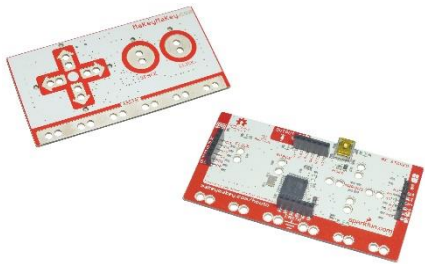
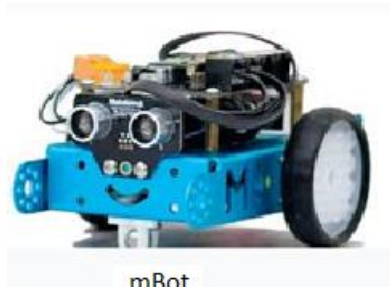


# Überlegungen

zu einer  
methodisch-didaktisch wohlgedachten, standortbezogenen, zielführenden und kindgemäßen  
**Digitalen Grundbildung**



Makey Makey



mBot



Meowbit



Ozobot



Raspberry Pi



Calliope mini



BBC Micro:bit



Codey Rocky



RCX



micro:Maqueen



Codeybot



Jewelbot



BrainPad Arcade

## Zum Geleit

Aus der Verordnung der Bundesministerin für Bildung, mit der die Verordnung über die Lehrpläne der Neuen Mittelschulen sowie die Verordnung über die Lehrpläne der allgemeinbildenden höheren Schulen geändert werden: „**Digitale Grundbildung umfasst digitale Kompetenz, Medienkompetenz sowie politische Kompetenzen.**“

„Ziel(e): Ausstattung aller Schülerinnen und Schüler in der Sekundarstufe I mit digitalen Kompetenzen, mit dem Ziel eines **informierten, souveränen und verantwortlichen** Umgangs mit Medien und Technik durch **mündige Bürgerinnen und Bürger in der Demokratie** und einer zunehmend von Digitalisierung beeinflussten Gesellschaft. Im Mittelpunkt steht dabei die **reflektierte Verwendung von Medien und Technik.**“



„Programmieren oder programmiert sein“

- Programmieren ist so etwas Ähnliches wie **Denkunterricht!** Beschäftigen wir uns mit dem Programmieren, dann beschäftigen wir uns mit dem **Denken**, wir werden zu **Denkern**.
  - Werden wir **keine Denker**, dann werden **andere** das **Denken** für uns übernehmen.
- In **Georg Büchners** Komödie „Leonce und Lena“ aus dem 19. Jahrhundert spielt ein König mit, der Peter heißt.
- Peters Leitspruch lautet: „**Der Mensch muss denken, und ich muss für meine Untertanen denken, denn sie denken nicht, sie denken nicht!**“
  - Büchners König Peter ist jedoch ein eher unfähiger, realitätsferner König!
- Was passiert also, wenn solch ein König (solch ein Politiker) das Denken für seine Untertanen (für die Staatsbürger) übernimmt, nur weil diese nicht denken können?



**Die Nicht-Denker  
werden zu  
Marionetten!**



# Teil A

## Didaktisches Konzept

### 1. Fehlentwicklung des Informatikunterrichts - Informatikunterricht ist keine Fahrschule für den Computer [1]

Unsere Verbindliche Übung "Digitale Grundbildung" soll einen Weg aufzeigen, wie **tradiierter, ECDL-gefügiger** „Informatikunterricht“ überwunden werden kann und zu **Sach-, Methoden-, Sozial-, Selbst- und Medienkompetenz** führen kann.

Eine Didaktik der Schulinformatik beschäftigt sich mit der Frage, wie der **kreative Kern des Fachs Informatik** vermittelt werden kann. Der Informatikdidaktiker Roland Mittermeier meint: "Wir brauchen mehr Konzepte, die die **kreativitätsfördernden Inhalte des Fachs Informatik** in den Vordergrund rücken - die also auf das so genannte '**computational thinking**' abzielen. Letztlich wird es wohl darum gehen, **IKT-Anwendungskompetenzen mit technischen und kreativen Fähigkeiten zu einem adäquaten Mix für den Informatikunterricht zusammenzuführen.**" [2]

„Informatik ist kein Lern- und Prüfstoff. Informatik ist ein Gebiet das durchwandert und erarbeitet werden will“, so Roland Mittermeier. [3] Mittermeier formuliert weiter, dass jene die nur Anwendung und Skills sehen, den Bildungsauftrag der Schule übersehen. „Schule soll nicht Skill-Training sein. Schule will Bildung vermitteln. Mit ungelernten Office-Hilfsarbeitern und Hilfsarbeiterinnen werden jene Büros, die zwar kurzfristig Bürohilfsarbeiterinnen benötigen, einige Zeit kostengünstig Personal einstellen können. Längerfristig werden österreichische Unternehmen aber weiterbildungsfähige Kollegen und Kolleginnen benötigen. Dazu müssen diese aber mehr mitbringen als oberflächliches Produktwissen.“ [4]

Die Europäische Union empfahl 2006 acht Schlüsselkompetenzen zum lebensbegleitenden Lernen, eine dieser acht Schlüsselkompetenzen ist die Computerkompetenz. Sie umfasst die **sichere und kritische Anwendung der Technologien der Informationsgesellschaft (TIG)** und daher die **Beherrschung der Informations- und Kommunikationstechnologien (IKT)**.

Diese so definierte Schlüsselkompetenz ließ und lässt die Schulverantwortlichen der österreichischen Pflichtschule einen nicht sehr effizienten Weg von Informatikunterricht gehen. Der derzeit noch immer in der Pflichtschule **vorherrschend propagierte Bildungswert der Informatik**, nämlich das **Benutzenkönnen von Software** (*Computer Literacy, also Word, Excel, Powerpoint etc*) ist zu hinterfragen. Wenn wir die äußerst akute Kurzlebigkeit von Hardware und Software betrachten, erscheint das sich Beschränken auf das Benutzenkönnen von Software äußerst bedenklich. Zukunftssicherheit der Kenntnisse ist also garantiert nicht garantiert. "Es ist nicht erkennbar, welche grundlegenden und prinzipiell wichtigen Inhalte und Fähigkeiten damit gelernt werden können, worin also letztlich der 'Bildungswert' dieser Systeme (der oben erwähnten Software) besteht." [5] **Dieses oberflächliche und kurzlebige Produktwissen über Computer Literacy liegt höchstens auf der Ebene des Erlernens des Maschinschreibens und vermittelt fast kein langlebige zukunftssicheres Wissen.**

Vielmehr sollen die SchülerInnen vor allem die Möglichkeit bekommen, in altersadäquater Form subjektiv relevante Problemstellungen zu definieren, zu bearbeiten und ihren Erfolg dabei zu

kontrollieren, d.h. im Sinne konstruktivistischer Pädagogik die Überprüfung der Brauchbarkeit der Lösung ihrer Konstrukte durchzuführen.

## 2. Computational Thinking - eine grundlegende Fähigkeit für das 21. Jahrhunderts

In den USA denkt man derzeit laut über „Computational Thinking“ („Informatisches Denken“) im Rahmen der Allgemeinbildung nach.

"Computational Thinking" entwickelt eine Vielzahl von Fähigkeiten (Logik, Kreativität, algorithmisches Denken, Gliederung komplexer Probleme in leichter lösbare Teilprobleme, Modellbildung), beinhaltet die Verwendung von wissenschaftlichen Methoden und hilft sowohl Erfindergeist als auch innovatives Denken zu entwickeln. Es hat seine Wurzeln in der Mathematik, in Technik, Technologie und Wissenschaft, ist letzten Endes eine Synthese von Ideen aus all diesen Bereichen.

