

Physical Computing im Modul Medien und Informatik

Prof. Dr. Mareen Przybylla

Goldau, 08. Oktober 2020

Blockwochen "Medien und Informatik" (2/3)

Tagesablauf und Materialien

<https://mia.phsz.ch/Sekundarstufe/DonnerstagOktober20>

| Zeit | Dauer | Beschreibung | Methode und Sozialform |
|-------|-------|---|---------------------------------|
| 09:15 | 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 | Vortrag, Plenum |
| 09:45 | 30 | <p>Praxis I: Werkzeuge kennenlernen:</p> <ul style="list-style-type: none"> • Snap!Arduino • Arduino • TinkerKit • Installation der nötigen Software | Tutorial, Einzel-/Partnerarbeit |
| 10:15 | 30 | Pause | |
| 10:45 | 60 | <p>Praxis II: Tutorial und Gruppenpuzzle</p> <ul style="list-style-type: none"> • ausschnittweise Durchführung des Gruppenpuzzles bzw. Stationenlernens • Diskussion und Erfahrungsaustausch | Gruppenarbeit, Plenum |
| 11:45 | 60 | Mittagesien | |

| | | | |
|-------|----|--|---------------------------------|
| 12:45 | 30 | <p>Didaktische Überlegungen zur Projektarbeit mit agilen Methoden der Softwareentwicklung:</p> <ul style="list-style-type: none"> • Project Board • User Stories • Taski • Projektarbeit in Iterationen • Stand-Up-Meetings • Pair Programming | Vortrag, Plenum |
| 13:15 | 90 | Praxis III: Planung und Durchführung eines eigenen Physical-Computing-Projekts | Pair Programming, Gruppenarbeit |
| 14:45 | 20 | Pause | |
| 15:05 | 30 | Projektpräsentation und Reflexion | Plenum |
| 15:35 | 10 | Klärung offener Fragen und Abschluss | |
| 15:45 | 15 | Aufräumen | Teamwork :-) |
| 16:00 | | Ende des Kurstages | |

Ausgangslage

Phänomene der vernetzten «Digitalen Welt»

- *Ubiquitous Computing*: interaktive und eingebettete Informatiksysteme dominieren unseren Alltag in vielen Bereichen
- Computer haben sich weiterentwickelt: sie sind häufig kaum präsent wahrnehmbar, dennoch extrem leistungsfähig und allgegenwärtig
- Zahlreiche Anwendungsgebiete, z. B. Robotik, Heimautomatisierung, Verkehr und Transport, Wearables
- Neue Anforderungen an Schulen, die das Ziel verfolgen, Lernende dazu zu befähigen, die Welt in der sie leben zu verstehen und mitzugestalten



<http://www.iotphils.com/wp-content/uploads/2014/07/Smart-Cities.png>

Physical Computing

Begriffsklahrung I - Ursprung

- Kunstler und Designer nutzen programmierbare Hardware zur Herstellung *interaktiver Objekte* und Installationen
 - programmierte, gegenstandliche Artefakte
 - kommunizieren uber Sensoren und Aktoren mit ihrer Umgebung zur Erweiterung des Spektrums der Dinge, die sie wahrnehmen und auf die sie reagieren konnen
 - fokussieren auf Ideen und intendierte Interaktionen
- *Tinkering* als Herangehensweise: Verbessern und erweitern existierender Hard- und Software auf experimentelle Weise, angetrieben durch Neugierde, Vorstellungskraft und Kreativitat

[cf. O'Sullivan & Igoe, 2004, Banzi 2011]



Physical-Computing-Projekte bei der Make Munich 2016

Physical Computing

Begriffsklärung II - Schulkontext

- 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



Physical Computing

Begriffsklärung II - Schulkontext

- 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



Helligkeitsgesteuerte Farbwechselnder Blumentopf

Physical Computing

Abrenzung zur Robotik

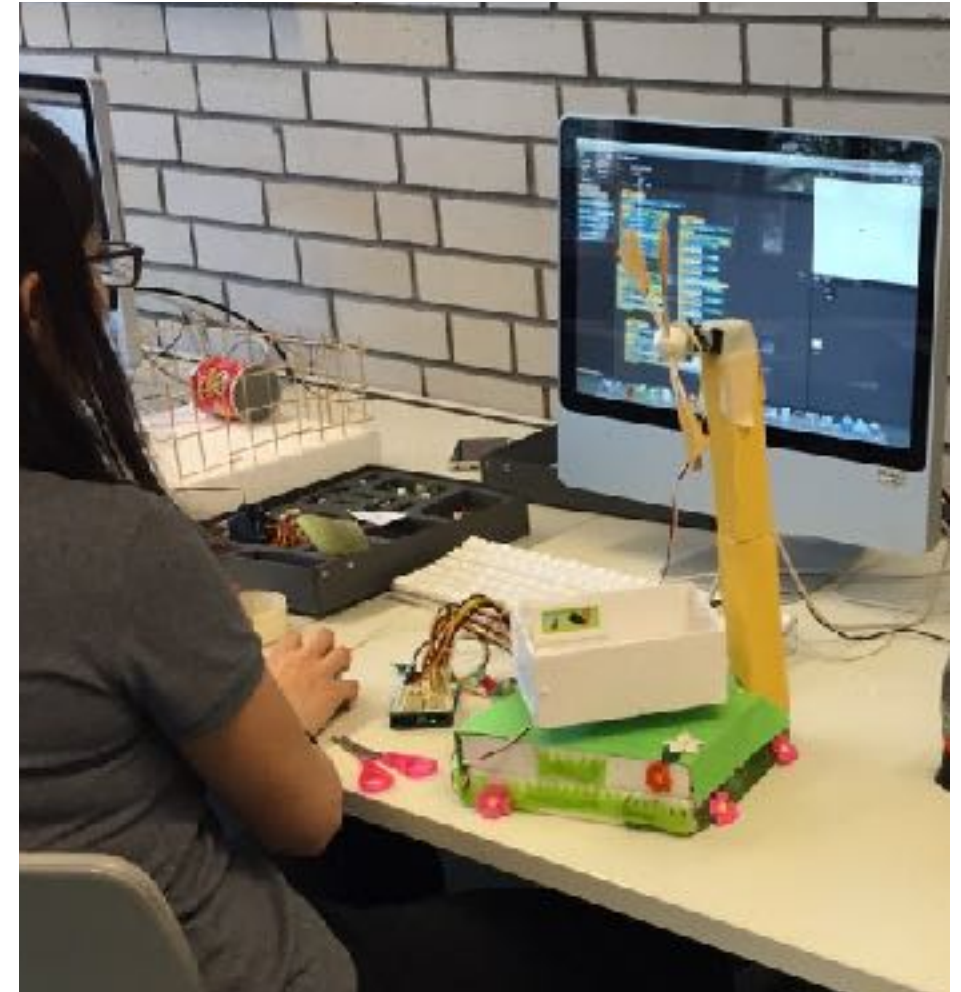
- Roboter im Unterricht sehr beliebt
- Ähnlich wie beim Physical Computing Verbindung von physischer und virtueller Welt mit Sensoren und Aktoren
- Hauptunterschied: Anwendungen und Gestaltungsprozesse
- Robotik fokussiert stärker auf Themen wie
 - physische Tätigkeiten
 - Orientierung und Bewegung in 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

