

Physical Computing im Modul Medien und Informatik

Prof. Dr. Mareen Przybylla

Goldau, 16. Juli 2021

Blockwochen "Medien und Informatik" (1/3)

Tagesablauf und Materialien

| Zeit | Dauer | Beschreibung |
|-------|-------|--|
| 09:00 | 30 | <p>Einführung: Physical Computing im Schulfach Medien und Informatik.</p> <ul style="list-style-type: none"> • Fachliche und fachdidaktische Überlegungen • Bezüge im Lehrplan 21 • Werkzeuge für den Unterricht • Leitfaden zur Projektplanung • Beispielprojekte |
| 09:30 | 30 | Praxis I: BBCmicro:bit |
| 10:00 | 15 | Pause |
| 10:15 | 75 | <p>Praxis II: Arduino</p> <ul style="list-style-type: none"> • Snap4Arduino • Arduino • TinkerKit und Grove-System • Installation der nötigen Software (Dateien siehe rechts): <ul style="list-style-type: none"> ◦ Arduino IDE installieren, modifizierte Firmata auf den Arduino spielen ◦ Snap4Arduino installieren, Block-Bibliothek importieren ◦ in Snap4Arduino auf "Arduino verbinden" klicken, Happy Prototyping! • erste Mini-Challenges mit Snap4Arduino |

| Zeit | Dauer | Beschreibung |
|-------|-------|--|
| 11:30 | 60 | Mittagessen |
| 12:30 | 30 | <p>Didaktische Überlegungen zur Projektarbeit mit agilen Methoden der Softwareentwicklung:</p> <ul style="list-style-type: none"> • Project Board • User Stories • Tasks • Projektarbeit in Iterationen • Stand-Up-Meetings • Pair Programming |
| 13:00 | 75 | Praxis III: Planung und Durchführung eines eigenen Physical-Computing-Projekts |
| 14:15 | 15 | Pause |
| 14:30 | 30 | Projektpräsentation und Reflexion |
| 15:00 | 15 | Klärung offener Fragen und Abschluss |
| 15:15 | 15 | Aufräumen |
| 15:30 | | Ende des Kurstages |

<https://mia.phsz.ch/Sekundarstufe/FreitagJuli21>

Physical Computing

Begriffsklärung

- Physical Computing ist die kreative Gestaltung und Entwicklung interaktiver Objekte
 - programmierte, gegenständliche Artefakte
 - kommunizieren über Sensoren und Aktoren mit ihrer Umgebung
 - erfüllen bestimmten Zweck
 - Verhalten ist als Software beispielsweise auf einem Mikrocontroller implementiert
 - laufen kontinuierlich und interagieren stetig mit ihrer Umgebung
 - können vernetzt werden

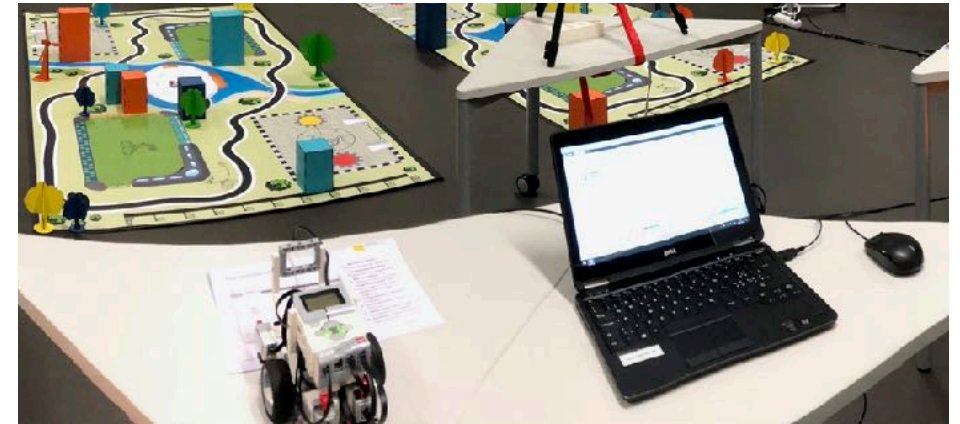


Helligkeitsgesteuerter farbwechselnder Blumentopf

Physical Computing

Abrenzung zur Robotik

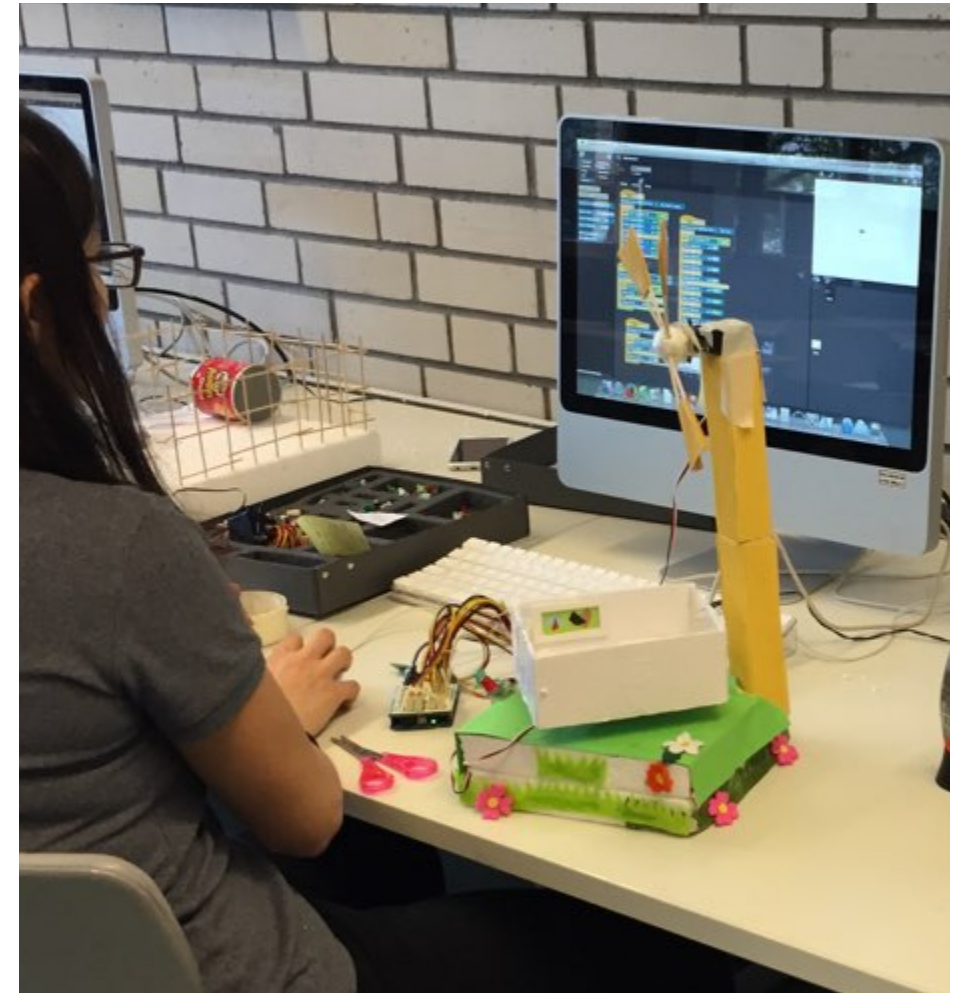
- Roboter im Unterricht sehr beliebt
- Ähnlich wie beim Physical Computing Verbindung von physischer und virtueller Welt mit Sensoren und Aktoren
- Hauptunterschied: Anwendungen und Gestaltungsprozesse
- Robotik fokussiert stärker auf Themen wie
 - physische Tätigkeiten
 - Orientierung und Bewegung in unbekanntem/schwierigem Gelände, autonome Navigation
 - Vermeidung von Kollisionen mit Hindernissen
- Typische Beispielprodukte: Industrieroboter, Roboterarme, Drohnen, Fahrzeuge, humanoide Roboter
- Produkte des Physical Computings sind vielfältiger, z. B. E-Textilien, interaktiver Schmuck, intelligente Spielzeugtiere, sensorgesteuerte Bewässerungsanlagen, Stimmungslampen, Smart-Home-Anwendungen