Jeanette M. Wing hat den Begriff „Computational Thinking“ geprägt [6]. Sie geht davon aus, dass Computational Thinking eine fundamentale Fertigkeit werden wird, die von jedem Menschen in der Mitte des 21. Jahrhundert genutzt werden wird. Wer analytisch und präzise wie ein Computerwissenschaftler denken lernt, kann komplexe Probleme effizienter lösen. Von einem solchen mentalen Hilfsmittel wie dem "Computational Thinking" können Menschen in allen Lebensbereichen profitieren. Wichtig dabei sind die informatischen Konzepte, also jene Inhalte der Informatik, die zum Unterschied vom nicht zukunftssicheren Beherrschen von Informatikprodukten langfristige Gültigkeit und Bedeutung haben.

Solche informatischen Konzepte machen einen Informatikunterricht sehr attraktiv, da sie für das Leben in unserer Gesellschaft relevant sind, sie mit der Zeit nicht an Bedeutung verlieren, sie für alle Altersstufen verständlich gemacht werden können und auch in anderen Gegenständen angewendet werden können. [7] Solche informatischen Konzepte sind nach Andreas Schwill [8] Algorithmierung (Entwurfspadigmen, Programmierkonzepte, Ablauf, Evaluation), strukturierte Zerlegung (Modularisierung, Hierarchisierung) und Sprache (Syntax und Semantik).

Diese informatischen Konzepte müssen natürlich für unsere Schüler altersgerecht und kindgemäß aufbereitet werden. Spezielle Software hat diese Aufgabe der altersgerechten und kindgemäßen Aufbereitung bereits übernommen und ausgeführt. (z. B. Micro Worlds, Turtle Art, Scratch, ByoB)

**Kompetenzen entwickelnder Unterricht ist derzeit das Um und Auf der österreichischen Pflichtschule. Ein Kompetenzen entwickelnder Informatikunterricht, der Schlüsselkompetenzen wie beispielsweise Denken in komplexen Zusammenhängen, Problemlösen, Abstrahieren, Analysieren, Zielstrebigkeit aber auch Teamfähigkeit aufbauen soll, wird besonders den informatischen Disziplinen "Programmieren" und "Robotik" aber auch "Physical Computing" breiten Raum geben müssen und das Benutzenkönnen nicht zukunftssicherer Anwendersoftware recht deutlich in den Hintergrund rücken müssen.**

### 3. Programmieren, Robotik, Physical Computing

Programmieren ("Coding") als Basisteil der Informatik wird in mehreren Ländern immer mehr ein obligatorischer Teil des Informatikunterrichts. Der Tradition der osteuropäischen Länder und Russlands schlossen sich mehrere asiatische Länder an, und in den USA überlegt man laut über „Computational Thinking“ im Rahmen der Allgemeinbildung nach. Diese Art von Denken ist für die Forschung und Entwicklung in den naturwissenschaftlichen und technischen Disziplinen unvermeidbar.

Programmieren in der Schule fördert die Bildung der Kinder auf vielen Ebenen. Die Schule soll zur Entwicklung zu eigenständigen Persönlichkeiten beitragen und zu lebenslangem Lernen befähigen. Dabei steht nicht umfangreiches Faktenwissen im Mittelpunkt, sondern der Aufbau von Schlüsselkompetenzen wie beispielsweise Problemlösen, Abstrahieren, Analysieren oder im Team arbeiten.

Durch das Programmieren erhalten Kinder nicht nur Einblicke in die Geheimnisse moderner Technologien, sie können so auch diese Schlüsselkompetenzen aufbauen. Der Programmierunterricht fördert diese Kompetenzen auf spielerische Art.

Natürlich wollen wir keine Programmierer ausbilden. Aber: Wir lernen z.B. ja auch Musik in der Schule, aber fast keiner wird professioneller Musiker. Programmieren bedeutet vor allem Schulung für das logische Denken. Auch die Kreativität wird gefordert und gefördert.

Es ist derzeit (leider noch nicht in Österreich) eine "Renaissance der Schildkröte" zu beobachten, also jener schon in den 60er Jahren des 20. Jahrhunderts entwickelten Computer-Schildkröte, die zeichnen kann, wenn man ihr nur die richtigen Befehle gibt. Programmiersprachen wie FMS-LOGO, MicroWorlds, Python, Scratch, Turtle Art, SmallBasic, RoamerWorld, Tynker sind heute ganz hoch im Kurs und bieten solch eine **Schildkröte** oder ein **ähnliches Reptil** an, mit der man das Programmieren erlernen kann.

Manche dieser Programmiersprachen gehen weit über das Zeichnen mit der Schildkröte ("Turtlegrafik") hinaus und wurden konsequent um vor allem multimediale Komponenten wie Import oder auch Erstellung von Sound, Grafik, Video und Animationen erweitert. Der Erstellung von Multimediaprojekten, -spielen und -simulationen steht somit nichts mehr im Wege. Auch die Kindgemäßheit bei der Handhabung dieser Programmiersprachen wurde erheblich weiterbetrieben. Manche haben auch das Internet erobert, man kann mit ihnen HTML-Seiten erstellen, die mit diesen modernsten Versionen kreierte interaktiven Projekte können mit Hilfe von Plugins lauffähig im Internet veröffentlicht werden.

Der programmierende Schüler wird somit vom passiven Web-Benutzer zum aktiven Web-Designer.

Manche dieser Programmiersprachen haben sich auch verstärkt den Kapiteln Robotik und Physical Computing zugewandt, Kapitel, die in der österreichischen Pflichtschule noch wenig bis keinen Stellenwert haben.

Denn die Verwendung eines **Microcomputers** und die Möglichkeit ihn zu **programmieren** eröffnet neue Möglichkeiten im Informatikunterricht.

Microcomputer sind selbständig in der Lage, Daten zu koordinieren und auszuwerten. Wir brauchen dazu neben dem Microcomputer einen Computer und eine geeignete Programmiersprache. Die mit dieser Programmiersprache entwickelten Programme werden dann über eine Infrarotschnittstelle oder über Bluetooth oder WLAN in Form von Befehlen an den Microcomputer weitergeleitet.

Man kann mit Hilfe dieser geeigneten Programmiersprache den Microcomputer in der Art programmieren, dass dieser unter Verwendung unterschiedlicher Sensoren (wie etwa Licht-, Wärme, Bewegungs- und Berührungssensoren) seine Umgebung passiv beobachtet und physisch wahrnehmbare Aktionen registriert und mittels sogenannter Aktoren (wie Motoren, Lampen oder Tönen) Reaktionen und Aktionen auslöst und somit seine Umwelt verändert. So ist es möglich, einen beweglichen, selbstgesteuerten Roboter zu entwerfen.

Eingebettet wird die Nutzung der Microcomputer in eine handlungs- und problemorientierte Lernumgebung, in der das "Lernen mit allen Sinnen" praktiziert und eine positive emotionale Bindung an den Informatikunterricht gefördert wird. In solch einem Informatikunterricht besteht die Verzahnung von Theorie und Praxis auf besondere Weise: Einen Roboter kann man in die Hand nehmen und seine Fähigkeiten bereits erahnen, wenn man ihn "morphologisch" untersucht: Ein Roboter mit "Füßen" oder Rädern kann sich potenziell fortbewegen. Ein Roboter mit Tastsensoren kann registrieren, wenn er an ein Hindernis stößt. Es existiert somit eine problemhaltige Ausgangssituation und damit ein guter Einstieg für genetisches, entdeckendes, forschendes und selbständiges Lernen.

Roboter bieten eine gute Grundlage, um Schülerinnen und Schüler in einem handlungsorientierten Unterricht an das technikbasierte Thema Informatik heran zu führen. Dies gilt für alle Altersgruppen und jede Vorbildung der Schülerinnen und Schüler. Beim Entwerfen, Konstruieren, Programmieren und Testen von mobilen autonomen Robotern erfahren sie, dass Technik Spaß macht, lernen wie technische Systeme entwickelt werden und erwerben Fachkenntnisse. Der Bau und die Verwendung von Robotern vereint in idealer Weise viele Elemente technischen Wissens, die für ein Verständnis technologischer Probleme bis hin zu philosophischen Fragen – wie etwa der der Intelligenz und Autonomie von Gegenständen, die ihre Form und Funktion durch menschliche Einwirkung und Interpretation erhielten – hilfreich sind.