Lerntheoretische Überlegungen: Konstruktivismus

- Begriff geprägt durch Seymour Papert, 1980er
- baut auf der Idee auf, dass Lernen als aktiver Konstruktionsprozess verstanden wird (Piaget, Konstruktivismus):
 - vernetzte Wissenstrukturen entstehen durch die Interpretation neuer Informationen in Relation zu bereits vorhandenem Wissen und basierend auf vorherigen Erfahrungen
 - Kreation bedeutsamer, sichtbarer und greifbarer Artefakte
 - Lernen aus Eigeninitiative und zu einem persönlich bedeutsamen Zweck

Papert emphasises:
“‘diving into’ situations rather than looking at them from a distance, that connectedness rather than separation, are powerful means of gaining understanding”

– Edith Ackermann.
“Piaget’s Constructivism, Papert’s Constructionism: What’s the difference?”.

Physical Computing

Lerntheoretische Überlegungen: Konstruktivismus

- Begriff geprägt durch Seymour Papert, 1980er
- baut auf der Idee auf, dass Lernen als aktiver Konstruktionsprozess verstanden wird (Piaget, Konstruktivismus):
 - vernetzte Wissenstrukturen entstehen durch die Interpretation neuer Informationen in Relation zu bereits vorhandenem Wissen und basierend auf vorherigen Erfahrungen
 - Kreation bedeutsamer, sichtbarer und greifbarer Artefakte
 - Lernen aus Eigeninitiative und zu einem persönlich bedeutsamen Zweck

“[Learning] happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe”

– Seymour Papert and Idit Harel.
“Situating Constructionism”.

Physical Computing

Effekte im Unterricht

- kontextualisiertes und handlungsorientiertes Lernen im Vordergrund
- Förderung von Motivation und Kreativität
- Gestaltungsorientierter Ansatz spricht breite Schülerschaft an
- Prägung eines positiven Bilds der Fachwissenschaft Informatik (weg vom “Nerd-Image”)

Physical Computing

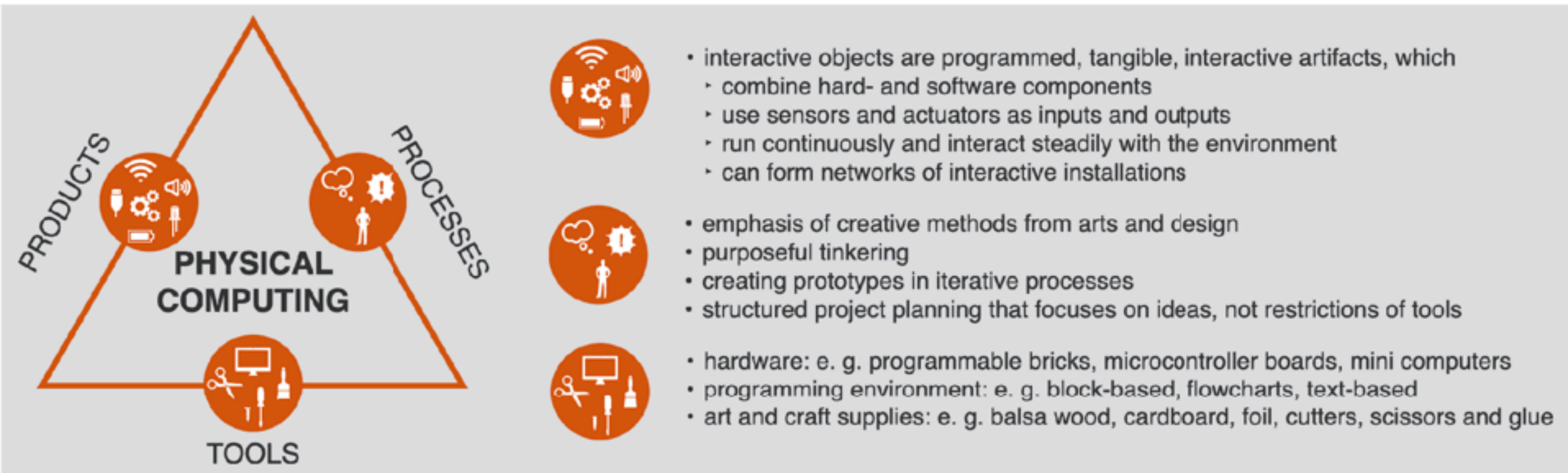
Bezüge im Modullehrplan Medien und Informatik des Lehrplan 21 (Zyklus 3)

Schülerinnen und Schüler ...