Physical Computing

Nutzen für den Schulunterricht

- Nimmt den Fokus von der reinen Softwareentwicklung, beinhaltet Aspekte des Hardwaredesigns
 - Verbindet die virtuelle und die reale Welt
 - Macht Artefakte des Lernens sichtbar, anfassbar und “mittelbar”
 - Ermöglicht konstruktivistisches und kreatives Lernen
 - legt viele fächerübergreifende Bezüge nahe
- ➔ Lernende werden zu Erfindern und wenden Ideen und Methoden des *Embedded Systems Designs* in kreativen, konstruktivistischen Unterrichtsettings an



Physical Computing

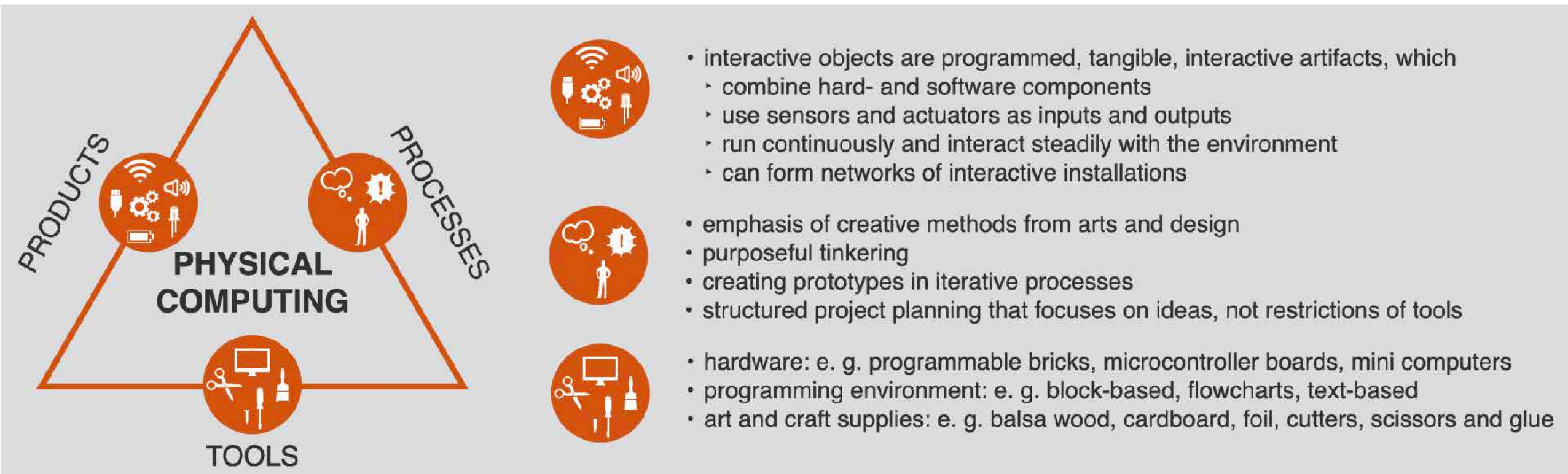
Bezüge im Modullehrplan Medien und Informatik des Lehrplan 21 (Zyklus 3)

Schülerinnen und Schüler ...

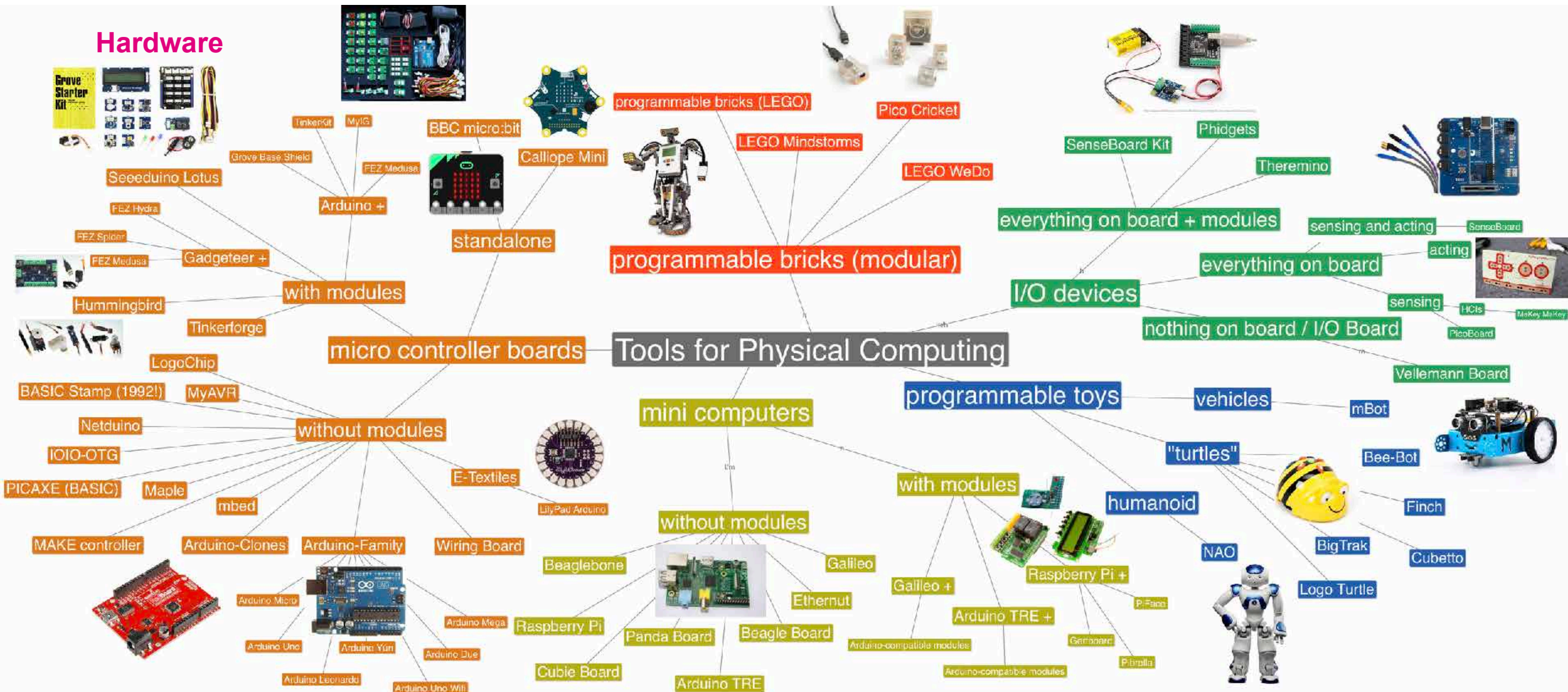
- können logische Operatoren verwenden (und, oder, nicht). *[Kompetenzbereich Datenstrukturen]*
- können selbstentdeckte Lösungswege für einfache Probleme in Form von lauffähigen und korrekten Computerprogrammen mit Schleifen, bedingten Anweisungen und Parametern formulieren. *[Kompetenzbereich Algorithmen]*
- können selbstentwickelte Algorithmen in Form von lauffähigen und korrekten Computerprogrammen mit Variablen und Unterprogrammen formulieren. *[Kompetenzbereich Algorithmen]*
- kennen die wesentlichen Eingabe-, Verarbeitungs- und Ausgabeelemente von Informatiksystemen und können diese mit den entsprechenden Funktionen von Lebewesen vergleichen (Sensor, Prozessor, Aktor, Speicher). *[Kompetenzbereich Algorithmen]*

