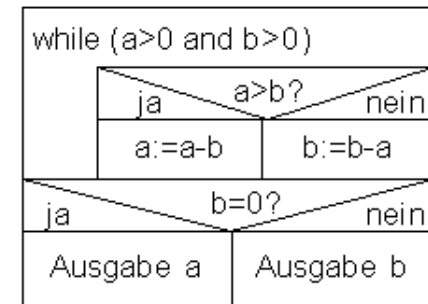
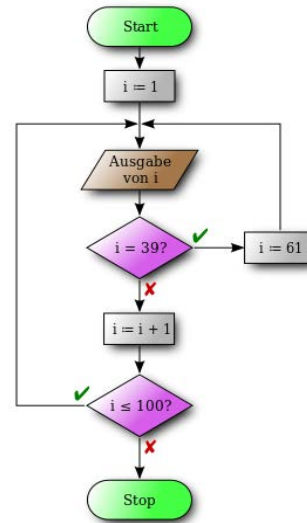


Der Calliope-Mini als vielseitiges Lernmedium

Kurz-Workshop



1. Zielsetzung

Studentische Programmier- und Digitalkenntnisse fördern

Der Einsatz von Calliope in der Lehramtsausbildung und in technisch-naturwissenschaftlichen Studiengängen

Beschreibung: Christian Hulsch arbeitet an der Professur Grundschuldidaktik Wirtschaft-Technik-Haushalt sowie an der Professur Montage- und Handhabungstechnik der TU Chemnitz und hat sich auf das **spielerische** Erlernen von Programmierkenntnissen mittels des Mikrocomputers "Calliope" spezialisiert. In diesem Shortcut gibt er einen Einblick in die **Verwendbarkeit und das breite Spektrum praktischer Kompetenzen**, welches im Umgang mit Calliope erlernbar sind und wie man diese für die eigene **Lehre und Forschung** in **anderen Fachbereichen** adaptieren könnte. Christian Hulsch geht ebenso auf die Möglichkeiten der Nutzung von Calliope in der **reinen digitalen Lehre** ein.

Wo: <https://webroom.hrz.tu-chemnitz.de/gl/jan-ayz-ywc>

Lernziele - Die Teilnehmenden ...

- kennen die Möglichkeiten des Mikrocomputers bzw. **Mikrocontrollers** "Calliope"
- reflektieren über die Vermittlung von **Systementwurfs-** und Programmierkenntnissen in verschiedenen Fachbereichen
- diskutieren Möglichkeiten der Übertragung auf den jeweils eigenen Fachbereich.

Spielerisch?

Spielerisch lernen - Glücksempfinden beim Programmieren?! ;-)

Vier Erscheinungsformen des **Glücks** in einem gegenseitigen
Kompensationsverhältnis:

- Arbeit (das Werk als Ziel)
- Interaktion (die Gesellschaft)
- Betrachtung (die Ästhetik des Erlebens)
- **Spiel** (die erfüllte Gegenwart)

Aber: Spiel ≠ einfach
Spiel = motivierter, intensiver, langsamer
Das Spiel muss stark angeleitet werden.
Ohne Kenntnis der Regeln kann man nicht (Schach)
spielen.

„Das **Spiel** kompensiert den Nachteil der **Arbeit**, im Vollbringen immer auf Ziele aus zu sein, mit deren Erreichen die Arbeit zum Erliegen kommt. Die **Betrachtung** kompensiert den Nachteil der **Interaktion**, in der Begegnung mit anderen stets auf die Antwort der anderen angewiesen zu sein. Die **Arbeit** kompensiert wiederum den Nachteil des **Spiels**, nichts Bleibendes hervorbringen zu können. Die **Interaktion** kompensiert den Nachteil der **Betrachtung**, kein verstehendes und respondierendes Gegenüber zu haben.“

Seel, Martin: Versuch über die Form des Glücks. Frankfurt/M. 1995 (→ Flow, Glück und Unglück liegen nahe beieinander)