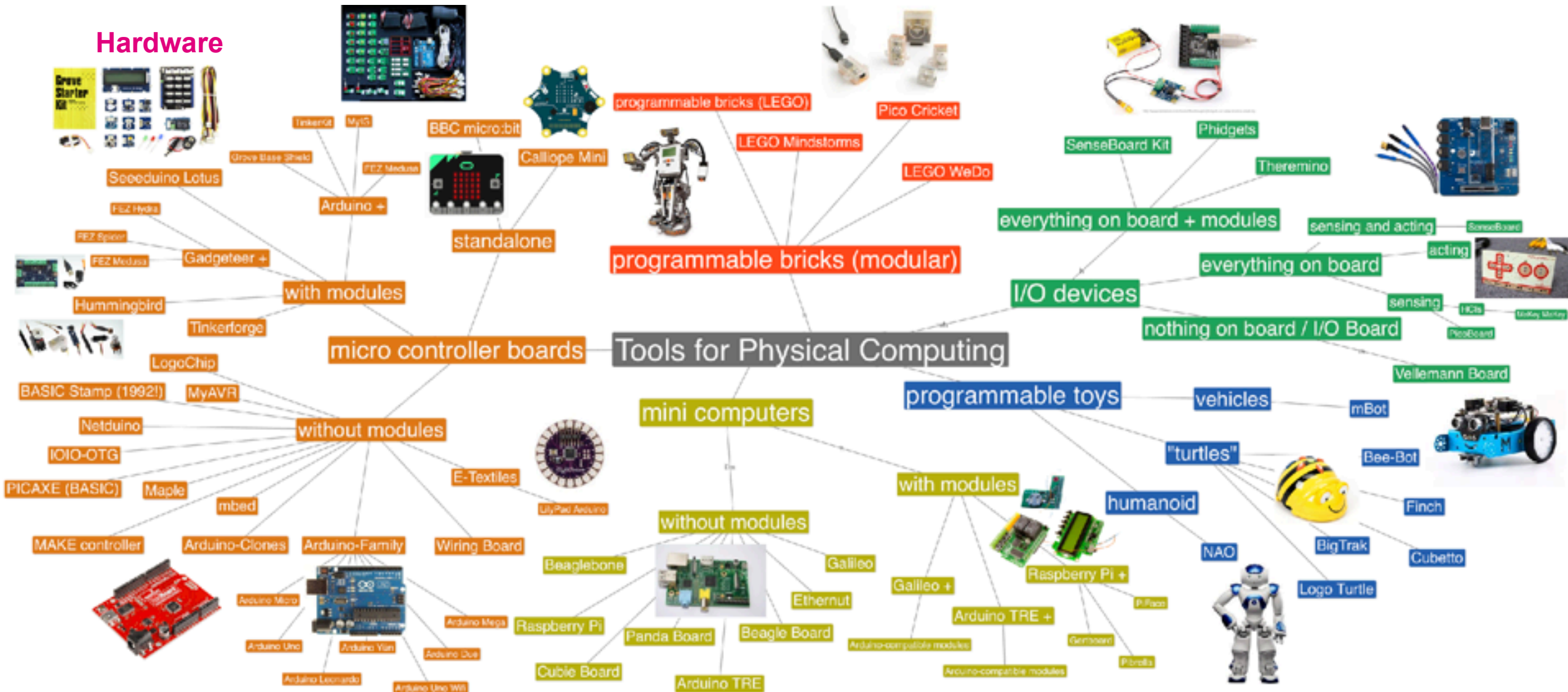
- können logische Operatoren verwenden (und, oder, nicht). *[Kompetenzbereich Datenstrukturen]*
- können selbstentdeckte Lösungswege für einfache Probleme in Form von lauffähigen und korrekten Computerprogrammen mit Schleifen, bedingten Anweisungen und Parametern formulieren. *[Kompetenzbereich Algorithmen]*
- 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