Physical Computing

Dreieck des Physical Computings

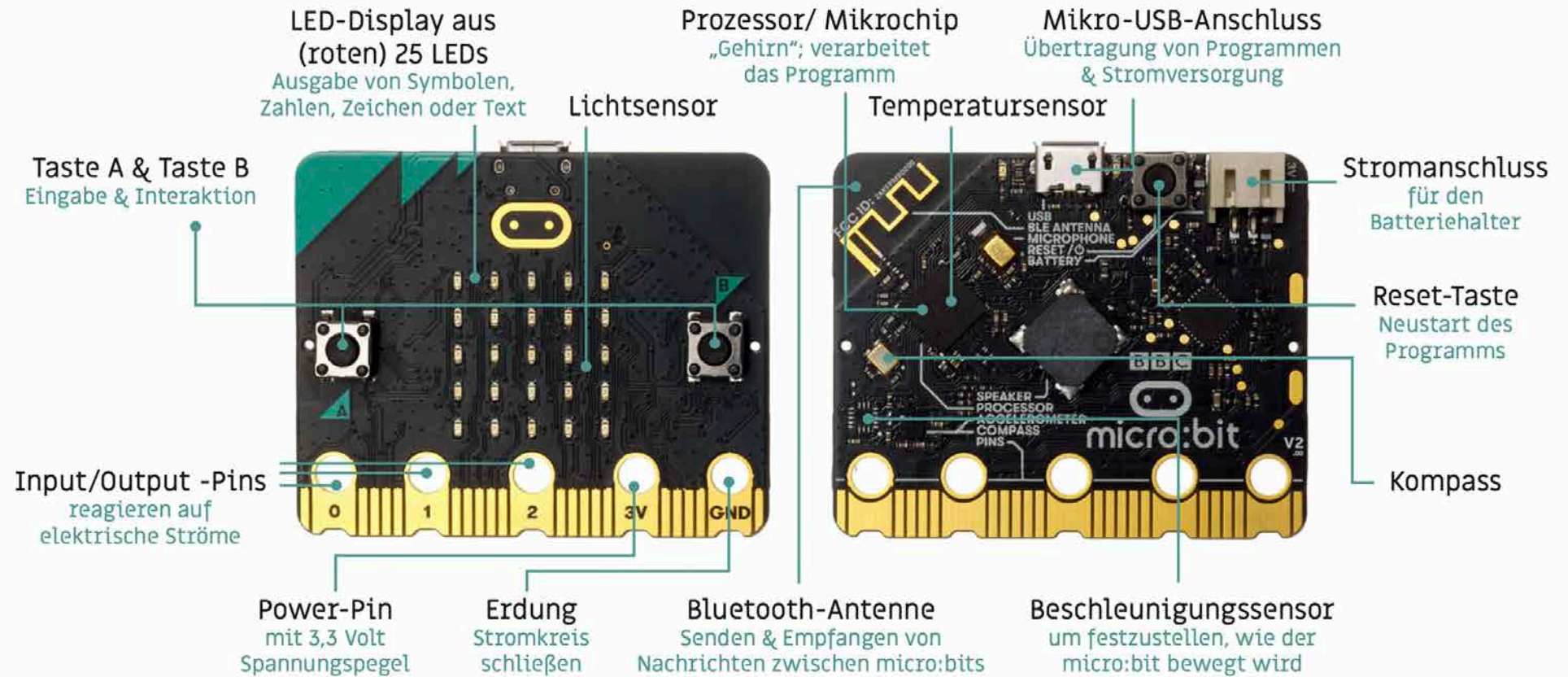


Unterrichtswerkzeuge



Unterrichtswerkzeuge

micro:bit



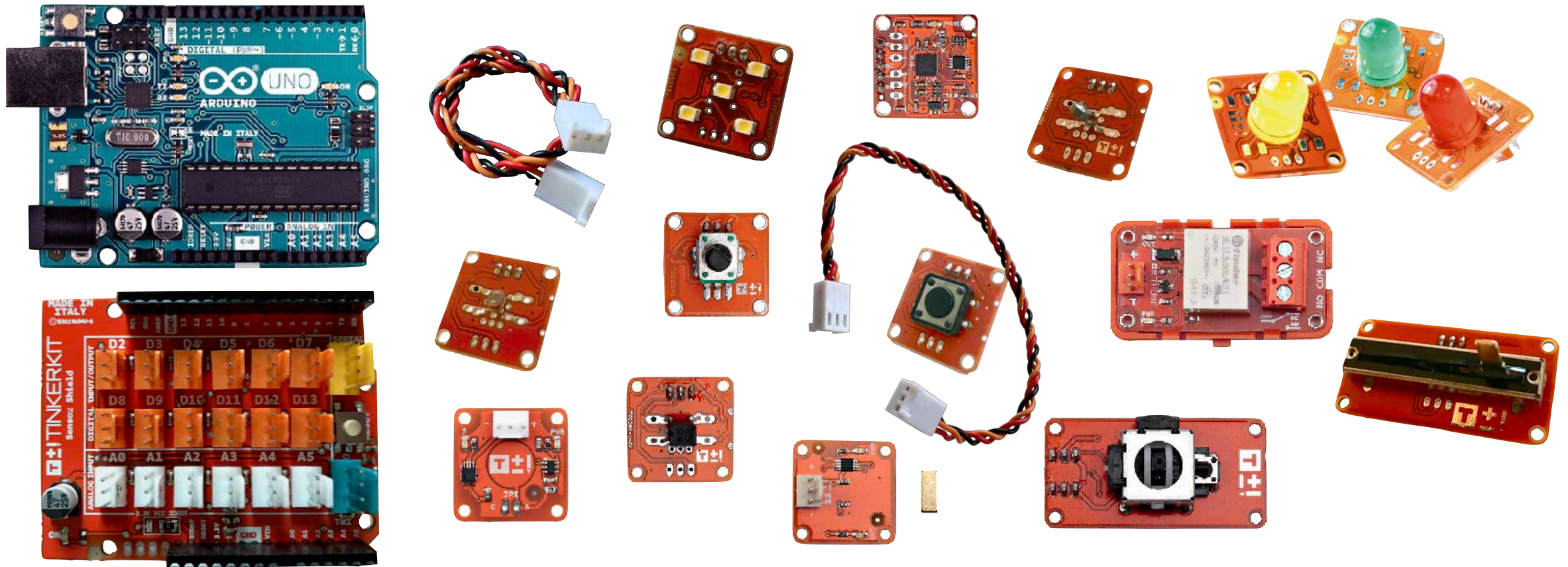
TüftelAkademie | 1. Überarbeitung: Digital Literacy Lab
CC-BY-SA 4.0: <https://creativecommons.org/licenses/by-sa/4.0/deed.de>