2. Neues Gebiet, wir wollen Inhalte auswählen, aber nach welchen Kriterien?

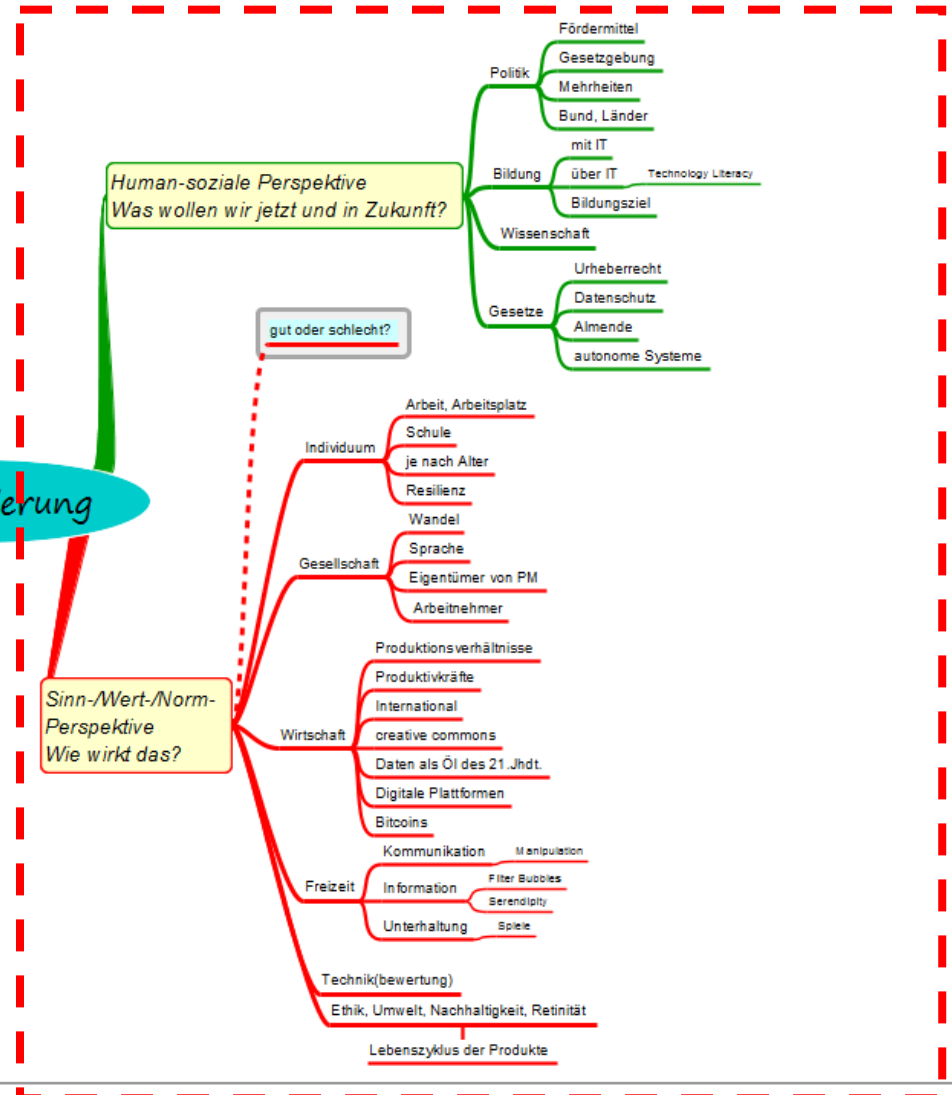
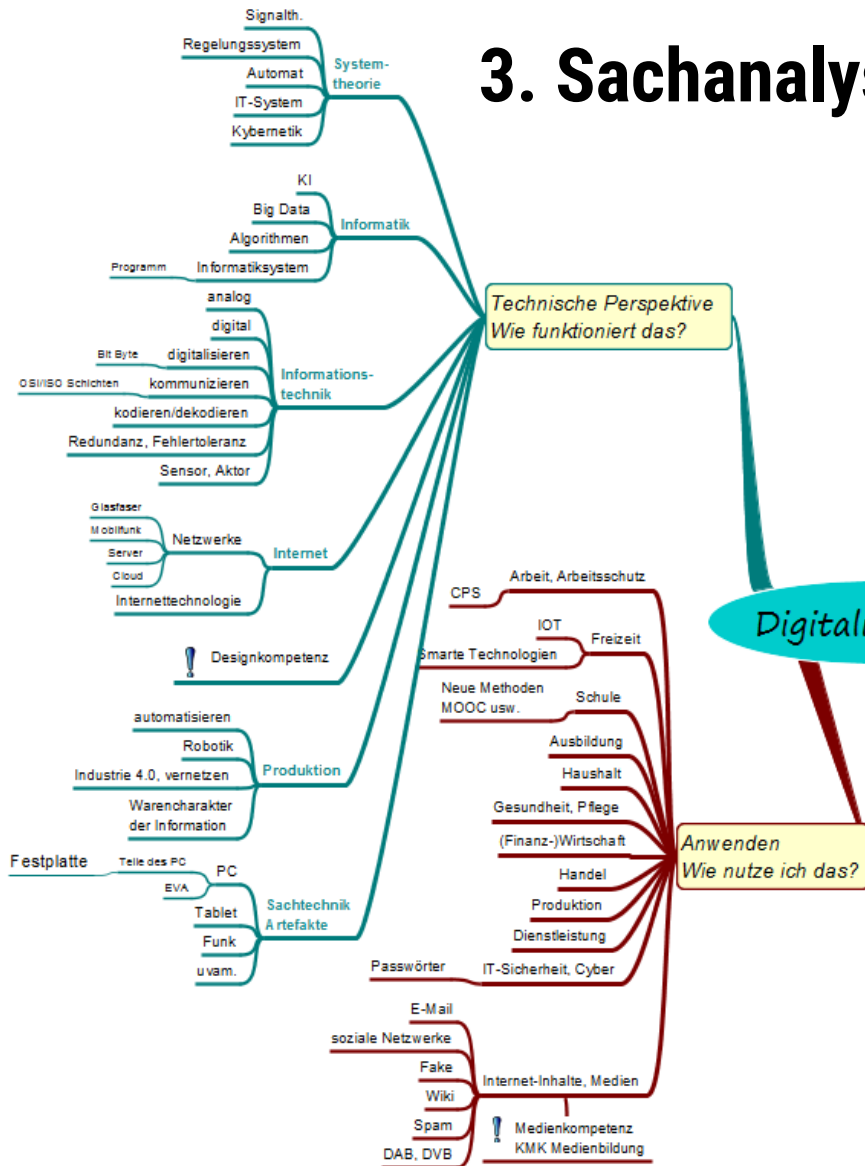
Die erste Frage ist immer, wer ist der Auftraggeber der Bildungsaufgabe und welche Maßgaben hat er explizit und implizit mitgegeben:

- Studienordnungen und Bildungs- oder Lehrpläne,
- gesellschaftlicher Auftrag der Bildungseinrichtung: Allgemeinbildung **oder** Berufspropädeutik **oder** berufliche Bildung **oder** Fortbildung

Der zweite Schritt ist die Sachanalyse nach einem geeigneten Ansatz.

Dann kommen Didaktik und Methodik ins Spiel und es wird richtig tricky ...

3. Sachanalyse zum Thema „Digitalisierung“



4. Fundamentale Ideen der Digitalisierung

„Eine fundamentale Idee (bzgl. einer Wissenschaft) ist ein Denk-, Handlungs-, Beschreibungs- oder Erklärungsschema“, das vier Kriterien erfüllt:

Vertikalkriterium, Horizontalkriterium, Zeitkriterium und Sinnkriterium. [vgl. 1993 Schwill, S. 20-31]

1993 Schwill, A.: Fundamentale Ideen der Informatik. Zentralblatt für Didaktik der Mathematik 1 (1993) 20-31 zitiert in Schwill: Ab wann kann man mit Kindern Informatik machen? Download von <https://pdfs.semanticscholar.org/f27d/50d19d2119c71ec6d341e33c3e85083b9f05.pdf>

Das Horizontalkriterium. Eine fundamentale Idee ist in **verschiedenen Bereichen** der Informatik vielfältig anwendbar oder erkennbar...

Das Vertikalkriterium. Eine fundamentale Idee kann auf jedem intellektuellen **Niveau** aufgezeigt und vermittelt werden.

Das Zeitkriterium. Eine fundamentale Idee ist in der historischen Entwicklung der Informatik deutlich wahrnehmbar und bleibt **längerfristig** relevant.

Das Sinnkriterium. Eine fundamentale Idee besitzt eine Verankerung im Alltagsdenken und eine lebensweltliche Bedeutung.

4. Fundamentale Ideen der Digitalisierung

Zur Auffindung fundamentaler Ideen schlägt [1993 Schwill, S. 10] folgenden Algorithmus vor:

„1. Schritt: Man **analysiert** konkrete Inhalte einer Wissenschaft und ermittelt Beziehungen und Analogien zwischen ihren Teilgebieten (wg. Horizontalkriterium) sowie zwischen unterschiedlichen intellektuellen Niveaus (wg. Vertikalkriterium). So erhält man eine erste Kollektion von fundamentalen Ideen.

2. Schritt: Diese Liste verbessert und modifiziert man, indem man nachprüft, ob jede der Ideen auch eine **lebensweltliche** Bedeutung besitzt und im Alltag nachweisbar ist (Sinnkriterium). **Hier muss man aber ehrlich zu sich selbst sein!**

3. Schritt: Anschließend versucht man die **historische** Entwicklung jeder Idee nachzuzeichnen. So gewinnt man evtl. weitere Gesichtspunkte und stabilisiert die Ideenkollektion.