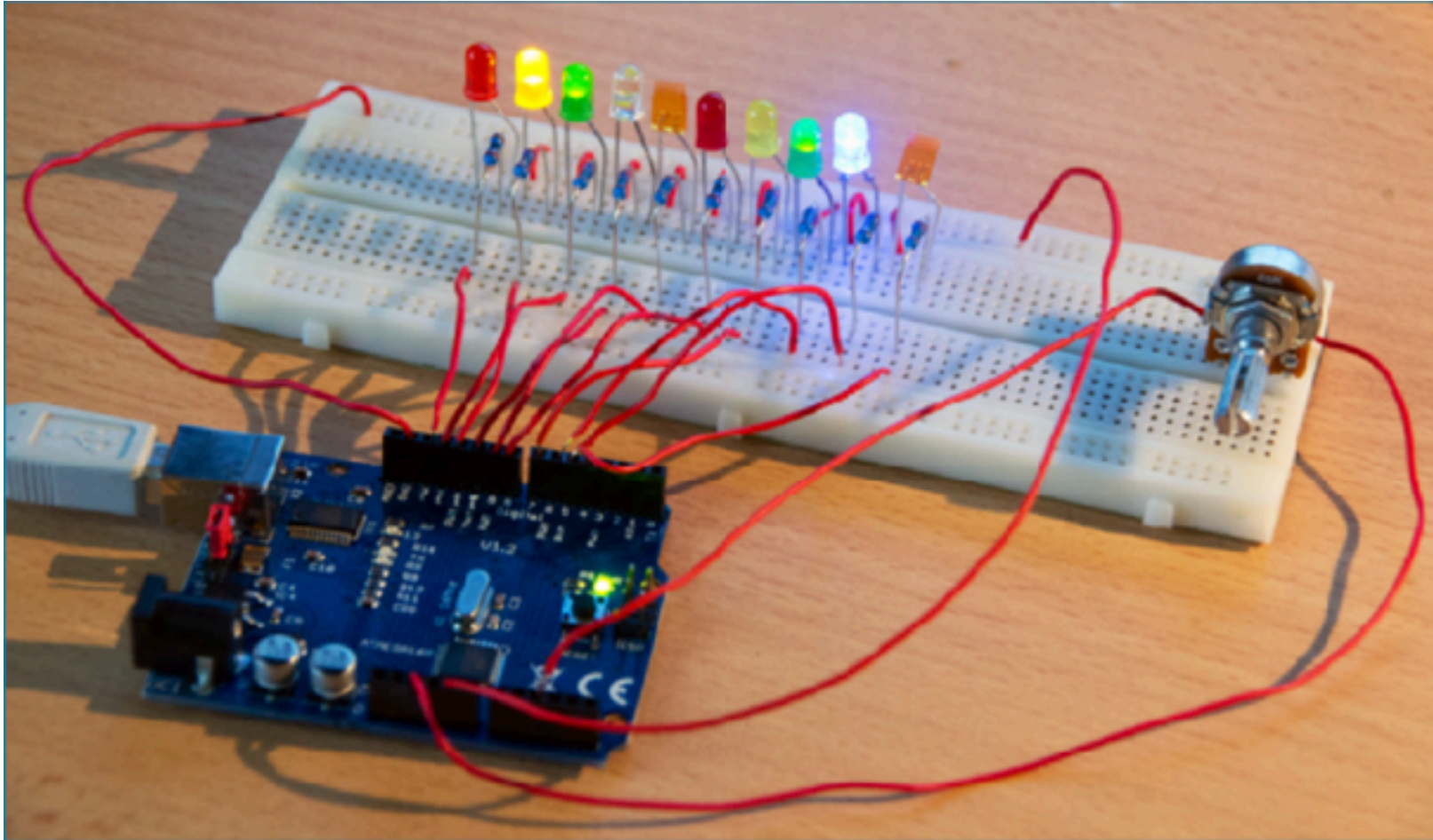


Unterrichtswerkzeuge



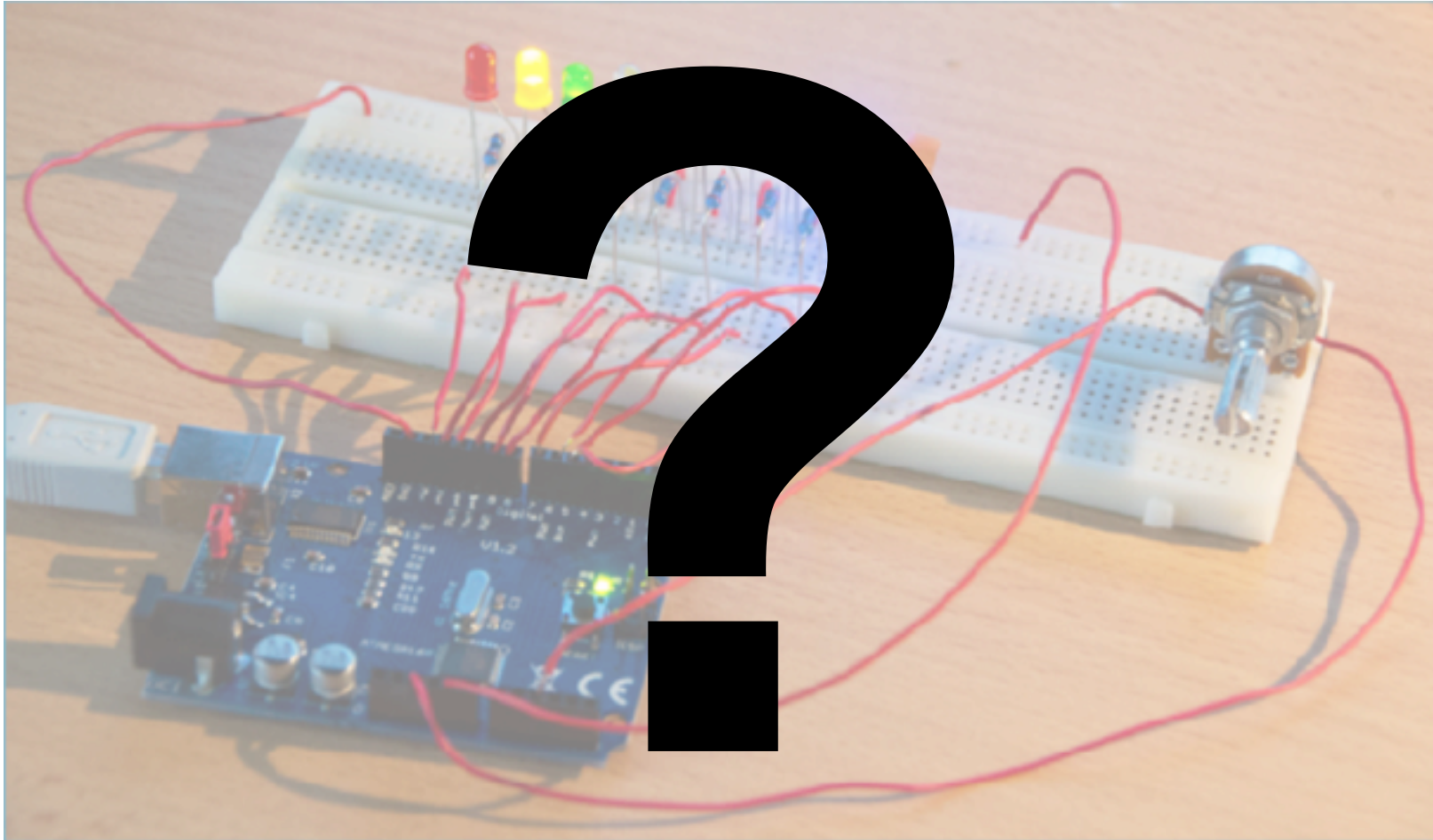
Unterrichtswerkzeuge

Arduino mit Steckbrett, Litzen und losen Bauteilen