<https://tueftelakademie.de/wp-content/uploads/2021/02/202101-lernkartenmicrobits-01.pdf>

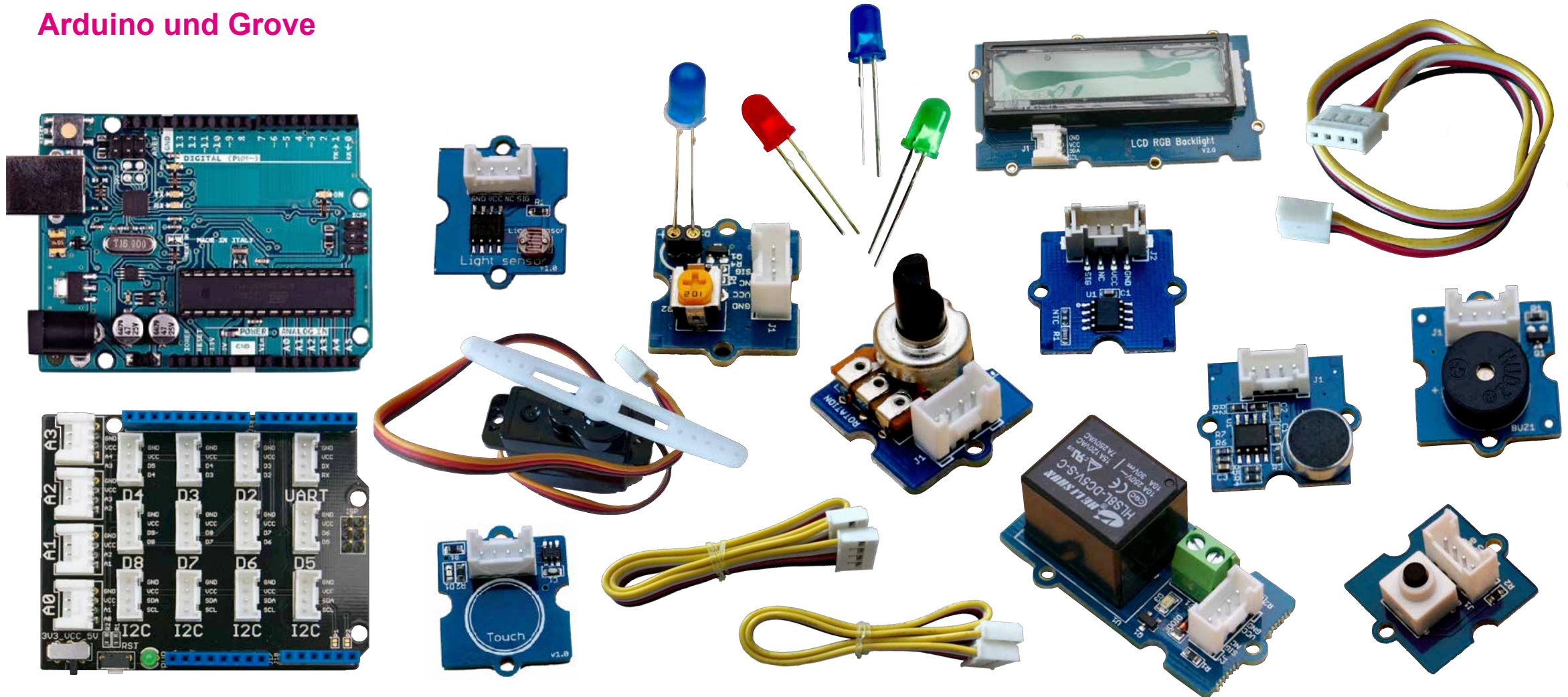
Unterrichtswerkzeuge

Arduino und TinkerKit



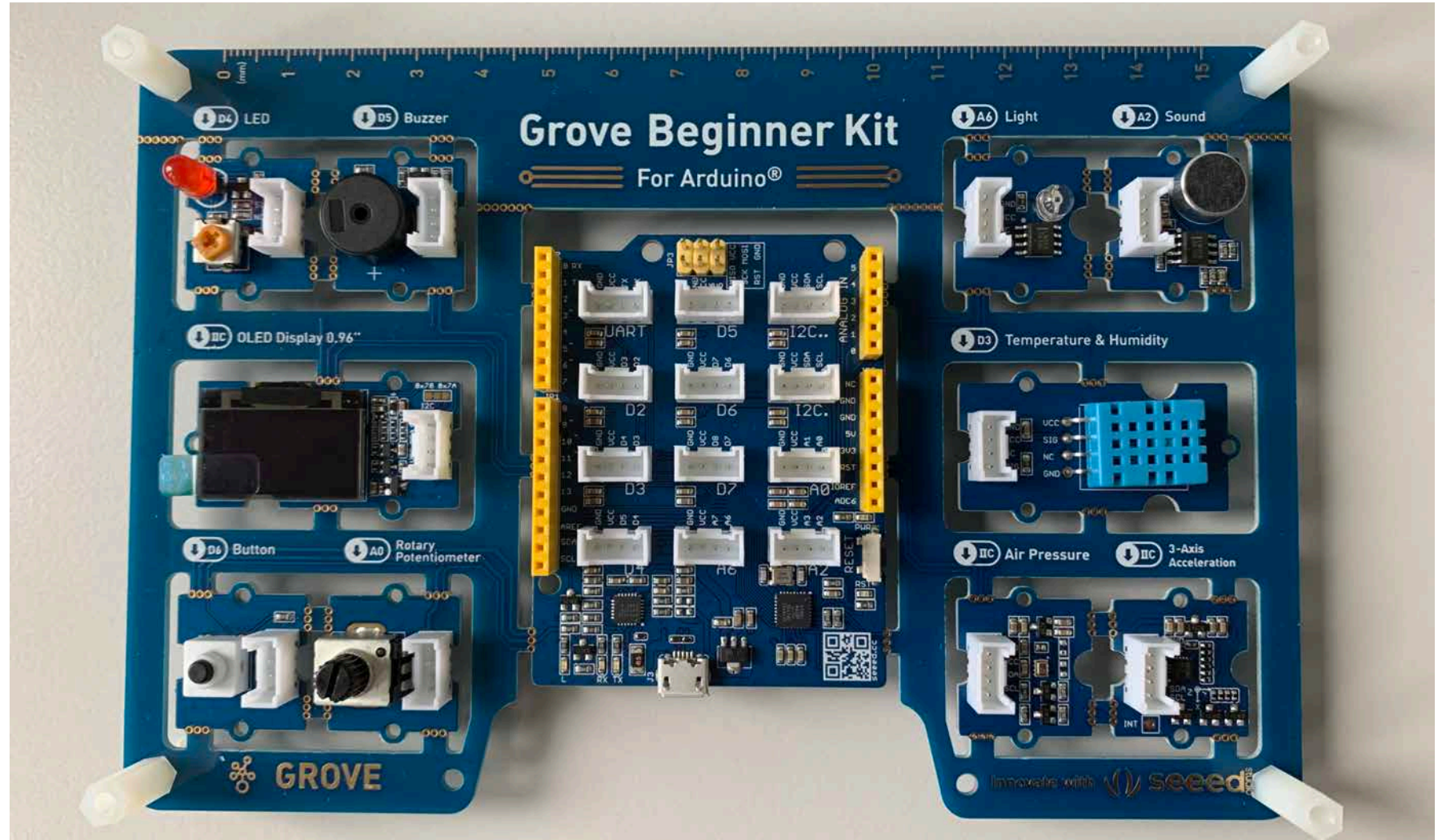
Unterrichtswerkzeuge

Arduino und Grove



Unterrichtswerkzeuge

Arduino und Grove