Roboter bieten einen auf Technik bezogenen Zugang zu Informationstechnik. Neben vielen fachlichen Kenntnissen und Methodenwissen werden vor allem nicht-fachliche Kompetenzen, sogenannte Soft Skills wie zum Beispiel Teamfähigkeit, gefördert und Interaktion, Kommunikation, Präsentation und Dokumentation geübt. Die Schülerinnen und Schüler erledigen die Arbeitsaufträge in Gruppenarbeit mit 2-3 Lernenden pro Gruppe. Idealerweise verfügt dabei jede Gruppe über einen eigenen Roboter-Bausatz.

Der **Raspberry Pi** wiederum macht es uns sehr einfach, in die Welt des „Physical Computing“ einzusteigen, indem man eigene Projektideen mit Hilfe von computergesteuerten Sensoren, Tasten oder Leuchtdioden realisiert.

Er besitzt nämlich eine so genannten „**General Purpose Input / Output**“- Schnittstelle (GPIO), mit deren Hilfe sich weitere elektronische Bauteile an den Raspberry Pi anschließen und steuern lassen.

Add-on-Boards für den Raspberry Pi wie die **LedBorg-Platine** oder auch **Pibrella**, das **Makey Makey** oder das **PicoBoard** ermöglichen zusammen mit der Programmiersprache Scratch recht interessante ja manchmal sogar recht verrückte Projekte.

Neuere Boards wie der **CodeBug**, der **BBC Micro:bit**, der **Calliope mini**, das **Sense Hat** oder das **Pixel Kit** müssen erst auf ihre Brauchbarkeit für den Informatikunterricht in der Sekundarstufe 1 getestet werden.

#### **4. Roboter mit Smartphones oder Tablets steuern**

*Roboter der neuesten Generation können neben dem Computer auch bereits mit Tablet oder Smartphone programmiert und gesteuert werden. Die etwas kostspieligen Roboter **mBot**, **Ozobot**, **Sphero**, **Ollie**, **BB-8**, **Dash and Dot**, **Cubelets** und die österreichische Entwicklung **Robo Wunderkind** können mit speziellen Apps über Smartphone oder Tablet programmiert und gesteuert werden.*

#### **5. Making - kreatives digitales Gestalten und Experimentieren [9]**

Making, das kreative Gestalten und Selbermachen insbesondere mit digitalen Technologien, soll in den Schulalltag auch der Pflichtschule Einzug halten. Die Schüler sollen zu „Makern“ werden. Technisches Verständnis, kreatives Problemlösen, soziales Miteinander und handwerkliches Geschick werden dabei trainiert.

Makers sind Menschen, die basteln, etwas Neues gestalten. Sie verwenden dazu einerseits digitale Technologien wie z.B. eine Programmiersprache, andererseits Materialien, mit denen sie experimentieren und plötzlich etwas ganz Interessantes aber auch manchmal etwas ganz Verrücktes herstellen. Makers wissen, dass z.B. ein Bleistift nicht nur zum Schreiben zu gebrauchen ist, oder dass man leere Küchenrollen für eine interessante Musikanstallation verwenden kann.

Weil es darum geht, dass etwas „gemacht“ (engl. „to make“) wird, also etwas Neues entwickelt und produziert wird, wird diese Entwicklung des „digitalen Do-It-Yourself“ auch als Maker-Bewegung bezeichnet.

Wenn man davon ausgeht, das Making in den Schulen Einzug halten soll und muss, dann fällt das zu großen Teilen in den Verantwortungsbereich des Informatikunterrichtes. Auch das Technische Werken wird hier einen großen Beitrag leisten müssen. Die Abschaffung des Technischen Werkens in unseren 8. Schulstufen ist somit zu überdenken.

Making ist nicht nur für den Informatikunterricht sondern für alle anderen MINT-Fächer interessant. Aber es werden in den Making-Projekten nicht nur MINT-Interessen geweckt und MINT-Kompetenzen entwickelt, sondern auch Soft Skills wie Teamfähigkeit, analytische Kompetenz, Problemlösekompetenz, Kreativität, eigenständiges Arbeiten, Flexibilität und vor allem Neugierde bei der Suche nach Lösungen.

Making mit Schülern stellt das aktive Handeln und die damit verbundenen Wissenszuwächse in den Mittelpunkt und fördert diese im späteren Berufsleben unserer Schüler von ihnen erwarteten Soft Skills.

Making baut auf den Ideen des Konstruktivismus des Mathematikers und Vater der Computerpädagogik Seymour Papert auf. Der Erfinder der Programmiersprache LOGO geht davon aus, dass Menschen besonders dann etwas lernen, wenn sie etwas selbst konstruieren und sich im Konstruktionsprozess das dafür nötige Wissen aneignen und anwenden. Durch das eigene aktive Handeln werden schwierige Inhalte leichter durchschaubar und begreifbar.

## Literatur

[1] Alpen-Adria-Universität Klagenfurt: Uni News

[Http://www.uni-klu.ac.at/main/inhalt/41394\\_41280.htm](http://www.uni-klu.ac.at/main/inhalt/41394_41280.htm)

[2] ebenda

[3] Roland Mittermeir in Reiter, Scheidl, Strohmayer, Tittler, Weissenböck (Hrsg.): Schulinformatik in Österreich. Erfahrungen und Beispiele aus dem Unterricht. Ueberreuter Wien. 2003. S. 10

[4] ebenda, S. 11

[5] H. LÖTJE, in: Mindstorms: Kinder, Computer und Neues Lernen  
Birkhäuser, Basel 1982

[6] Jeanette M. Wing: Computational Thinking. In: CACM Viewpoint, March 2006

[7] Gerald Futschek: Österreich braucht gut ausgebildete InformatiklehrerInnen  
[http://publik.tuwien.ac.at/files/PubDat\\_200228.pdf](http://publik.tuwien.ac.at/files/PubDat_200228.pdf)

[8] Andreas Schwill: Fundamentale Ideen der Informatik. In: Zentralblatt für Didaktik der Mathematik 25 (1993) Nr. 1, S. 20-31.

[9] Sandra Schön, Martin Ebner und Kristin Narr (Hrsg.): Making-Aktivitäten mit Kindern und Jugendlichen. Handbuch zum kreativen digitalen Gestalten, 2016

## Teil B

### Mögliche Highlights für die Umsetzung dieses Konzepts

**"Learn to code, it might save your job!" / "If you learn to code, you can code to learn"**

Ohne vernünftige Programmierer würden Unternehmen – und der Rest der Welt – vermutlich ganz schön alt aussehen. Umso wichtiger ist es, dass die Schüler bereits sehr früh an das Thema herangeführt werden. Programmieren von kindgemäßen Robotern macht es möglich!

Oft wird der typische Programmierer gerne als Stubenhocker und Außenseiter, als „Nerd“, dargestellt. Doch in unserem digitalen Zeitalter könnte fast behauptet werden, dass Programmierer die Welt regieren – ohne sie funktioniert nichts. Die IT-Abteilung spielt in modernen Unternehmen eine zentrale Rolle und Programmierer kriegen gute Jobs. Um den steigenden Bedarf an Programmierern zu decken, ist es natürlich von Vorteil, bereits die Jüngsten spielerisch an das Thema heran zu führen – und hier kommen kindgemäße Roboter und Programmierumgebungen ins Spiel!