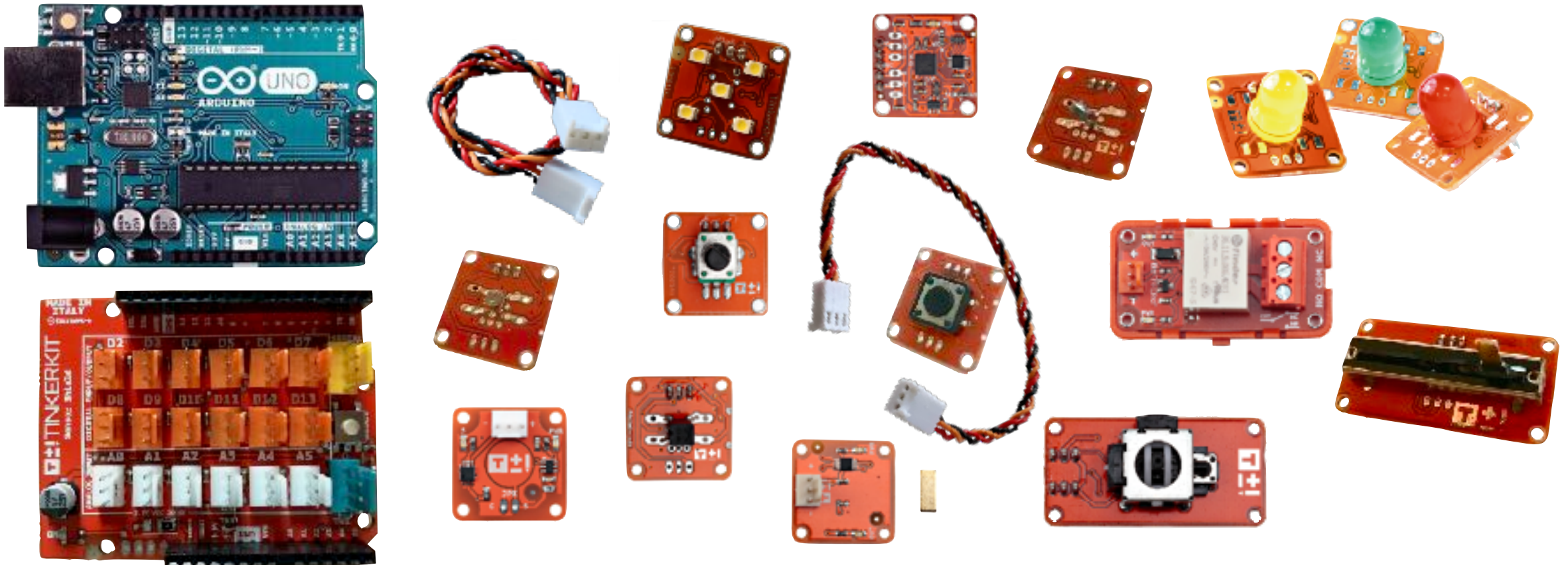
Unterrichtswerkzeuge

Arduino mit Steckbrett, Litzen und losen Bauteilen



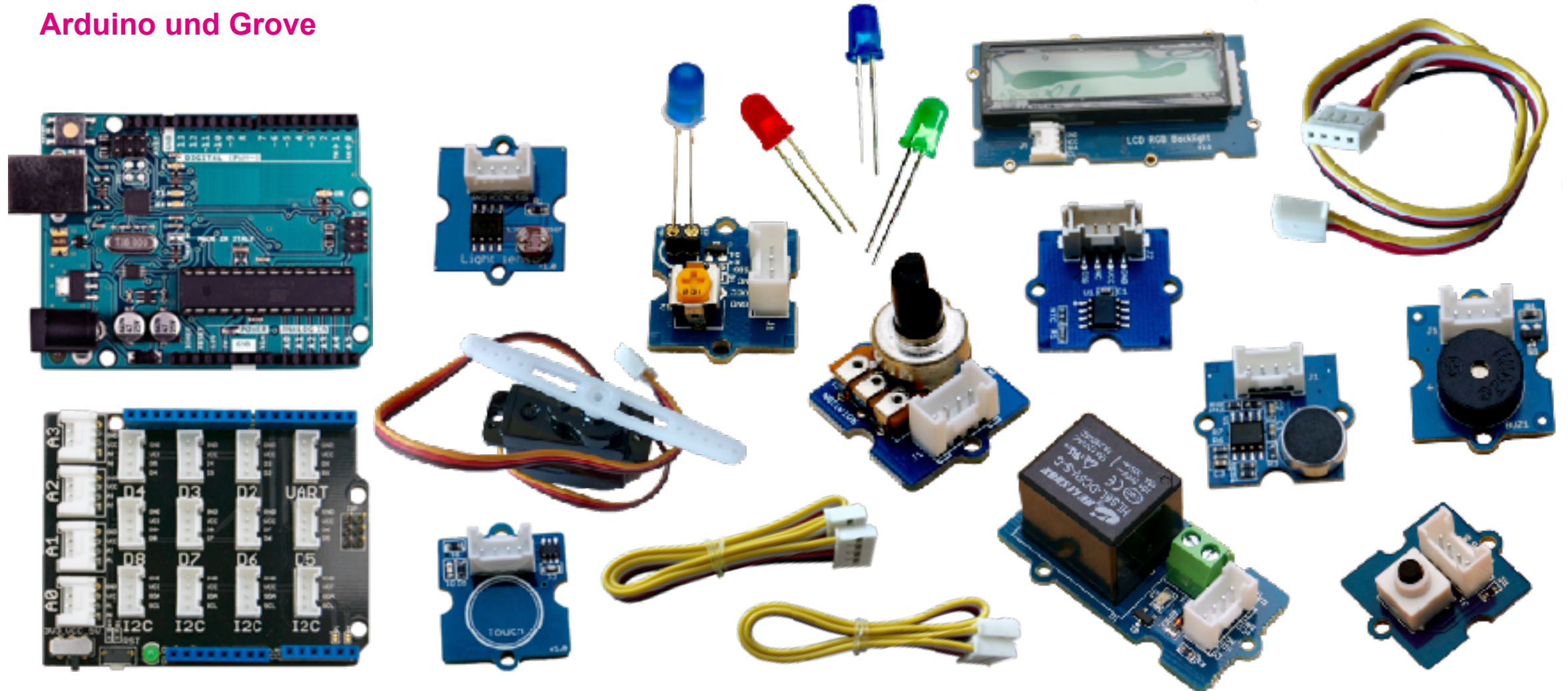
Unterrichtswerkzeuge