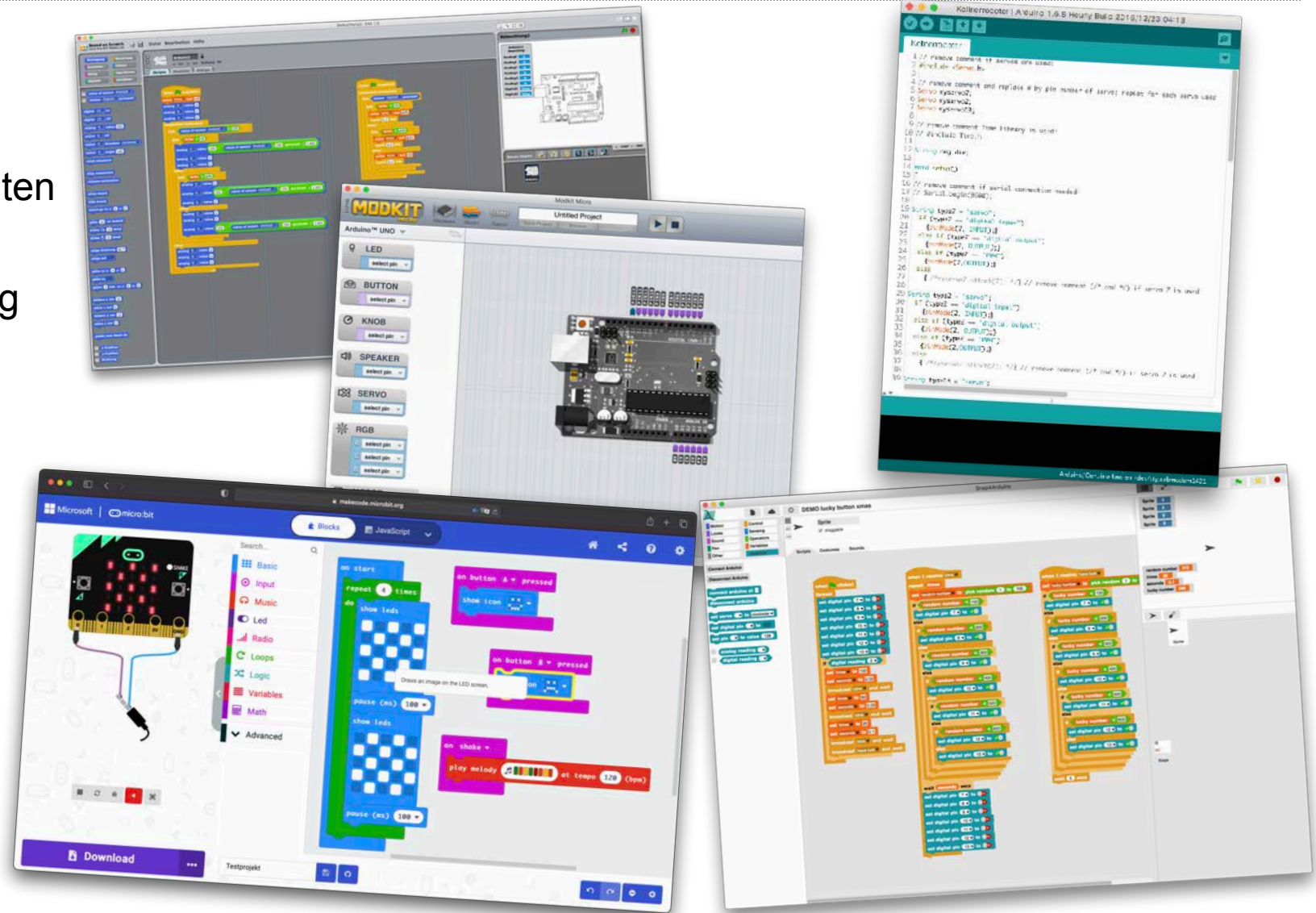


Unterrichtswerkzeuge

Programmierumgebungen

- Nahezu unbegrenzte Möglichkeiten
- Wahl der Hardware beeinflusst Wahl der Programmierumgebung
- Diskussion zur blockbasierten Programmierung: in “echter” Entwicklerwelt wird ebenfalls abstrahiert (Implementation in höherer Sprache)

