um schnell Abläufe wie Loops u. Ä. verständlich zu machen (Bild 13). Neben den primär angesprochenen Einsteigern finden sich hier auch erfahrenere Maker schnell zurecht, die über das einfache Lernen hinaus sind und einfach nur eine schnelle grafische Programmierumgebung suchen, die auch komplexere Programme erlaubt.

Schließlich wird für den Calliope mini auch ein MicroPython-Editor angeboten: TigerJython 4 Kids (Bild 14). Der ist auch für den BBC micro:bit nutzbar, kann offline und online eingesetzt werden. Auf der zugehörigen Webseite [7] findet sich dazu ein hervorragendes deutsches Tutorial mit zahlreichen Anwendungsbeispielen für den Calliope mini.

Starterkits BBC micro:bit

Während sich der Calliope mini mit seiner bereits umfangreichen Onboard-Ausstattung, den möglichen Anbindungen der externen Grove-Bausteine und zahlreichen konkreten Tutorials vorwiegend an den Bil-

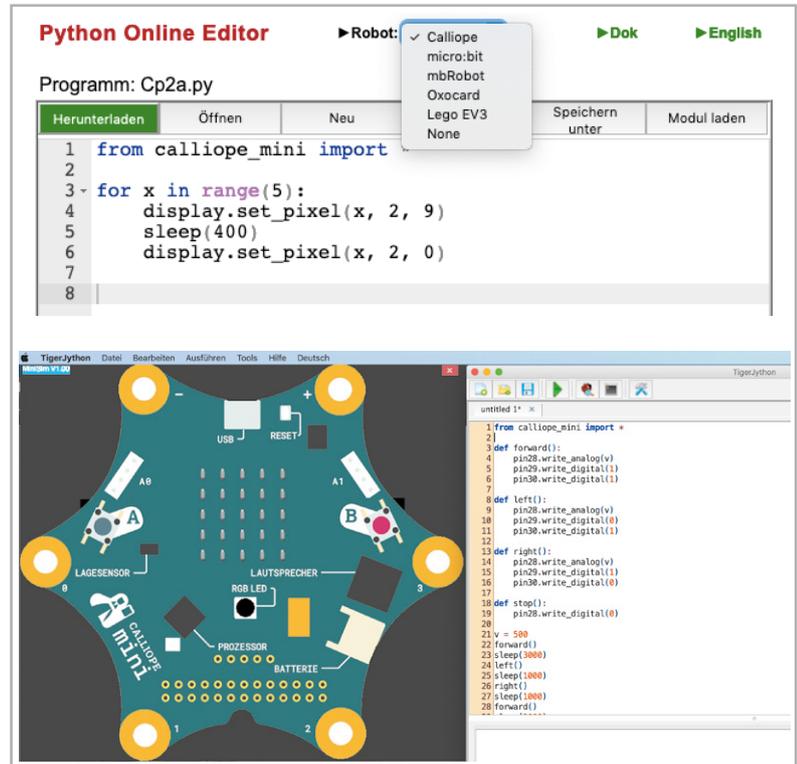


Bild 14: Den MicroPython-Editor TigerJython gibt es als Online- und Offline-Version.



Bild 15: Umfangreiches Set mit gutem Tutorial – das Velleman-micro:bit-Tinker-Kit

dungsbereich richtet, erlangt der preiswertere BBC micro:bit, nicht zuletzt dank zahlreicher Distributoren, inzwischen eine größere Reichweite auch im privaten Anwenderbereich. Um den Einstieg in die erweiterte Hardwarenutzung hier einfacher zu machen, haben z. B. Velleman und JOY-iT Starter- und Education-Sets zusammengestellt. Bild 15 zeigt das micro:bit-Tinker-Kit von Velleman, das wir gleich noch näher betrachten. In Bild 16 ist das Roboter-Kit von JOY-iT zu sehen, sehr ähnlich ist das von Velleman aufgebaut, dessen Montage der Elektronikeinheit (Bild 17) ein typisches Merkmal des Andockens für den kleinen Computer zeigt – die nur wenige Bauteile für Motoranschlüsse und Stromversorgung enthaltende Zusatzplatine wird einfach über fünf Schraubbuchsen verbaut, man muss nichts löten und hat eine für den eher rauen Roboterbetrieb mechanisch äußerst solide Verbindungsart.