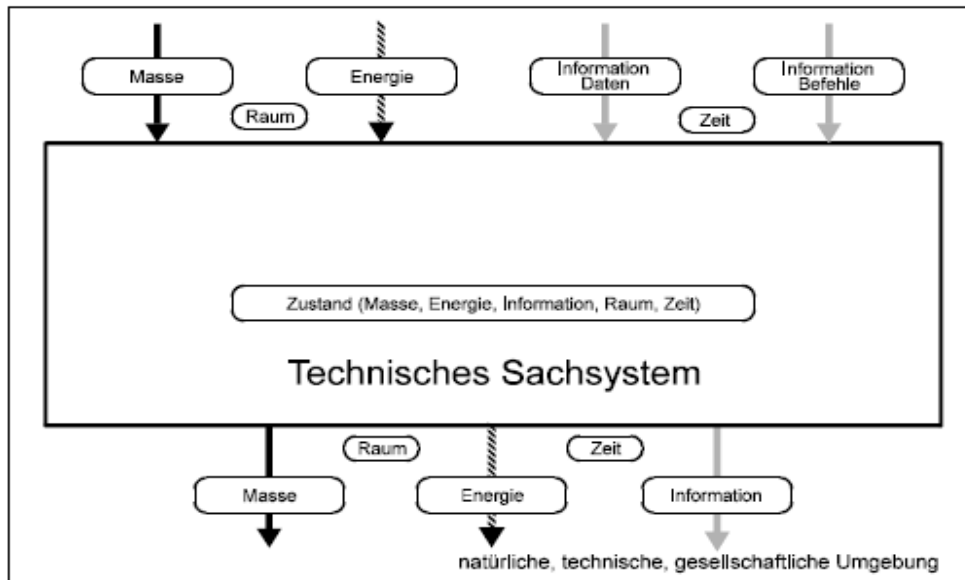
4. Schritt: Schließlich stimmt man die Ideen aufeinander ab und analysiert Beziehungen zwischen ihnen. (**Unabhängigkeit**)“

4. Anwendung der Fundamentalen Ideen auf „Digitalisierung“

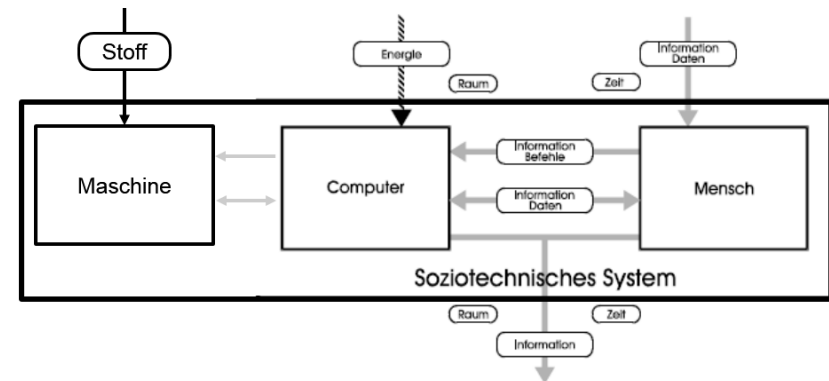
1. Analoge und digitale Signale, Signalquellen, Übertragung, Verarbeitung/Operationen und Signalsenken (Sensoren und Aktoren)
2. Steuerbare Prozesse beinhalten immer Daten, Strukturen, Algorithmen (Technik, Wirtschaft, Gesellschaft)
3. Analoge und digitale Baugruppen und Geräte, Aufbau, Wirkungsweise, Benutzung
4. Programmieren als Technik der Steuerung programmierbarer Geräte
5. Digitalisierung, ihre Technik und Bedeutung, **Bewertung, Auswirkungen**
6. Die automatisierte Fertigung: Industrie 4.0, IoT, (Big Data, KI, QR-Code, Kostenrechnung, VR, 3D, Blockchain, Nanotechnologie, Biotechnologie,) Robotik
7. Kommunikation als wechselseitige (reziproke) Übertragung von Bedeutung
8. (Internettechnologie)
9. Arbeitsschutz, Ergonomie, Umgang mit Technik

Das kann alles mit Calliope vermittelt werden!