Arduino und TinkerKit



Unterrichtswerkzeuge

Arduino und Grove

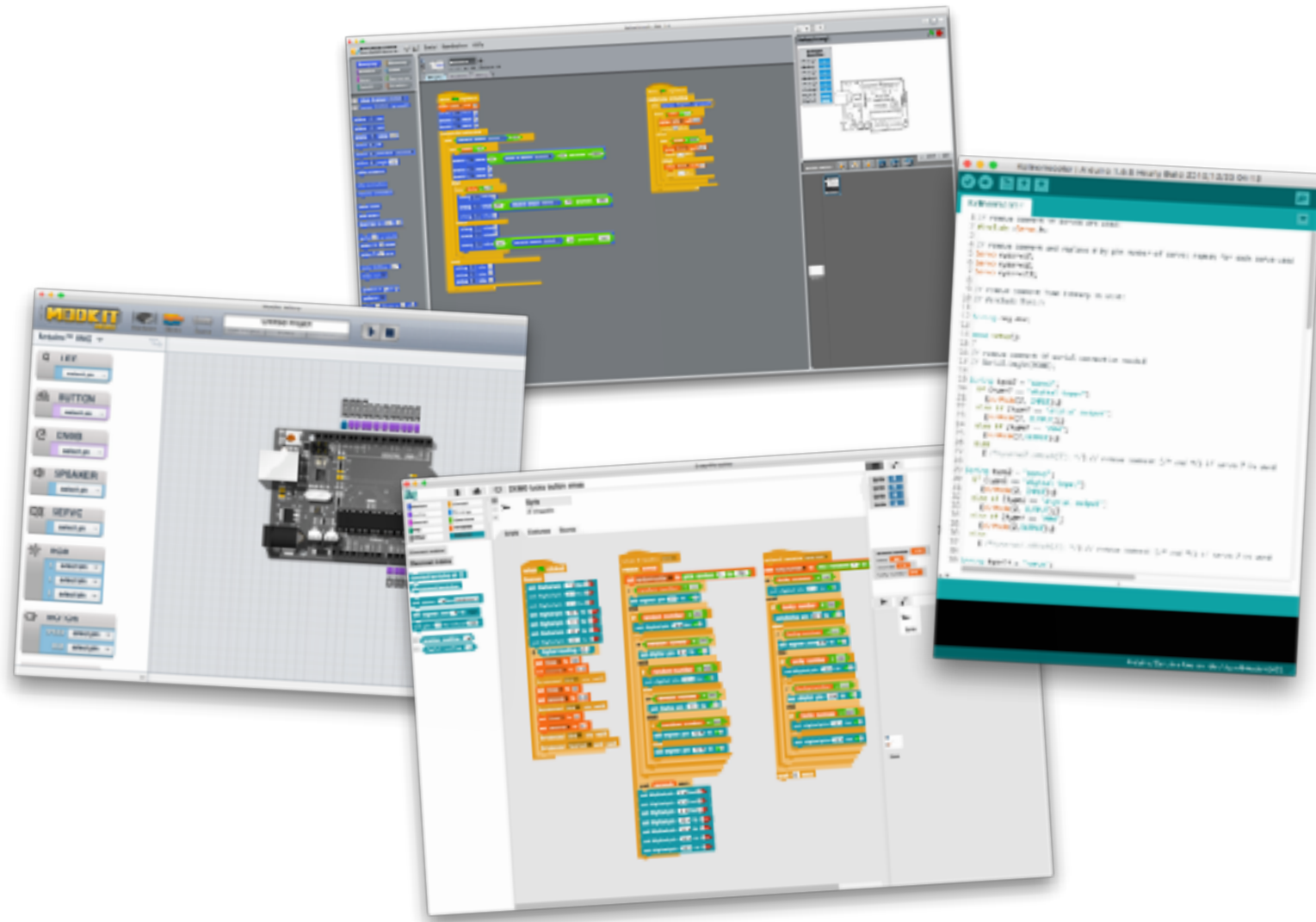


Unterrichtswerkzeuge

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)