Für die ersten praktischen Versuche mit externen Komponenten ist das bereits erwähnte Velleman-Tinker-Kit besonders gut geeignet. Es löst nicht nur das schnell entstehende Anschlussproblem, vor das die prop-



Bild 16: Der Roboter-Bausatz „Move Mini Buggy“ von JOY-iT basiert auf dem BBC micro:bit.

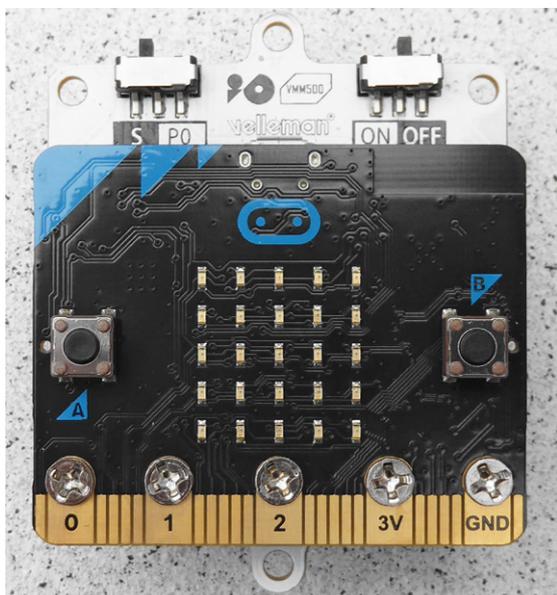


Bild 17: Mechanisch und elektrisch sichere Verbindung z. B. für Robotereinsatz über die fünf Befestigungslöcher, hier am Velleman-Roboter-Shield

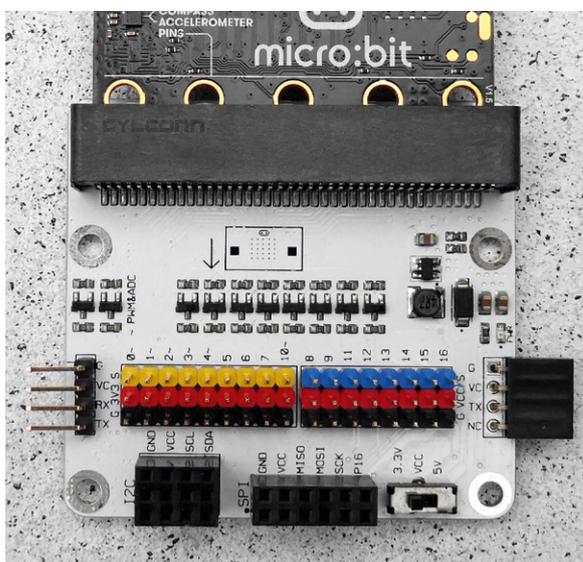


Bild 18: Das zentrale Breakout-Board (BoB) ermöglicht den einfachen Anschluss der Peripherie.

riäre Anschlussleiste des micro:bit den Anwender stellt, es bietet mit seinem zentralen Breakout-Board (BoB) auch Schutz für die Ports, die Leistungstreiber sowie diverse Anschlussleisten für die Peripherie (dreipolige Tinker-Kit-Module, I2C, SPI, UART, [siehe Bild 18](#)). Um Kurzschlüsse zu verhindern, kann das Board mit vier Stehbolzen bzw. Abstandsstücken auf einer nichtleitenden Unterlage montiert werden. Die farblich abgestimmten Anschlussleitungen und Verbindungskabel machen das Verkabeln der Module einfach.

Zum Set gehören neben den erwähnten Teilen ein Batteriekasten für zwei Mignonzellen, ein USB-Kabel, ein 128x64-Pixel-OLED-Display, ein Signalgeber, ein Tastenboard, ein Potentiometer-Modul, drei LED-Module, ein PIR-Modul, ein Bodenfeuchtesensor-Modul, ein Mikrotastermodul („Crash-Sensor“) sowie ein Satz Verbindungskabel. Ein 144-seitiges, mehrsprachiges Handbuch, beschreibt zahlreiche konkrete Projekte detailliert, unterstützt durch genaue Aufbaugrafiken ([Bild 19](#)) und viele Fotos.

Die Programmierung erfolgt über den bereits beschriebenen MakeCode Editor. Dieser bietet im Fortgeschritten-Modus den Zugang zu zahlreichen fertigen Bibliotheken von Third-Party-Herstellern, so auch zu der Tinker-Kit-Bibliothek namens „Tinkeracademy“. Nachdem man diese in den Editor geladen hat, sind hier alle im Kit vorhandenen Bauteile als fertige Funktionsblöcke versammelt, für das OLED-Display gibt es im Editor ebenfalls eine eigene Bibliothek.