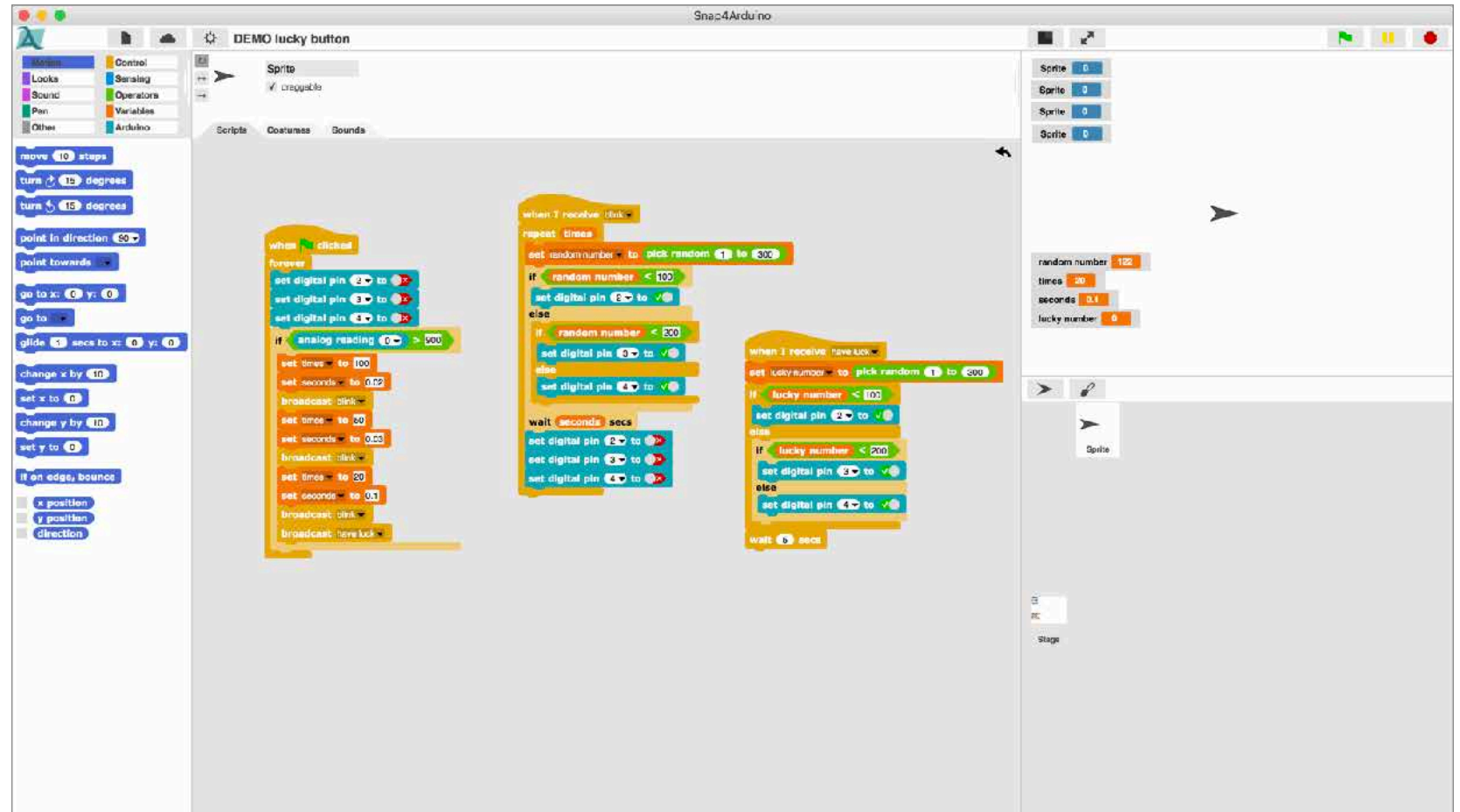
→ in Schule (Sekundarstufe I)
z. B. mit visuellen Umgebungen



Unterrichtswerkzeuge

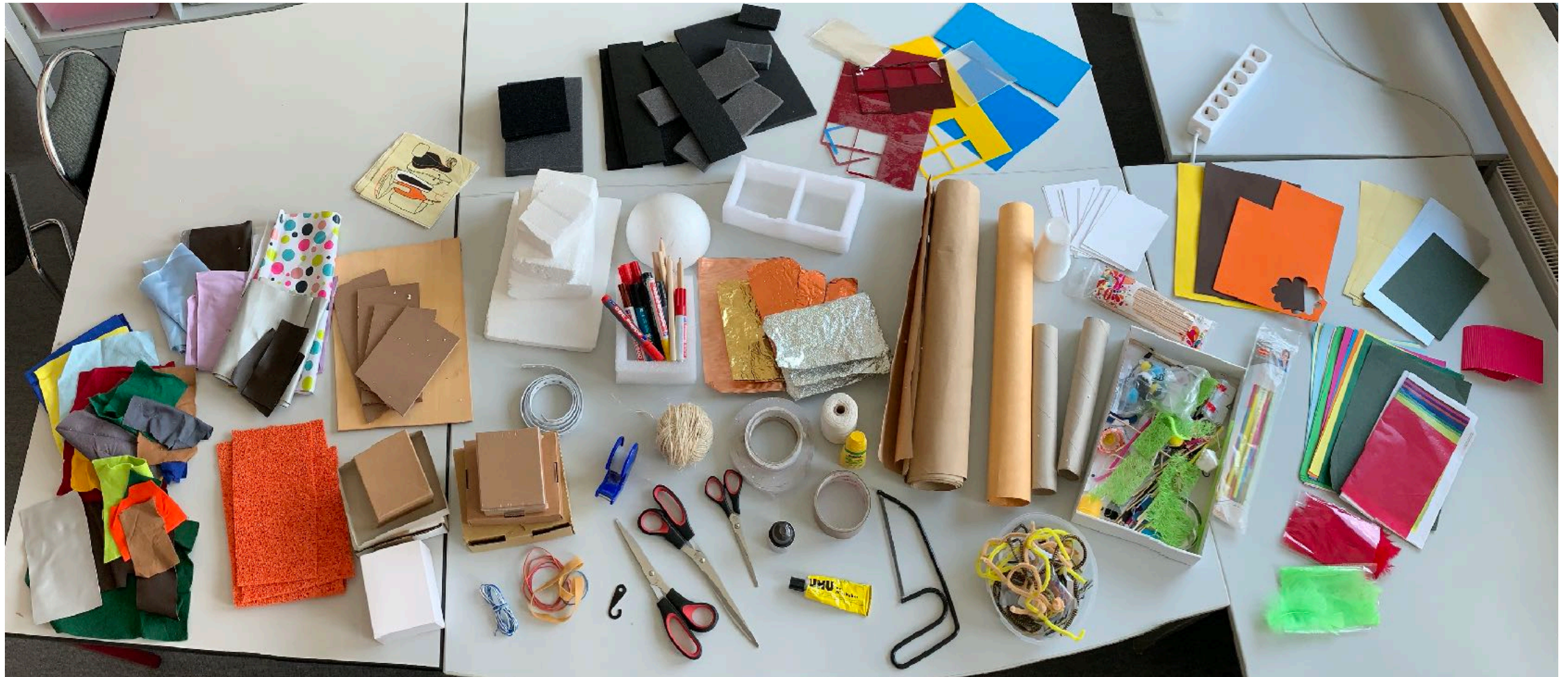
Snap4Arduino

- ✓ Vorteile blockbasierter Programmierung
- ✓ flexibel erweiterbar
- ✓ niederschwelliger Zugang
- ✓ intuitive Bedienung
- keine "echte" Programmierung, USB-Verbindung zwischen Computer und Mikrocontroller muss bestehen bleiben



Unterrichtswerkzeuge

Künstler- und Bastelmaterial



Unterrichtsumsetzung

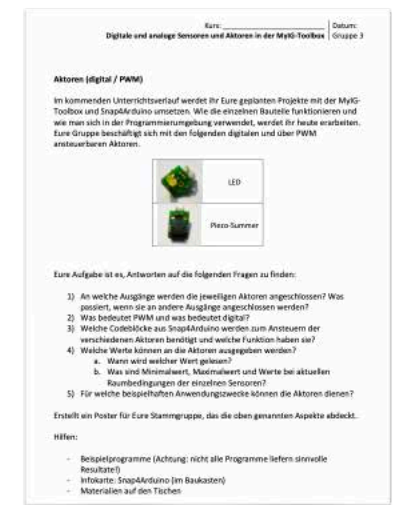
Gestaltungsprinzipien für Physical-Computing-Projekte