5. Allgemeine Technologie, Technikdidaktik als Techniklehre und die Informationstechnik als Fachwissenschaft



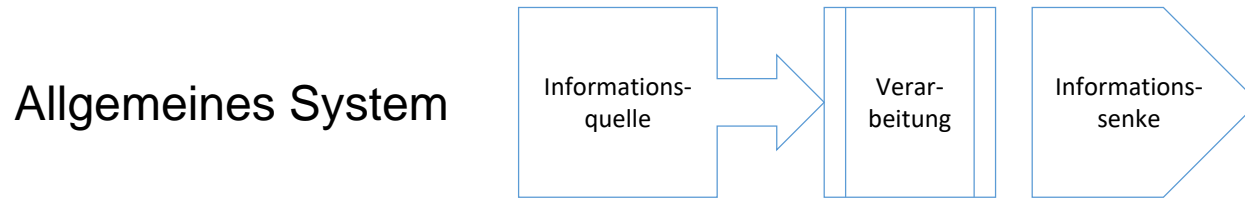
2009 Ropohl: Allgemeine
technologie, S. 120



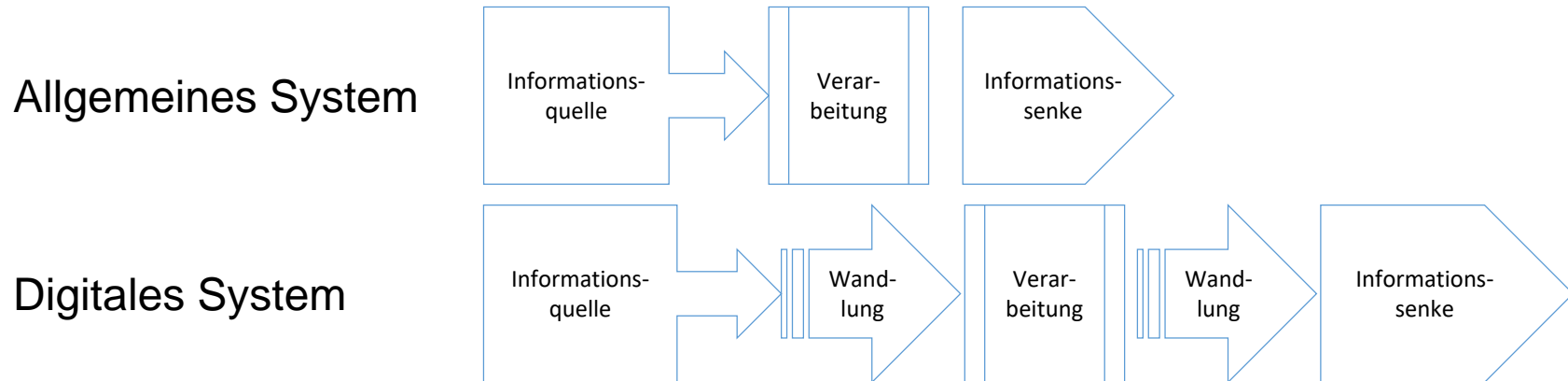
Abgewandelt nach 2009 Ropohl: AT, S. 58

Weder ET noch Informatik allein reichen aus, das Lernpotential zu erfassen
→ Trotz der „Krise der IT“ wieder Grundlagenwissenschaft Informationstechnik!