Sehen wir uns einmal anhand von zwei Beispielen an, wie einfach jetzt eine kleine Applikation erstellt werden kann.

Im ersten Beispiel soll der PIR-Sensor beim Registrieren einer Bewegung eine Leuchte, hier ein LED-Modul, einschalten. Das Programm sowie den Aufbau dazu zeigt [Bild 20](#).

Das zweite Programm kann der Pflanzenpflege dienen. Der Bodenfeuchtesensor des Sets erfasst die Bodenfeuchte im Topf und entsprechend dem im Programm gewählten Schwellwert zeigt das Display an, ob gegossen werden soll oder ob es nicht nötig ist. Zusätzlich soll ein Signalton an das Gießen erinnern.

8.1 Projekt 1 – Musikinstrument

Lassen wir Musik machen.

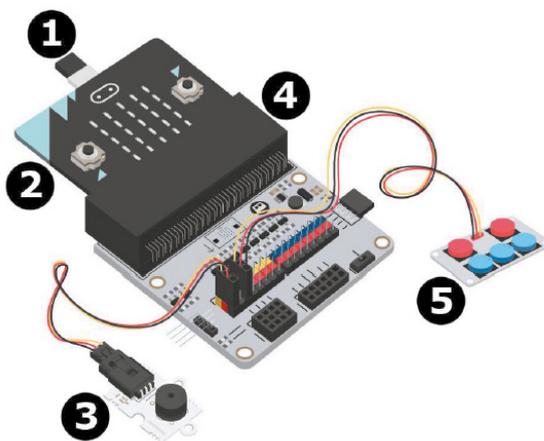


Bild 19: Detaillierte Aufbaugrafiken unterstützen die einzelnen Projekte des Tinker-Kit-Starter-Sets. Bild: Velleman

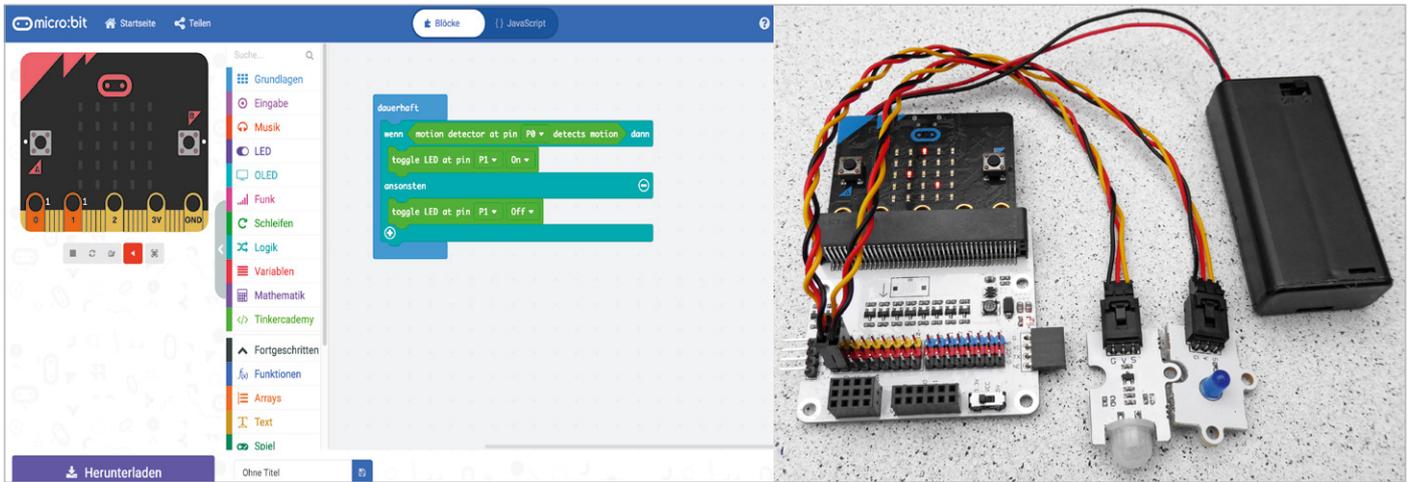


Bild 20: Ein einfaches Projekt mit zwei Tinker-Kit-Modulen realisiert einen Bewegungsmelder.

Ist die Bodenfeuchte im wortwörtlich grünen Bereich, meldet das Display, dass alles in Ordnung ist. Bild 21 zeigt dazu den Versuchsaufbau und das Programm.