**Online-Editoren**[Scratch 3.0](#)[mBlock 5](#)[Microsoft MakeCode for Microbit](#)[Microsoft MakeCode for Calliope mini](#)[Microsoft Makecode für Arcade und Meowbit](#)[Lightbot](#)[Codebug](#)[Code-Studio](#)[Tynker](#)[Blockly](#)**Roboter**[Codey Rocky](#)[Codeybot](#)[Ozobot](#)[mBot](#)[Micro:Maqueen](#)[Callibot](#)**Micro:bit und Co.**[BBC Micro :bit](#)[Calliope](#)[Codebug](#)[Meowbit](#)[BrainPad Arcade](#)**Physical Computing**[Experimentierbox für Calliope](#)[Starterkit für Micro :bit](#)[Fischertechnik „Micro :bit“](#)[Fischertechnik „Calliope“](#)**Codey Rocky und das Internet der Dinge**

Im Internet der Dinge (Internet of Things, IoT) werden Objekte der realen Welt (Dinge also) miteinander und mit dem Internet verknüpft. Sie werden damit selbst zu Teilnehmern des Internets. Das Internet ermöglicht es den Objekten dann, Daten untereinander auszutauschen und gemeinsam zu nutzen. Sie kommunizieren also selbständig über das Internet und erledigen verschiedene Aufgaben für ihre Besitzer. So können sie z.B. der Informationsversorgung für ihre Besitzer dienen. In unserem Beispiel ist Codey Rocky so programmiert worden, dass er – er hat ja WLAN integriert – bei vorhandener WLAN-Verbindung Wetterdaten für Wien aus dem Internet abrufen und sie seinem Besitzer zur Verfügung stellt. Natürlich geht das z.B. mit einem Smartphone auch. Das Beispiel soll aber zeigen, was Codey Rocky und die Programmierumgebung mBlock 5 dem programmierenden Schüler alles bieten, und es soll auf die computerpädagogische Bedeutung von Codey Rocky und mBlock 5 hinweisen.

[Demo](#)**Der Codey Rocky**

Der Codey Rocky ist ein kindgemäßer Lernroboter, mit dem Schülerinnen und Schüler erste Erfahrungen mit dem Programmieren machen können. Er besteht aus einem Controller (Codey), das "Gehirn" des Roboters, das entweder unabhängig oder in Kombination mit einem mobilen Fahrgestell (Rocky) einsetzbar ist. Der Roboter verfügt zur Kommunikation mit anderen Geräten über WiFi, Bluetooth und USB. Obwohl für Kinder ab 6 Jahren konzipiert, obwohl ein wenig sehr spielzeugmäßig ausschauend (er ist sehr robust) bietet dieser Lernroboter durch seine vielen Module (IR-Sender und Empfänger, Gyroskop, Lichtsensoren, Farbsensoren, Potentiometer etc.) viele Möglichkeiten des Einsatzes. Anfänger können ihre ersten Programme mit der Programmiersprache mBlock 5 (die auf Scratch 3.0 basiert) erstellen, während Fortgeschrittene

ihre Kenntnisse mit Python erweitern können. Wir beschäftigen uns vorerst mit den Grundlagen des Codey Rockys und der Software mBlock 5 (Erstellen von Animationen, Experimentieren mit Tönen, Einsatz des Codey als Game-Controller, Erkennen von Farben, Steuerung von Scratch-Programmen). Später werden wir uns mit dem Fahrkönnen des Codey Rockys auseinander setzen und mit verschiedenen Apps, die das Programmieren und Steuern des Codey Rockys auch mit Smartphone und Tablet ermöglichen.

[Beispiel 1](#)   [Beispiel 2](#)   [Beispiel 3](#)   [Beispiel 4](#)   [Beispiel 5](#)   [Beispiel 6](#)   [Beispiel 7](#)   [Beispiel 8](#)  
[Farbsensor](#)   [Game Controller 1](#)   [Game Controller 2](#)   [Labyrinth](#)

### Der Meowbit

Der „Meowbit“ ist eine ganz neue Spielkonsole, mit der man einfache Videospiele ganz leicht selbst programmieren kann. Da er mit einer Microbit-40 PIN-Schnittstelle ausgestattet ist, ist er mit fast allen auf dem Markt erhältlichen Microbit-Erweiterungskarten kompatibel.

Die Programmierung erfolgt für uns mit „Makecode für Arcade“ von Microsoft. Die Programmierumgebung ist einfach zu bedienen, da wir sie von der Programmierung unseres Micro :bit („Makecode für Micro:bit“ von Microsoft) bereits kennen. Das Aufspielen der fertigen Skripte erfolgt genauso wie beim Micro :bit.

Der „Meowbit“ hat einen Ein- und Ausschalter, eine Reset-Taste (um die Skripte aufspielen zu können) und eine DFU-Modus-Taste (Device Firmware Upgrade) für Upgrades aber auch um ein Menü zur Hardwaresteuerung aufzurufen .

Ein 160 mal 128 tft Farbdisplay (1,8''), 4 programmierbare Richtungstasten, 2 programmierbare A- und B-Tasten und ein programmierbarer Summer ermöglichen ein akzeptables Videospiele.

Weiters bietet der Meowbit einen Lichtsensor, zwei programmierbare LED-Lichter, einen Temperatursensor, ein 6-Achsen-Gyroskop und einen Beschleunigungssensor.

Eine Ladeanzeige zeigt den Fortschritt beim Laden einer 3,7 V Lithium-Batterie, eine Arbeitsanzeige den Fortschritt beim Aufspielen eines Skriptes.

Ein SD-Kartensteckplatz, die Microbit-40 Pin Schnittstelle und ein USB-Anschluss vervollständigen die Hardware-Ressourcen.

Der Master-Chip ist ein STM32F401RET6, 32-Bit-ARM-Cortex-M4-Kern-Prozessor, der gleiche den auch eine andere „Spielkonsole“, die erst in wenigen Wochen auf den Markt kommen wird, hat, nämlich das „[BrainPad Arcade](#)“ Deshalb ist unser Meowbit auch mit der Programmierumgebung „Makecode für Arcade“ programmierbar.

Geschützt wird der Meowbit mit einer Silikon-Hülle, die es in orange oder hellblau gibt! Der Meowbit kostet derzeit ca. € 35,00.-, er ist jedoch nur bei chinesischen Onlineshops erhältlich! Wir sollten ihn auf seine Brauchbarkeit im Unterricht testen!

Mozilla Firefox oder Google Chrome verwenden!

[Makecode für Arcade](#)  
[Meowbit](#)

### Der Micro:Maqueen

Der Micro:Maqueen ist ein Roboter der mit unserem schon bekannten BBC Micro :bit sehr einfach zum Leben erweckt werden kann. Der kleine Roboter muss erst zusammengebaut werden, das gelingt mit wenigen Handgriffen. Mit der Programmierumgebung „Makecode“ (bald auch mit „Scratch 3.0“) wird dann der Micro :bit programmiert, man muss nur eine Erweiterung für den Micro:Maqueen dazu laden. Nach dem Laden des fertigen Skripts auf unseren Micro :bit wird der in den dazu vorgesehenen Schacht des Micro:Maqueen gegeben, und schon geht es los.

Micro:Maqueen benötigt für die Stromversorgung drei AAA-Batterien. Er besitzt unter anderem zwei Infrarot-Graustufensensoren, einen Summer, einen Infrarotempfänger, zwei LED-Leuchten, vier RGB-Lichter, eine Ultraschallschnittstelle, zwei Motoren und natürlich auch zwei Räder. Obwohl sehr klein, wirkt der Roboter sehr robust.

Neben dem einfachen Fahren kann der Micro:Maqueen in der Art recht einfach programmiert werden, dass er einer schwarzen Linie folgt, oder Hindernisse erkennt und ihnen ausweicht, dass er auf Lichtstärke reagiert, oder mit einer IR-Fernbedienung oder mit einem zweiten BBC Micro :bit gesteuert werden kann!