6. Sensor-Controller-Aktor: Die Keimzelle der IT

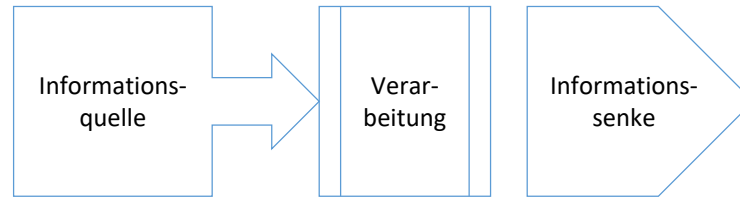


6. Sensor-Controller-Aktor: Die Keimzelle der IT

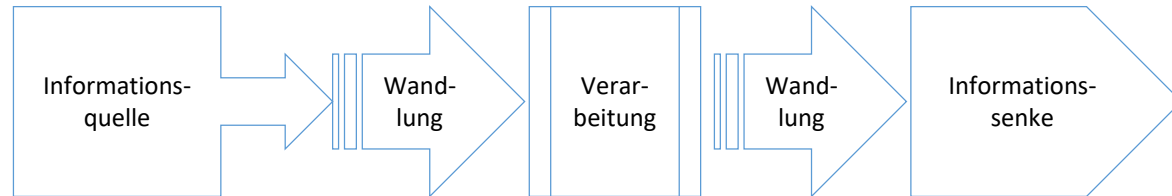


6. Sensor-Controller-Aktor: Die Keimzelle der IT

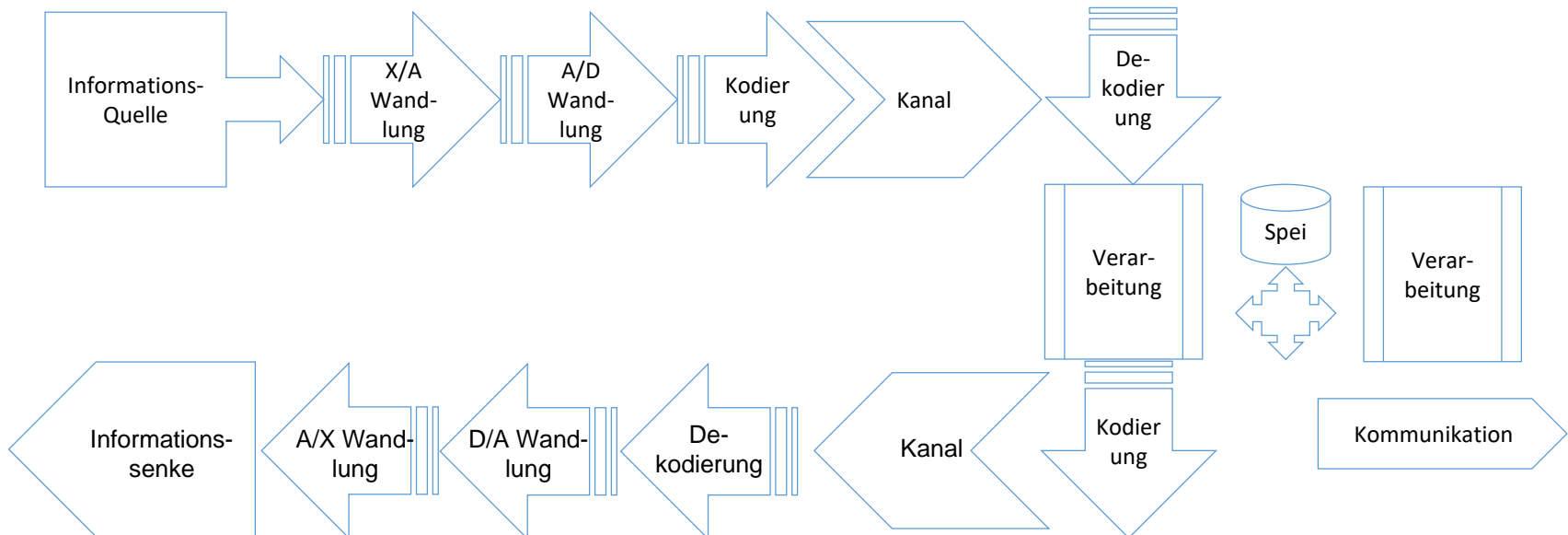
Allgemeines System



Digitales System



Im Detail:



6. Praxisbezug: Industrie 4.0

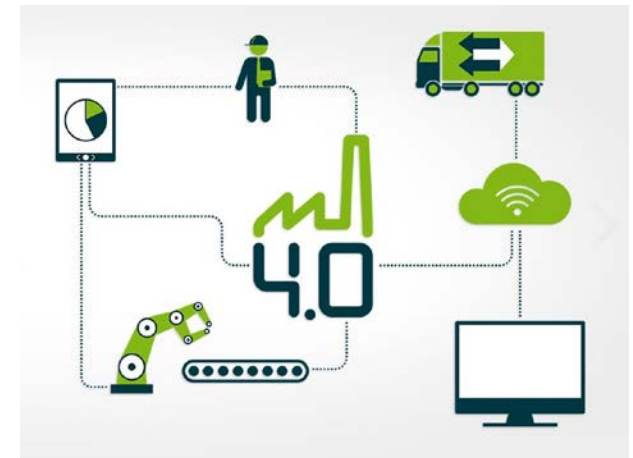
„Industrie 4.0“ beschreibt eine Form industrieller Wertschöpfung, die durch (weitgehende) Digitalisierung, Automatisierung sowie Vernetzung aller an der Wertschöpfung beteiligten Akteure charakterisiert ist und auf Prozesse, Produkte oder Geschäftsmodelle von Unternehmen einwirkt. [2019 Obermaier Handbuch I4.0, S. 7]

Industrie 4.0:

Zusammenführung der Produktion mittels Digitaltechnik auf Grundlage **intelligenter** und miteinander **vernetzter Systeme**, die sich **selbst** regeln und konfigurieren können.

Innerhalb der Produktion werden Maschinen, Anlagen, Produkte, Logistik und Menschen miteinander durch **Sensoren** und **Aktoren**, Elektrik und Elektronik, Informations- und Kommunikationsnetzwerke verbunden, so dass sie miteinander **kommunizieren** und agieren können (vgl. Yokogawa).