Anhand nur dieser beiden kleinen Beispiele kann man sehr gut sehen, wie einfach es ist, erste Programme für Mikrocontroller zu schreiben. In der Hand experimentierfreudiger Kinder sind solche Systeme ein Selbstläufer. Mit den umgebenden Ergänzungen, Sets und Bausätzen kann man sehr schnell, ohne Löten und aufwendige Montage, eigene Lösungen erarbeiten und später auch in selbst gebauter Hardware umsetzen. Denn der Sprung z. B. zum Arduino oder zum ESPxx bzw. zum Raspberry Pi

(z. B. in Python) ist dann nur noch marginal. Die einfachen Editoren machen das Erarbeiten des Grundverständnisses, das man auch später für Hochsprachen braucht, sehr einfach – das ist dann wie das sprichwörtliche Fahrradfahren: einmal gelernt, nie wieder vergessen. **ELV**

BBC micro:bit bei ELV:

www.elv.com: Bestell-Nr. 145170, 250876, 250875

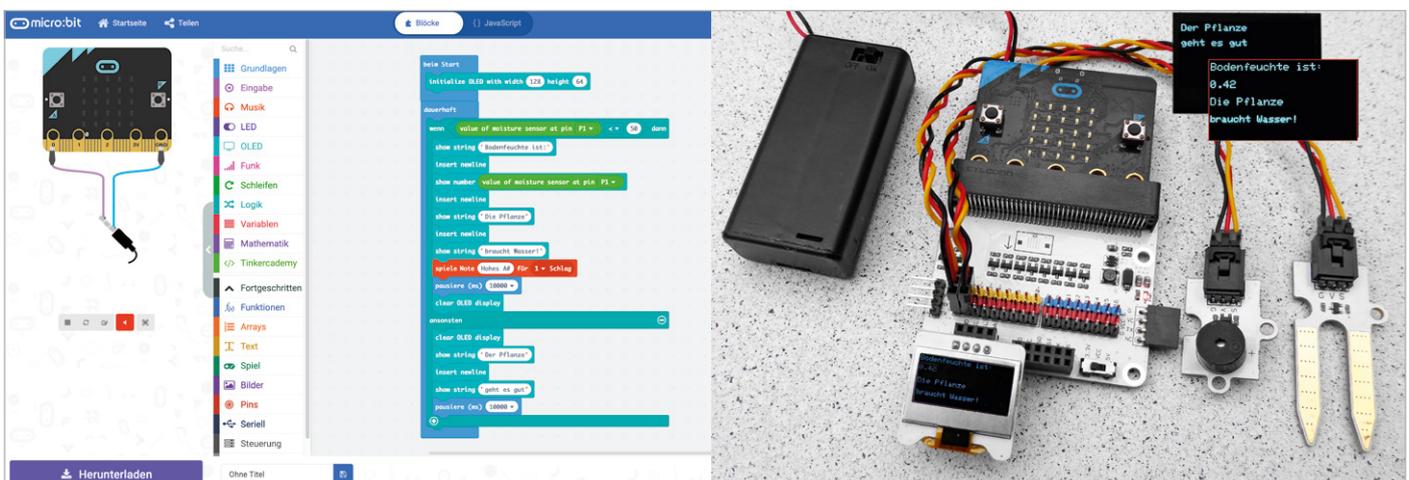


Bild 21: Dank der importierbaren Bibliotheken „Tinkeracademy“ und „OLED“ kann man schnell auch komplexere Projekte wie den Pflanzenwächter realisieren.



Weitere Infos:

- [1] Zentrale Seite BBC micro:bit und die Hardwarebeschreibung: <https://microbit.org>
<https://lancaster-university.github.io/microbit-docs/ubit/>
- [2] Zentrale Seite Calliope mini: <https://calliope.cc>
- [3] Informationen zum Grove-System: http://wiki.seedstudio.com/Grove_System/
- [4] Programmierumgebung Open Roberta Lab!: <https://lab.open-roberta.org/>
- [5] BBC micro:bit-MicroPython-Wiki: <https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/>
- [6] „abbozza!“-Homepage: <https://inf-didaktik.rz.uos.de/abbozza/calliope/firststeps.php>
- [7] TigerJython4Kids-Homepage (Calliope mini unter „Robotik“): <https://www.tigerjython4kids.ch/>

Alle Links finden Sie auch online unter: de.elv.com/elvjournals-links