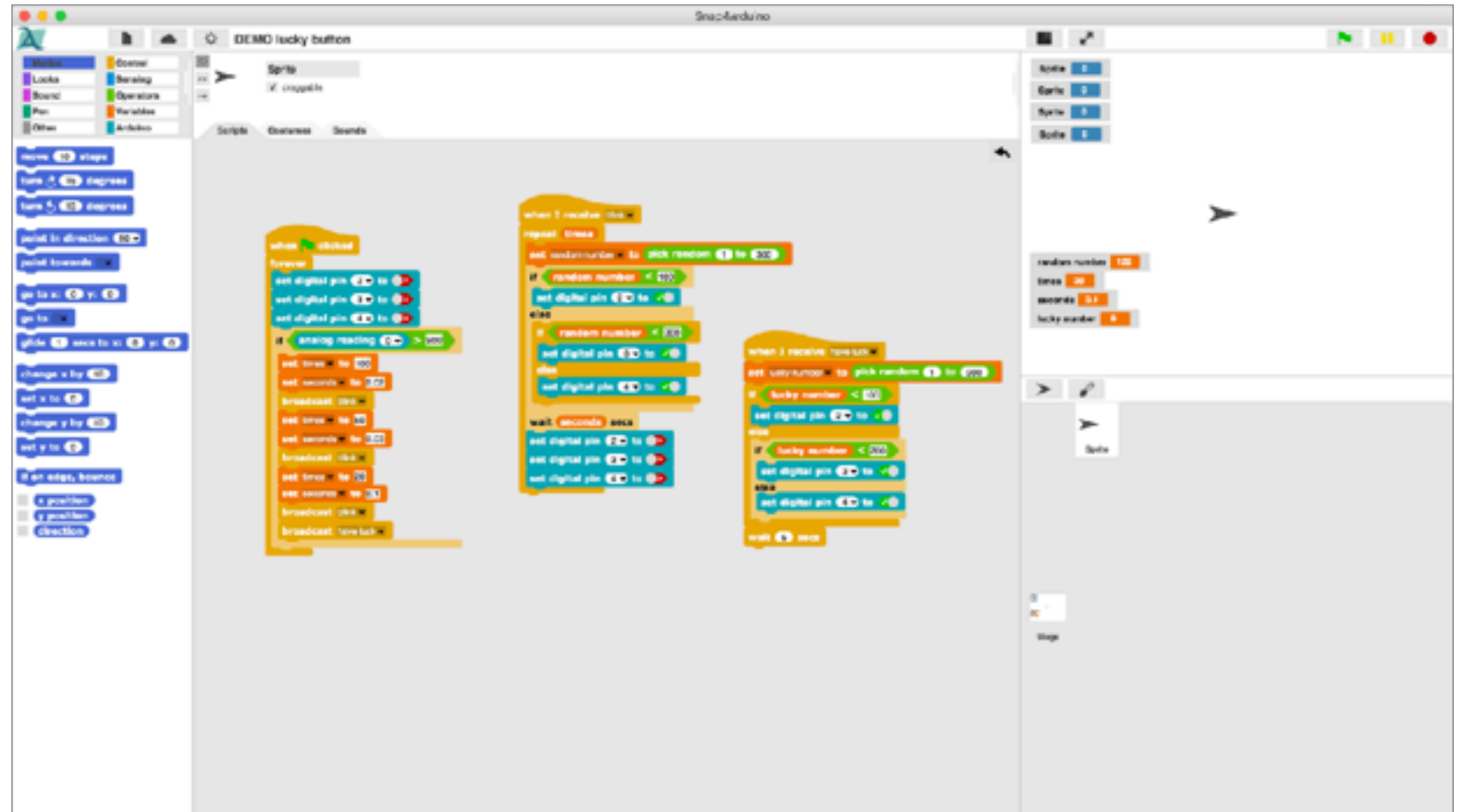
→ in Schule z. B. mit visuellen Umgebungen



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

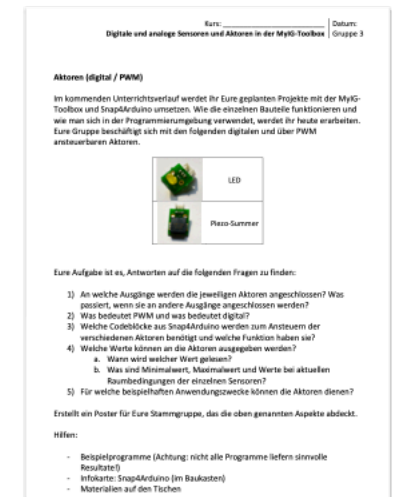
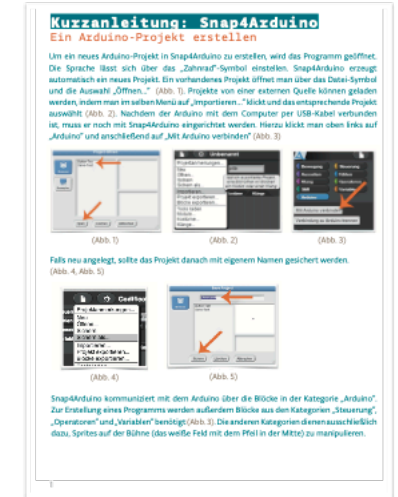
Gestaltungsprinzipien für Physical-Computing-Projekte

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

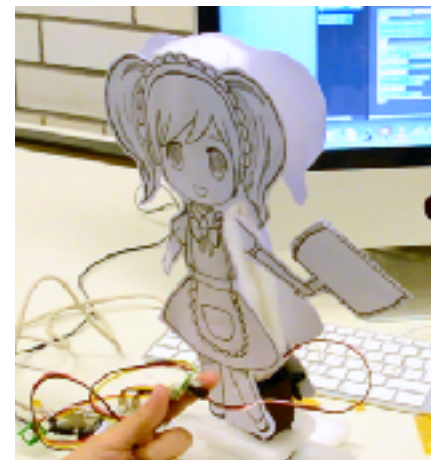
Anwendungsbeispiel "My Interactive Garden"

- 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.