Anmerkung: Internet der Dinge (IoT) bereits als äquivalente Technik im Home-Bereich



www.yokogawa.com

7. Didaktisches Potential des Calliope



Didaktisch – Praktisch – Gut:

Ich meine, der Calli kann auch kleine Steuerungsaufgaben erledigen, andere Controller sind auch nicht genauer, er zeigt bei solchen Aufgaben ein bauarttypisches (Fehl-)Verhalten.

Sensor	Aktor
Temperatur	5x5 LED Display
Mikrofon	Farb-LED WS2812
Magnetfeld	Lautsprecher
Kompass	1-2 Motoren A/B
Lagesensor x/y	Relais (extern)
Beschleunigung x/y/z	Servomotor (extern)
Helligkeit	Grove-Anzeige (ext.)
Touchsensoren (Pins)	Neopixel-LEDs
Grove-Sensoren (ext.) (Ultraschall, IR, usw.)	Grove-Sender für IR, US, Fernbedienung
Funkwellenempfänger	Funkwellensender



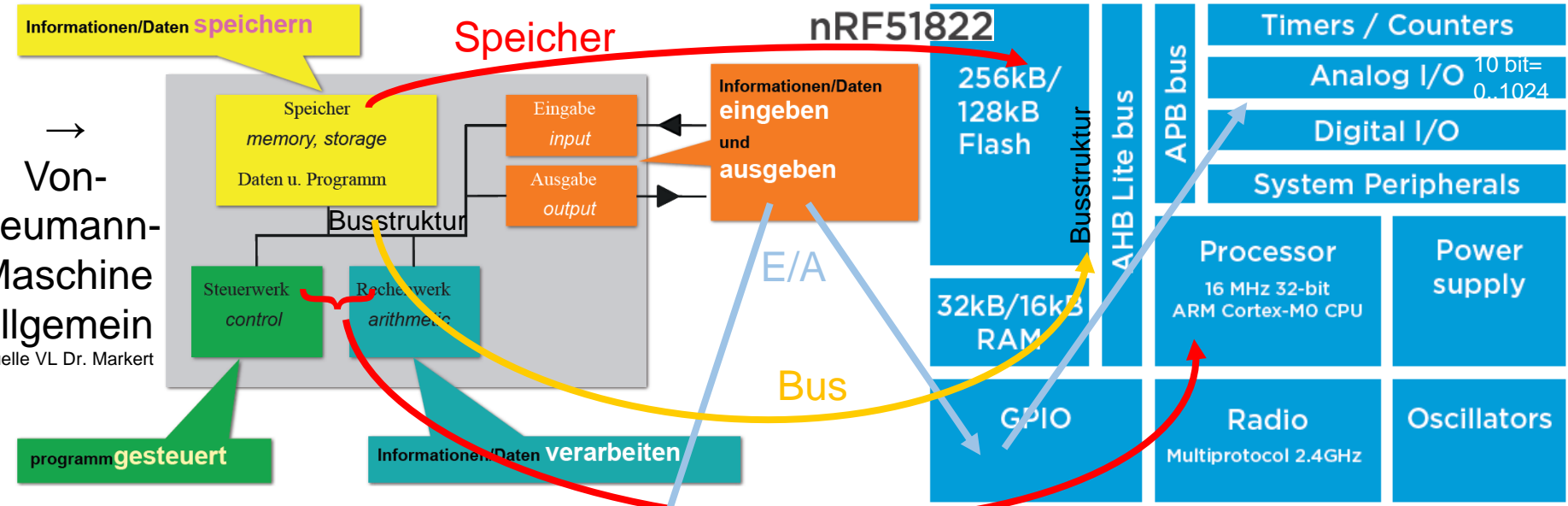
7. Didaktisches Potential des Calliope

Calliope als Lernprojekt – Was daran alles gelernt werden kann:

- Schaltzeichen, Schaltplan u.a. techn. Dokumentationen lesen und verstehen
- Fachsprache (Taster statt „Knopf“)
- Handwerkliche Fertigkeiten: Löten auf Universalleiterplatte
- Digitale und analoge Schaltungstheorie (Reihe/Parallel/Netzwerke)
- Zahlensysteme (binär-dezimal-hexadezimal)
- Elektronische Bauelemente, Sensortechnik, physikalische Grundlagen für messbare Größen (Anwesenheit, Licht, Schall, Gase, Staub, Bewegung) aus den Bereichen der Mechanik, Umwelt, Bewegung, Biologie usw.
- Umgang mit Messwerten: Mapping, Übermittlung, Verifikation, Verarbeitung, Präsentation
- Human Machine Interface