Da er sehr klein ist, eignet er sich sehr gut für im Werkunterricht hergestellte Labyrinth, die er dann bewältigen muss. Eine nette und auch interessante Maker-Idee!

Der Roboter ist mit einem derzeitigen Preis von unter € 20,00.- sehr preisgünstig. Der Onlineshop „digikeye.at“ bietet ihn derzeit günstiger an als Shops aus der chinesischen Heimat des Roboters.  
[Micro:Maqueen](#)

### Das neue SCRATCH 3.0 ist da

Seit 2007 gibt es **SCRATCH**. Die Programmierumgebung richtet sich vor allem an Kinder und Jugendliche, um sie mit den Grundlagen der Programmierung vertraut zu machen. Seit Anfang Jänner nun gibt es die neueste Version von SCRATCH nämlich **SCRATCH 3.0**. Es gibt sie in einer **Online-Version** für die Browser Chrome, Edge, Firefox und Safari und in einer **Offline-Version**. Die Offline-Version nennt sich „**Scratch Desktop**“

**Tablet-Nutzer** können jetzt ebenfalls mit Scratch programmieren und mit dem Touchscreen arbeiten, vorausgesetzt sie nutzen **Mobile Chrome** oder **Safari** als Browser.

Das Besondere an SCRATCH 3.0 sind die vielen **Erweiterungen**, so können der **BBC Micro :bit** oder das **Makey Makey** nun mit Scratch programmiert werden. Erweiterungen für die LEGO Roboter **MINDSTORMS EV3** und **LEGO WeDO 2.0** werden ebenso zur Verfügung gestellt wie eine **Google-Translate-Erweiterung**, um Figuren in einer anderen Sprache sprechen zu lassen. Eine **Ext-to-Speech-Erweiterung** kann Text von Figuren in gesprochener Sprache darstellen. Mit einer **Video-Sensing-Erweiterung** gibt es die Möglichkeit, mit den programmierten Figuren zu kommunizieren. Sie können auf die mit einer Webcam aufgezeichneten Bewegungen der Nutzer reagieren. Eine spezielle **Musik-Erweiterung** und eine **Malstift-Erweiterung** runden das tolle Module-Angebot ab.

[Offline-Version](#)  
[Online-Version](#)

### Sortieralgorithmen - Sortieren wie ein Computer

Kennenlernen und Erproben von Sortieralgorithmen - Diesmal: Bubblesort und Quicksort  
 Modul 8 der vom Bundesministerium für Bildung, Wissenschaft und Forschung vorgesehenen Inhalte der Digitalen Grundbildung nennt sich "Computational Thinking" (Informatisches Denken) und beinhaltet unter anderem das "Arbeiten mit Algorithmen". Kennenlernen und Erproben von Sortieralgorithmen zeigen uns, wie Computer "denken"! Gleichzeitig gehen wir dabei selbst auf Entdeckungsreise unserer eigenen Denkweise und können versuchen, so unser Denken zu vervollkommen und zu wirklichen Denkern werden. Das vorliegende Handout macht sowohl Schüler als auch Lehrer mit zwei eher bekannten Sortierverfahren bekannt nämlich den Algorithmen "Bubblesort" und "Quicksort". Nach einer Erklär- und Kennenlernphase sollen Beispiele selbst geübt und "gelöst" werden. Während "Bubblesort" relativ leicht durchschaubar ist, ist "Quicksort" recht anspruchsvoll. Deshalb werden zwei Varianten von diesem Sortierverfahren vorgestellt. Eine leichtere und eine schwerere. Die Übungen werden durch zwei Video-Tutorials unterstützt! Eingesetzt wird dieser Inhalt in der NMS von uns erst in der 8. Schulstufe!

[Sortieren wie ein Computer](#)

### Jewelbot - ein Freundschaftsarmband, mit dem (nicht nur) Mädchen programmieren lernen können

Jewelbot ist ein Freundschaftsarmband, das man programmieren kann, um geheime Botschaften an seine Freunde, die auch so ein Armband besitzen, zu schicken und um andere coole Dinge damit zu machen. Jewelbot ist speziell auf Mädchen ausgerichtet, aber das Design wird auch manche Buben ansprechen. Jewelbot wurde von zwei Frauen entwickelt, die sich von Kindheit an mit Informatik und Programmierung beschäftigten. Sie möchten mit ihrer Entwicklung mehr Mädchen für die MINT-Fächer begeistern. Sie möchten bei Mädchen "eine anhaltende Liebe zu Computern und zur Programmierung" erzeugen, eine "Liebe, die die Mädchen durch ihre Leben und Karrieren begleiten wird". Unter dem Charme des Jewelbots befindet sich ein kleiner Computer. Genauso wie wir ein Programm schreiben können, das auf unserem Computer läuft, können wir den Jewelbot so programmieren, dass er leuchtet und summt. Wir programmieren den Jewelbot mit einem Programm auf unserem Computer namens Arduino IDE. Arduino ist der Name des winzigen Computers in unserem Jewelbot. Der Jewelbot hat viele Modi. Die zwei wichtigsten, die wir verwenden werden, sind der Codiermodus und der Freundschaftsmodus. Im Freundschaftsmodus können wir auf den Pairing-Modus und den Messaging-Modus zugreifen, mit denen wir uns mit den

Jewelbots anderer Freunde verbinden und Nachrichten senden können. Im Codiermodus können wir Programme auf unseren Jewelbot hochladen. Sobald wir ein bestimmtes Programm auf unseren Jewelbot hochgeladen haben, ist dies alles, was der Jewelbot ausführen kann, bis wir es durch ein anderes Programm ersetzen, das wir geschrieben und hochgeladen haben. In den Vereinigten Staaten ist derzeit der Schlachtruf "Girls who code" überall zu hören. "Jewelbot" oder die "Boolean-Box" (ein Programmierset mit Raspberry Pi, Tastatur, Maus und vielen anderen Dingen zum Einstieg ins Programmieren) aber auch die Plattform "Bubble Sort Zines" stehen dort derzeit hoch im Kurs. Es gilt nun, auch bei uns eine neue Tür auf zu stoßen und mehr Interesse und Freude am Programmieren bei Mädchen zu wecken

[Jewelbots  
Video  
Jewelbots-Handout](#)

### **Apps für Spielereien mit dem BBC Micro :Bit**

Der BBC Micro :Bit bietet uns viele Möglichkeiten des kreativen digitalen Gestaltens, einer wesentlichen Forderung der Digitalen Grundbildung  
Der BBC Micro :bit hat Österreichs Verantwortliche für die Digitale Grundbildung erreicht ! Kreatives digitales Gestalten wird im neuen Lehrplan der Digitalen Grundbildung gefordert! Das vorliegende Beispiel zeigt einen recht interessanten aber auch lustvollen Einsatzbereich des BBC Micro :Bit, nämlich die Verwendung des BBC Micro :Bit zusammen mit einem Smartphone. Apps werden vorgestellt, die über Bluetooth mit dem Micro :Bit kommunizieren und allerlei Spielereien zulassen. Die dafür nötigen Skripte für den Micro :Bit können zwar auch aus dem Internet bezogen und auf den Micro :Bit hochgeladen werden, interessanter wird es jedoch für alle Beteiligte, wenn sie vom Schüler selbst programmiert werden. Das Herunterladen von Apps ist vielen Schülern zwar bekannt, das Koppeln des Micro :Bit mit dem Smartphone ist schon eine schwierigere Aufgabe. Das Programmieren ist übersichtlich gestaltet, auch Lehrern soll mit diesem Beispiel etwas die Angst genommen werden. Einziges Problem wird die Beschaffung diverser Apps sein, da sie oft ( zwar nicht viel aber doch) etwas kosten. Das vorliegende Beispiel widmet sich der ersten einer Reihe von recht interessanten Apps einer speziellen Internetplattform.