1. Tinkering-Aktivitäten in dedizierte Lernphasen zur Aneignung von Inhalten und Fähigkeiten integrieren
2. Lernende eigene interaktive Objekte herstellen lassen (“Informatisches Töpfern”)
3. Lernende funktionierende Prototypen entwickeln lassen
4. interessante Themen anbieten: Offenheit, um Ideen und Kreativität anzuregen
5. kreative Methoden integrieren
6. fachliche Aspekte mit Kunst/Basteln verbinden
7. Gerüste zur Strukturierung der Prozesse der Projektarbeit bereitstellen:
 - a) Planung aus Nutzersicht
 - b) Planung aus Entwicklersicht (nicht-technische und technische Perspektive)
8. Wahl geeigneter Baukästen und Programmierumgebungen für die Zielgruppe (“low floors”, “wide walls”, “high ceilings”)
9. Angebot geeigneter Werk- und Bastelmaterialien für die intendierte Projekte
10. Vorbereitung einer gemeinsamen Ausstellung aller interaktiven Objekte
11. Präsentation der Ergebnisse vor Publikum

Unterrichtsumsetzung

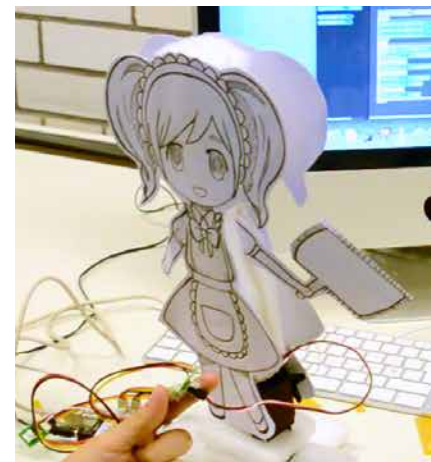
Anwendungsbeispiel "My Interactive Garden"

- Lernumgebung für Physical Computing-Unterricht
- vorgestellte Leitlinien werden umgesetzt, z. B.:
 - *breite Themen*: interaktive Objekte eines interaktiven Gartens herstellen
 - *Kollaboration* wird durch die gemeinsame Ausstellung des interaktiven Gartens betont
 - *intuitiver Baukasten*: Arduino + "Shield" + vorgefertigte Sensor- und Aktormodule
 - *Bastelmaterial*, Arbeitsblätter etc.



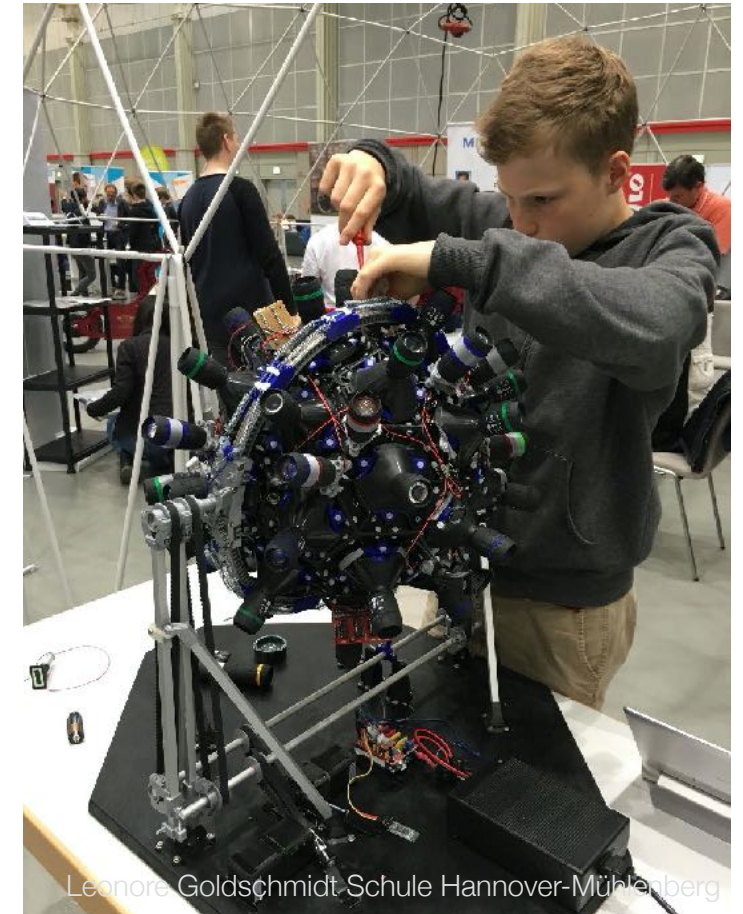
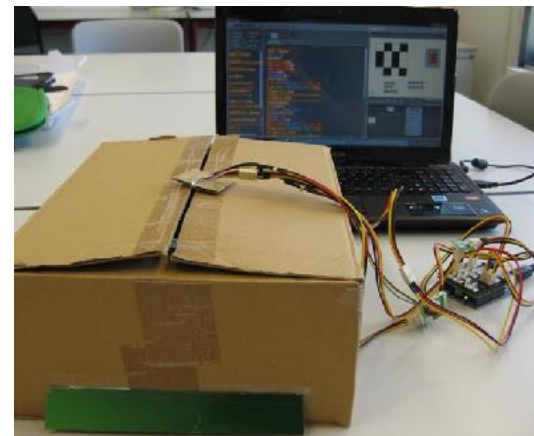
Unterrichtsumsetzung