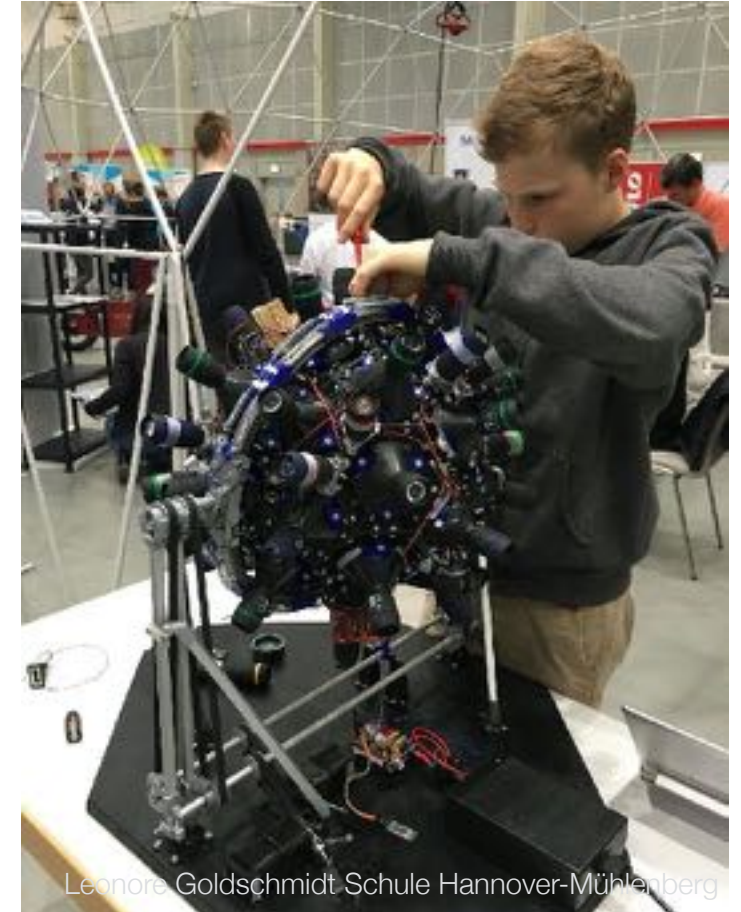
Unterrichtsumsetzung

Weitere Beispielprojekte



Unterrichtsumsetzung

Beispielprojekte



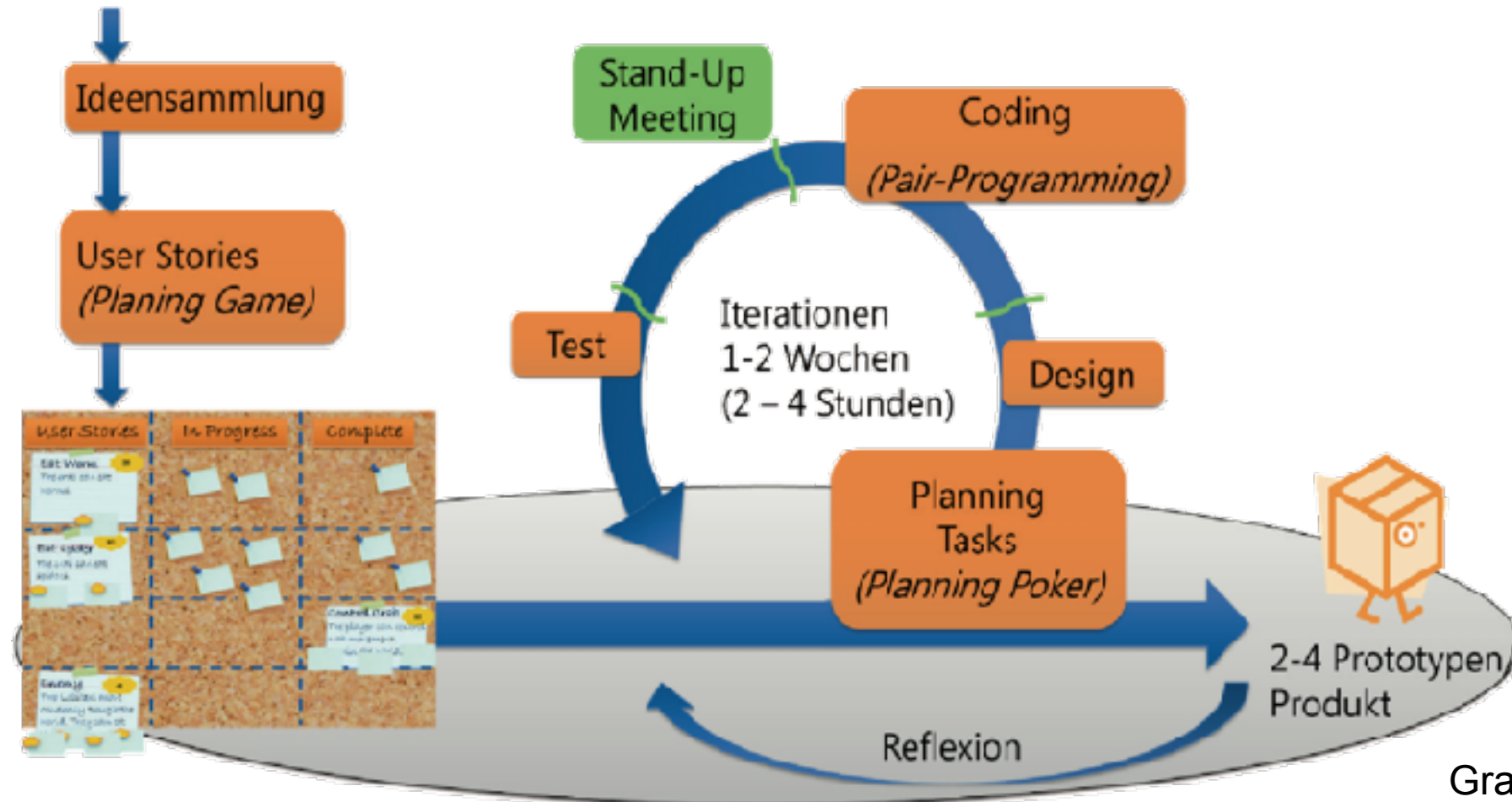
Praxis

- Installation der benötigten Software
- Kurzes Tutorial:
 - “Hello World” des Physical Computings - eine LED zum Blinken bringen
 - Auf einen Sensor reagieren
- Gruppenpuzzle bzw. Stationenlernen
- Miniprojekt mit Elementen Agiler Methoden der Softwareentwicklung: Wir planen eine Halloween-Party!

Happy Prototyping!

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

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