[Apps für Spielereien mit dem BBC Micro :Bit 1](#)

### **mBlock 5 für mBot, Codey Rocky und Micro:bit**

Die Entwicklerfirma Makeblock hat neben einer Offline-Version der Programmierumgebung mBlock 5 nun auch eine Online-Version dieser Programmierumgebung vorgestellt. Die auf Scratch 3.0 basierende Programmiersprache ermöglicht das Programmieren des Micro:bit aber auch der Roboter mBot und [Codey Rocky](#).

[mBlock 5](#)

### **Roboter mit Tablet oder Smartphone programmieren und steuern**

Viele Apps sowohl im App Store (IOS) als auch im Play Store (Android) stehen für das Programmieren und Steuern von preisgünstigen und vor allem kindgemäßen Robotern mit Tablet oder Smartphone zur Verfügung. Die folgende Tabelle gibt einen Überblick von einsetzbaren Robotern und der dazugehörigen Software.

[Tabelle](#)

### **fischertechnik education - Lernpaket „Calliope mini“**

„fischertechnik“ erleichtert Kindern mit seiner neuen Education-Serie den Einstieg ins Programmieren und verdeutlicht so technische Zusammenhänge unserer digitalisierten Alltagswelt. Im „fischertechnik education“ - Lernpaket „Calliope mini“ sind drei Modelle enthalten. Mithilfe der Bauanleitung lassen sich ein Händetrockner, eine Ampelanlage und eine Schrankenanlage leicht zusammenbauen. Die Modelle werden mit dem „Calliope mini“ gesteuert, der wiederum mit den Programmierumgebungen „Open Roberta“ oder „Makecode“ von Microsoft programmiert wird.

Ein dazugehöriges didaktisches Begleitheft mit Lehrer- und Unterrichtsmaterial bietet eine Schritt-für-Schritt-Anleitung zur Programmierung und verschiedene Aufgabenstellungen. Die Unterrichtsmaterialien sind als kostenloser Download im E-Learning-Portal von „fischertechnik“ verfügbar.

[fischertechnik education](#)  
[Tutorial „Ampelanlage“](#)  
[Tutorial "Händetrockner"](#)  
[Tutorial „Schrankenanlage](#)

### Der mBot

Mit dem Schuljahr 2018/19 stehen uns für die verbindliche Übung **Digitale Grundbildung** 6 neu angekaufte **mBots** zur Verfügung. Ein **mBot** ist ein kostengünstiger, einfach zu bedienender Roboter für Kinder und Jugendliche. Mit ihm kann man praktische Erfahrungen mit Programmierung, Elektronik und Robotik sammeln.

Einige Eigenschaften dieses Roboters: Er muss erst einmal zusammengebaut werden. Die Montage ist jedoch einfach und in kürzester Zeit zu bewerkstelligen. Die Elektronik ist auf einem Controller-Board namens **mCore** aufgebaut, einer Platine, die speziell für **mBot** entwickelt wurde. **mCore** bietet verschiedene **Onboard-Sensoren** wie Summer, Lichtsensor, RGB-LED etc., die uns einen einfachen Einstieg ins Physical Computing ermöglichen.

Für PC oder Laptop stehen uns folgende Programmierertools zur Verfügung: **mBlock 3** (ein Drag-and-Drop Programmierwerkzeug basierend auf **Scratch 2.0**) und **mBlock 5** (ebenfalls eine Drag-and-Drop Programmierumgebung, die jedoch auf dem neu entwickelten **Scratch 3.0** basiert und auch eine Programmierung in **Python** ermöglicht.)

Mit **mBlockly**, **Makeblock** und **mBlockBlockly** stehen uns auch verschiedene Apps für IOS und Android zur Verfügung. Der Roboter kann also auch bequem mit Tablet oder Smartphone programmiert und gesteuert werden.

Der **mBot** kann mit **Bluetooth-Modul** oder mit **2,4 GHz Wireless-Modul** ausgestattet werden. Eine **IR-Fernbedienung** ist beim **mBot** dabei. Mit ihr können drei voreingestellte Fahrweisen („normales Fahren“, „Hindernisse erkennen“ und „einer schwarzen Linie folgen“ ) erprobt werden. Kostengünstige **Add-On-Packs** ermöglichen interessante Roboteraktionen.

[mBot](#)  
[Video](#)

### Microsoft MakeCode und Physical Computing

Microsoft MakeCode ist eine Plattform, die einige webbasierte Programmier-Editoren vor allem für das Physical Computing bietet. Man kann zwischen Block-Editoren - ähnlich wie Scratch - und JavaScript-Editoren - für fortgeschrittene Programmierer - wählen. Derzeit stehen Editoren z.B. für den **BBC Micro:bit**, den **Circuit Playground** von Adafruit, oder den **Chibi Chip** von Chibitronics zur Verfügung, alles durchaus erschwingliche Hardware für Schulen. Auch eine Programmierumgebung für Microsofts **Minecraft** bietet die Plattform.

[Microsoft MakeCode](#)

### Erste Versuche mit Python

Die Website „Trinket“ stellt eine Programmierumgebung für den Internetbrowser zur Verfügung. Die mit dieser Programmierumgebung entwickelten Programme können sofort online ausgeführt werden. Derzeit unterstützt „Trinket“ **Python** und **HTML**. Das fertige Computerprogramm kann man sofort auf verschiedenen sozialen Netzwerken oder per E-Mail mit anderen teilen. Man kann es auch in andere Webseiten oder Blog-Beiträgen einbinden.

[Python lernen mit „Trinket“](#)

### Interessante Projekte mit dem BBC Micro:Bit - "Codierung und Binärsystem"

Es soll in diesem Projekt ein kleines Büchlein für den Micro:bit gebastelt und ein Programm geschrieben werden, mit deren Hilfe wir eine Möglichkeit der Codierung kennen lernen können. Es geht, genauer gesagt, um die Codierung unserer Zahlen aus dem Dezimalsystem. Sie sollen in ein anderes Zahlensystem umgewandelt (codiert) werden. Dieses andere Zahlensystem nennt man das BINÄRSYSTEM. Kennenlernen der Begriffe Codierung, Analog-Elektronik, Digital-Elektronik, Bit, Stellenwerttafel. Arbeitsblätter ermuntern zum Umrechnen vom Dezimalsystem ins Binärsystem und umgekehrt. Es wird ein Büchlein gebastelt, das genau 8 Seiten hat. Warum? Der Micro:bit hat 3 Input/Output-Anschlüsse (0, 1 und 2), wir haben also 3 Bits zur Verfügung. Wir wissen bereits, 3 Bits ergeben 8 Möglichkeiten (Im Binärsystem werden alle Zahlen zur Basis 2 dargestellt, also  $2^3$  ergibt 8) daher sind für unser Micro:bit-Buch 8 Seiten möglich! Unser Büchlein soll auf 7 Seiten eine Geschichte erzählen. Die Seite 0 dient der Erklärung, wie das Buch funktioniert. Ein Script wird erstellt mit dessen Hilfe der Micro:bit immer genau weiß, auf welcher Seite des Büchleins er sich befindet. Krokoklemmern werden an die drei Anschlüsse des Micro:bit angeschlossen, und mit einer Lesekarte verbunden. Die Lesekarte liest die Daten der jeweiligen Seite aus! Mit selbstklebendem Kupferband werden auf den einzelnen Seiten Schaltskizzen geklebt. Je nachdem, ob bei einem Anschluss eine Eingangsspannung herrscht oder nicht, ergeben alle drei Anschlüsse zusammen einen dreistelligen Binärcode (z.B. 001), der Micro:bit stellt auf seinem Display den Inhalt dieser CODIERTEN Seite dar. So kann man kurze Geschichten, Gedichte oder auch Animationen erfinden, die im Büchlein nur CODIERT stehen. Faszinierend! Da für den Außenstehenden etwas schwer nachvollziehbar sei auf das Video verwiesen. Bastelvorlagen erleichtern das Herstellen unseres Micro:bit-Büchleins!