Weitere Beispielprojekte



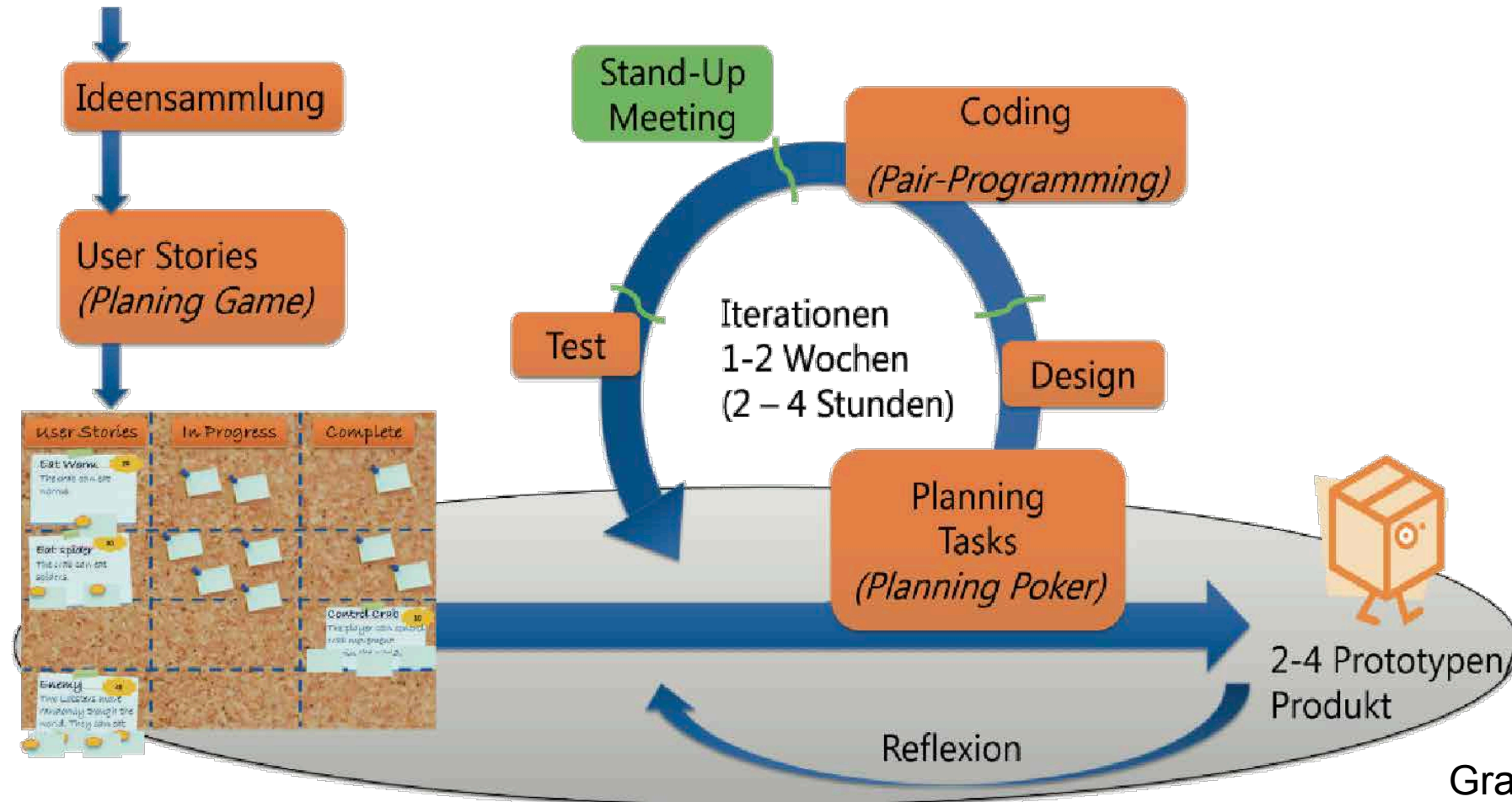
Unterrichtsumsetzung

Beispielprojekte



Projektarbeit

Agile Methoden der Softwareentwicklung



Grafik: Petra Kastl

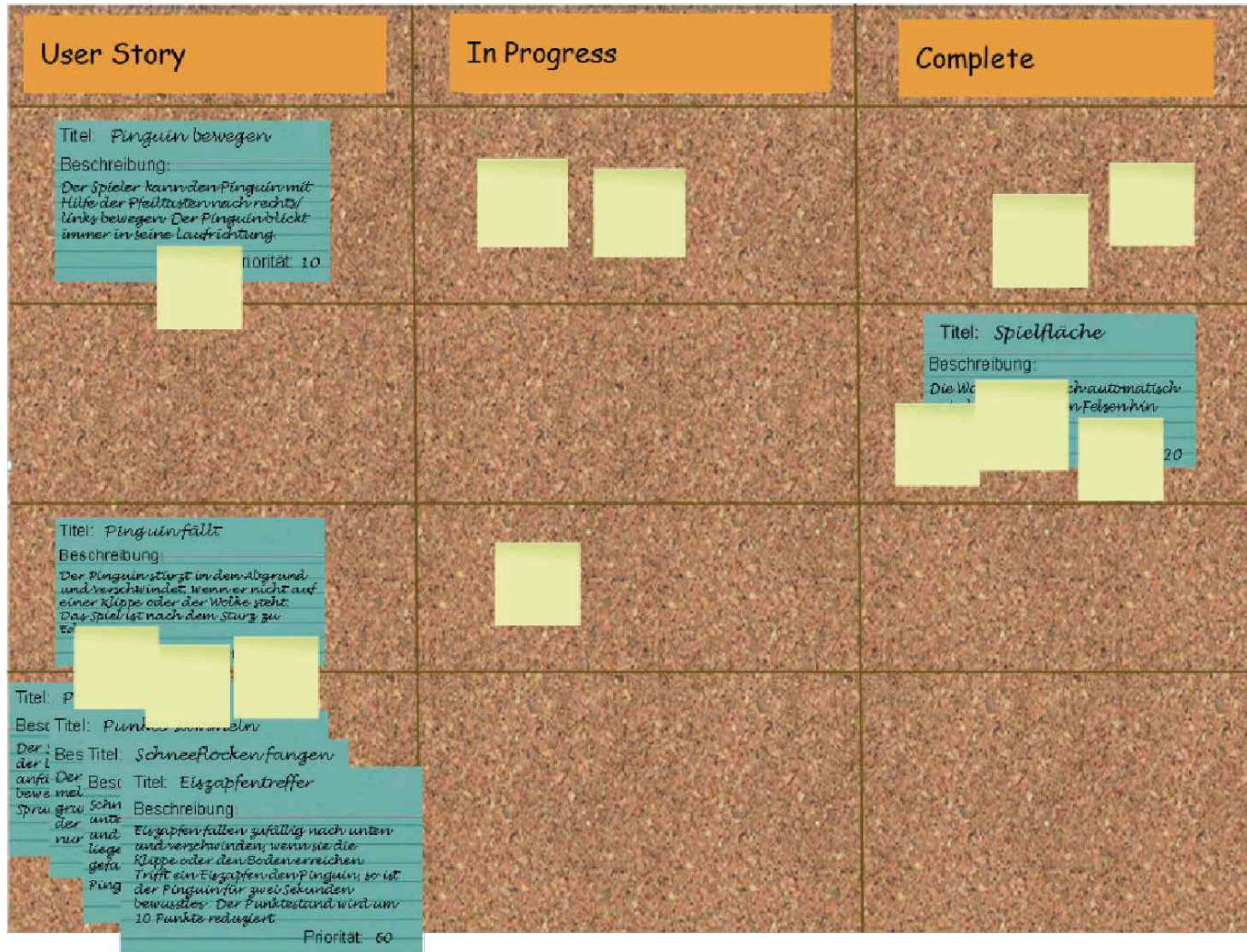
Projektarbeit

Pair-Programming

- Programmierpaare:
 - *Driver* bedient die Tastatur und erklärt, was er/sie sich bei der Programmierung denkt
 - *Navigator* verfolgt aufmerksam das Geschehen und überlegt sich, ob es eine bessere oder elegantere Lösung gibt
- Alle 7 Minuten wird gewechselt

Projektarbeit

Projektboard



Grafik: Petra Kastl

Projektarbeit

Stand-Up-Meeting

- Regelmäßiges Treffen zur Klärung von Aufgaben, Problemen, etc.:
 - Teammitglieder treffen sich am Projektboard
 - Jedes Team berichtet kurz den Stand (je max. eine Minute)
 - Bei Bedarf werden Meetings zur detaillierteren Besprechung einberufen