[Video](#)

### Interessante Projekte mit dem BBC Micro:Bit - "Nachrichten senden und empfangen"

Erproben der vielfältigen Möglichkeiten des BBC Micro:Bit - Diesmal: Bluetooth

Die Fähigkeit des Micro:Bit über Bluetooth Nachrichten an einen anderen Micro:Bit oder an mehrere andere Micro:Bits zu senden und umgekehrt, Nachrichten zu empfangen, soll hier den Schülern oder auch interessierten Lehrern vorgestellt und näher gebracht werden. In zwei Projekten werden Skripte für den Micro:bit erprobt, die es ermöglichen, einzelne Buchstaben oder auch ganze Wörter oder Sätze an einen Empfänger zu senden. Danach kann der Empfänger zum Sender werden und auf die empfangene Nachricht antworten. Es werden vor allem diesmal die Begriffe Variable, String und Stringvariable genauer erklärt und durch das anschließende Ausprobieren hoffentlich auch begreif- und durchschaubar gemacht. Mittlerweile besitzt unsere Schule 15 Micro:Bits, sodass jeder Schüler wirklich aktiv mitarbeiten kann. Zwei Videos veranschaulichen die Ergebnisse:

[Video 1](#)

[Video 2](#)

### Eine "Renaissance der Schildkröte"

Es ist derzeit (leider noch nicht in Österreich) eine "Renaissance der Schildkröte" zu beobachten, also einer Wiedergeburt jener schon in den 60er Jahren des 20. Jahrhunderts entwickelten Computer Schildkröte, die zeichnen kann, wenn man ihr nur die richtigen Befehle gibt. Programmiersprachen wie FMS - LOGO, MicroWorlds, Python, Scratch, Turtle Art, SmallBasic, RoamerWorld, Tynker sind heute ganz hoch im Kurs und bieten solch eine Schildkröte oder ein ähnliches Reptil an, mit der man das Programmieren erlernen kann. Entwickelt wurde die Programmiersprache LOGO - die uns zum ersten Mal die Schildkröte vorstellte - vom Vater der Computerpädagogik Seymour Papert im Jahr 1967 am Massachusetts Institute of Technology (MIT). Mittlerweile gibt es unzählige LOGO-Versionen. Zwei neue Versionen stellen wir hier vor  
Mozilla Firefox oder Google Chrome verwenden!

[TurtleCoder](#)

[TurtleStitch](#)

## Der Ozobot

Der Ozobot ist ein kleiner Roboter, den man programmieren kann. Er kann einer von dir gezeichneten schwarzen (roten, grünen, blauen) Linie folgen und verschiedene Farbcodes lesen. Mit Hilfe dieser Farbcodes kannst du den Roboter programmieren. Jeder Farbcode bedeutet für den Ozobot einen Befehl, den er dann ausführt. Es gibt aber auch eine eigene Programmiersprache mit verschiedenen Schwierigkeitsgraden für ihn. Auch Apps kannst du verwenden um den Ozobot z.B. das Tanzen zu lehren oder mit dem Ozobot leichte oder auch recht schwierige Probleme zu lösen! Du kannst den Ozobot mit deinem Smartphone oder deinem Tablet aber auch mit einem PC oder Laptop verwenden!

[Ozobot](#)  
[Video](#)

## Open Roberta Lap

Die [Roberta-Initiative](#) des [Frauenhofer-Instituts](#) hat sich zum Ziel gesetzt, das Interesse besonders von Mädchen (aber natürlich auch von Buben) für die MINT-Fächer nachhaltig zu fördern. „Lernen mit Robotern“ und „Programmieren lernen. Zukunft gestalten“ sind die Wahlsprüche dieser Initiative! Open Roberta Lap ist die Programmierumgebung dieser Roberta-Initiative. Der internetbasierte Editor dieser Open-Source-Plattform erfordert keine Installation. Wir benötigen lediglich eine Internetverbindung und einen -browser. Die verwendete Programmiersprache nennt sich NEPO. Es handelt sich um eine graphische Programmiersprache. Solche graphischen Programmiersprachen sind heute zahlreich vertreten (SCRATCH, BLOCKLY, TURTLE ART, mBLOCK,...), programmiert wird, indem man Befehlsblöcke aneinanderreicht. Da die Programmierung also immer nach dem gleichen Prinzip funktioniert, ist es relativ einfach in eine dieser Programmiersprachen einzusteigen. Das Besondere an NEPO ist, dass man damit verschiedene Roboter (z.B. NXT und EV3 von LEGO) aber auch die uns schon bekannten Mikrokontroller BBC Micro:bit und Calliope mini programmieren kann. Die Programmierumgebung ist unter folgendem Link zu erreichen:

[Open Roberta Lap](#)

## Der Calliope mini

Der **Calliope mini** ist ein Mikrokontroller, ein Ein-Chip-Computer, der ähnlich wie der **BBC Micro:bit** oder auch der **CodeBug** Kinder spielerisch in die Welt des Programmierens einführen soll. Konkret geht es um Grundschüler, die das Programmieren lernen und so auf die Digitalisierung unserer Welt vorbereitet werden sollen. In Deutschland will man diese Platine an (vielleicht später alle) Drittklässler verteilen. In Großbritannien wurde bereits an eine Million Schüler der BBC Micro:bit kostenlos verteilt. Und in Österreich?

[Calliope](#)  
[Video](#)  
[Calliope mini-Editor](#)  
[PXT-Editor](#)

## CodeBug

Der **CodeBug** ist ein kleiner Mikrokontroller in Form eines Käfers, mit dem man die Grundlagen von Programmierung, Physical Computing und Elektronik lernen kann. Er besitzt 25 quadratisch angeordnete LEDs, zwei Buttons und sechs Kontakte, die als „Käferbeine“ bezeichnet werden. Vier dieser Beine können als Sensoren verwendet werden. Die mit einem Online-Programm-Editor entwickelten Programme können über eine Micro-USB-Schnittstelle in den CodeBug geladen werden.

Der CodeBug wird nämlich auf dem Desktop PC als gewöhnlicher Speicherstick erkannt. Programme können, nachdem sie erstellt und auf dem PC abgespeichert wurden, auf den CodeBug auf herkömmliche Art kopiert werden. Über diese Schnittstelle wird der Mikrokontroller auch mit Strom versorgt! Der CodeBug kann auch mittels Batterie (Knopf-Batterie – Typ 2032) mit Strom versorgt werden. Über eine Buchsenleiste kann der CodeBug auf die entsprechenden Pins der GPIO-Leiste eines Raspberry Pi angeschlossen werden. Im Schuljahr 2017/18 soll der CodeBug auf seine Brauchbarkeit im Informatikunterricht getestet werden.--- [Handout 1](#) --- [Handout 2](#)

### [CodeBug](#)

#### **BBC micro:bit**

Der **BBC micro:bit** ist ein nur 4 cm x 5 cm großer Mikrokontroller, der ähnlich wie der CodeBug funktioniert. Er wird ebenfalls über eine Micro-USB-Buchse mit Strom versorgt. Ein gesondert zu erwerbendes Batteriefach ermöglicht mit 2 AAA-Batterien ebenfalls die Stromversorgung. Wie der CodeBug hat der Micro:bit eine 5 x 5 LED-Matrix zur Anzeige von blinkenden Texten, alphanumerischen Zeichen und Mustern. Er besitzt zwei programmierbare Tasten, die man z.B. als Game-Controller verwenden kann. Weiters hat er fünf Ein- und Ausgänge, zwei dienen der Stromversorgung, drei stehen zur Steuerung von Motoren oder Robotern zur Verfügung. Das Besondere ist die Bluetooth-Technologie. Man kann einen Micro:bit mit anderen Micro:bits verbinden, mit Smartphones oder Tablets. Eine spezielle App ermöglicht es, Programme zu entwickeln und über Bluetooth auf den Micro:bit zu übertragen. Über USB wird der Micro:bit genauso wie der CodeBug auf dem Desktop PC als gewöhnlicher Speicherstick erkannt. Programme können, nachdem sie erstellt und auf dem PC abgespeichert wurden, auf den Micro:bit auf herkömmliche Art kopiert werden. Über 20 Input/Output Pins kann der Micro:bit mit anderen Geräten wie dem Raspberry Pi verbunden werden. Programmiert wird er entweder mit dem „JavaScript-Blocks-Editor“ von Microsoft oder – wenn man ein Tablet verwendet - mit „Microsoft Touch Develop“. Auch der Micro:bit wird im Schuljahr 2017/18 erprobt.

### [BBC Micro:bit](#)

#### **Der Calliope mini**

Der **Calliope mini** ist ein Mikrokontroller, ein Ein-Chip-Computer, der ähnlich wie der **BBC Micro:bit** oder auch der **CodeBug** Kinder spielerisch in die Welt des Programmierens einführen soll. Konkret geht es um Grundschüler, die das Programmieren lernen und so auf die Digitalisierung unserer Welt vorbereitet werden sollen. In Deutschland will man diese Platine an (vielleicht später alle) Drittklässler verteilen. In Großbritannien wurde bereits an eine Million Schüler der BBC Micro:bit kostenlos verteilt. Und in Österreich?

### [Calliope](#) [Video](#)

### [Calliope mini-Editor](#) [PXT-Editor](#)

#### **DIY-Kits von KANO**

Das Londoner Startup KANO will mit seinen **DIY-Kits** mehr Kinder und Jugendliche zum Programmieren bringen. Die mit einer Designer-Firma entwickelten (und deshalb auch etwas teuren)



derzeit vier recht stylischen Baukästen sind ein [Computer-Kit](#), ein [Screen-Kit](#), ein [Pixel-Kit](#) und ein [Motion Sensor-Kit](#). Die einzelnen Kits sind recht einfach zusammenzubauen. Für unsere jüngeren Schüler interessant sind vor allem das **Pixel-Kit** und das **Motion Sensor-Kit**. Das Pixel-Kit ist ein interaktives Board mit 128 LEDs, mit dem man Muster, Daten, Animationen und Spiele darstellen kann. Das Pixel-Kit kann mit einem Bewegungssensor verbunden werden, sodass die LEDs auch auf Gesten reagieren können. Nach dem Zusammenbau wird das Pixel-Kit mittels USB an den Computer angeschlossen. Eine App („[Kano App](#)“) wird von der Kano-Webseite heruntergeladen, installiert und schon kann man beginnen, das Pixel-Kit zu programmieren. Bei der Programmiersprache handelt es sich um eine einfache ikonische Programmiersprache (Befehlsblöcke werden zusammengefügt). Ein **Emulator** zeigt dem Programmierer sofort die Ergebnisse seiner Arbeit und ermöglicht jederzeit ein Testen und Korrigieren derselben. Ein Account auf dem Kano-Server muss eröffnet werden, um die fertigen Projekte in diesem Account speichern zu können. Erst dann kann man sie in das Pixel-Kit laden!

### Code or be coded

Mozilla Firefox oder Google Chrome verwenden!

[Programmieren mit Kano](#)

[CodeBug](#)

[Micro: bit](#)

[Scratch](#)

[Blockly](#)

[Lightbot](#)

[Snap](#)

[Make Art](#)

[Make Pong](#)

[Code-Studio](#)

[Tynker](#)

### Raspberry Pi

Der **Raspberry Pi** und seine Klone der **Banana Pi** und der **Orange Pi** sind **Einplatinencomputer** in der Größe einer Kreditkarte. Es soll gezeigt werden, wie sie im Unterricht sinnvoll eingesetzt werden können. Sie eignen sich als **Desktopcomputer** mit Browser, Officepaket, Lernprogrammen aber auch - und deshalb sind sie ursprünglich auch entwickelt worden - zum **Programmieren** (Scratch, Python, Turtle Art,...) Auch für elektronische Spielereien und Basteleien sind diese Geräte ganz toll geeignet. - -- [Handout](#)

[Raspberry Pi](#)

[Banana Pi](#)

[Orange Pi](#)

### Physical Computing

Der **Raspberry Pi** macht es uns sehr einfach, in die Welt des „Physical Computing“ einzusteigen, indem man eigene Projektideen mit Hilfe von computergesteuerten Sensoren, Tasten oder Leuchtdioden realisiert. Er besitzt nämlich eine so genannten „**General Purpose Input / Output**“-Schnittstelle (GPIO), mit deren Hilfe sich weitere elektronische Bauteile an den Raspberry Pi anschließen und steuern lassen. **Add-on-Boards** für den Raspberry Pi wie die **LedBorg-Platine** oder auch **Pibrella**, der **Makey Makey** oder das **PicoBoard** ermöglichen zusammen mit der

Programmiersprache Scratch recht interessante ja manchmal sogar recht verrückte Projekte. ---  
[Handout](#)

### **Roboter mit Smartphone oder Tablet steuern**

[Cubelets](#)  
[Sphero, Ollie und BB-8](#)  
[Dash and Dot](#)  
[Robo Wunderkind](#)  
[Ozobot](#)  
[mBot](#)

### **Das PicoBoard**

Das PicoBoard ist eine kleine rote Platine für die Programmiersprache SCRATCH, die über USB an den Computer angeschlossen werden kann. Sie hat Sensoren für Licht und Schall, einen Schieberegler, einen Schalter und vier Anschlüsse (Widerstandseingänge) für weitere Sensoren oder Schalter. So können Licht, Klänge, und elektrische Widerstände in SCRATCH-Projekte eingebunden werden. Im Beispielordner von SCRATCH sind einige PicoBoard-Projekte zu finden.

[Picoboard](#)

### **Das MaKey MaKey**

Das "MaKey MaKey" ist ein Micro-Controller-Board, also eine kleine Platine mit einem Micro-Controller, im speziellen Fall basierend auf einem Arduino-Controller. Die Platine gibt sich dem PC gegenüber als „Human Interface Device“ (HID) aus: Sie tut so, als wäre sie eine Tastatur und eine Maus. Tastaturbefehle und Mausklick können durch externe Quellen, die den Strom leiten, übernommen werden.

Bei den Tastaturbefehlen handelt es sich um sechs Buchstabentasten (W, A, S, D, F und G) und die vier Pfeiltasten sowie die Leertaste und dem Mausklick.

Man kann mit dieser Technik leitfähige Gegenstände (z.B. Plastilin) und Lebensmittel (z.B. Bananen) in Eingabegeräte für Spiele am Rechner verwandeln. Das funktioniert bei den Betriebssystemen Windows, MacOS und Linux. Auch ein Raspberry Pi lässt sich damit steuern

Mit einem USB-Kabel wird das MaKey MaKey an den Rechner angeschlossen

Dann stellt man mit einem Kabel mit einer Krokodilklemme eine Verbindung zwischen dem Eingang „Space“ des MaKey MaKey mit einem leitfähigen Gegenstand (z.B. mit einer Banane) her und kann nun die Banane als Space-Taste verwenden

Den Kontakt stellt man her, indem man den eigenen Körper direkt mit dem Masseanschluss der Makey-Makey-Platine ("Earth") verbindet (z.B. durch ein Erdungsarmband) und dann den leitenden Gegenstand berührt, von dem ja das Kabel mit der Krokodilklemme zurück zum Makey Makey führt. Die Belegung der Tastaturbefehle kann über eine kostenlose Arduino-Software geändert werden.

[Makey Makey](#)