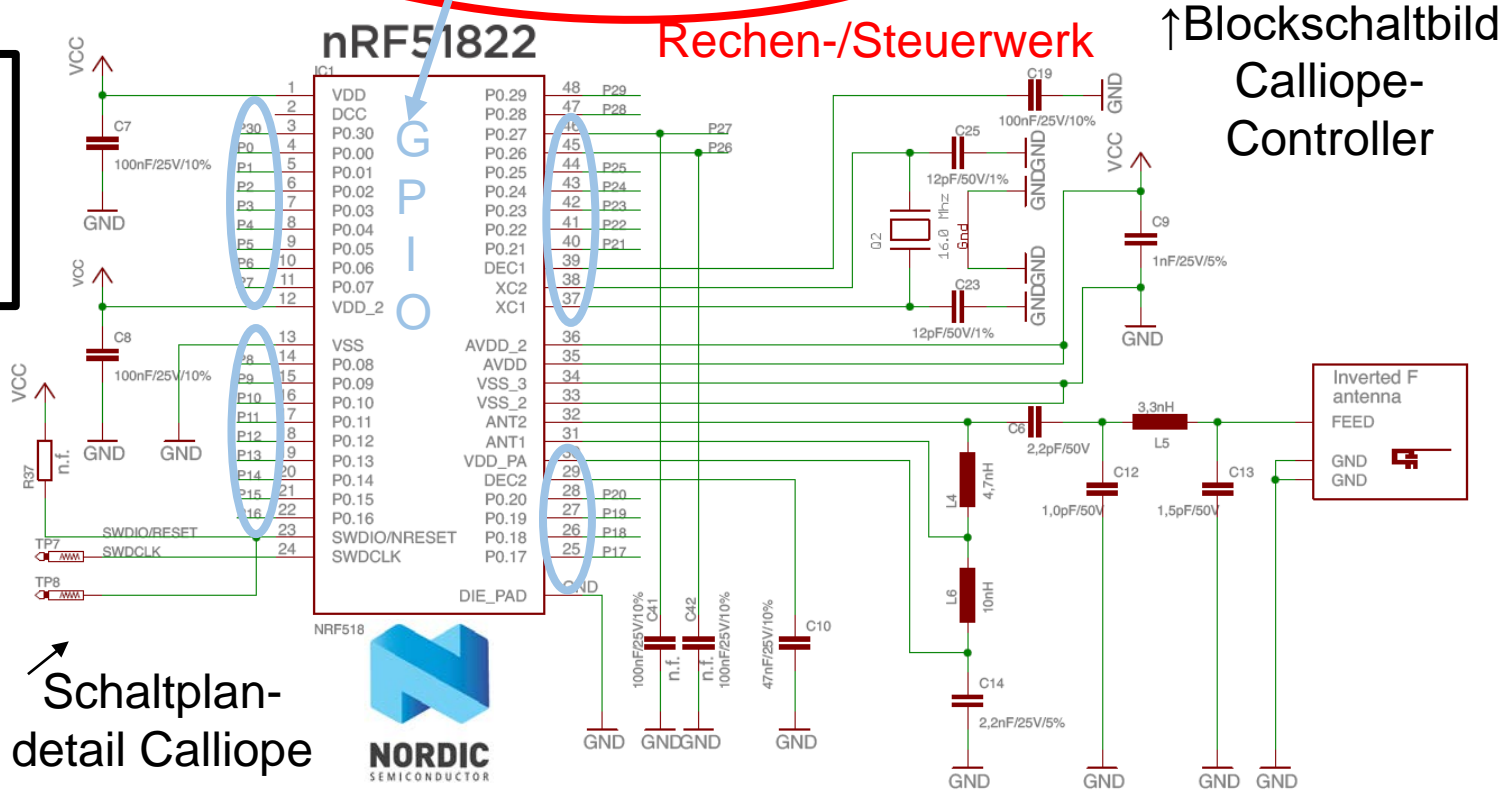
Geeignet für Studienrichtungen: ET/IT, (technische) Informatik, Physik, Maschinenbau, Sport/Sportgerätetechnik, Medizingeräte, Lehrerbildung, Chemie,

→ Von-Neumann-Maschine allgemein
Quelle VL Dr. Markert



Der Calliope ist ein Von-Neumann-Computer!

GPIO = General Purpose I/O = E/A = analoger oder digitaler Ein-/Ausgang



Schaltplan-detail Calliope



↑ Blockschaltbild Calliope-Controller

7. Beispiele für das technische Potential des Calliope

Elektronik

Bauelemente, Verwendung und Grundsaltungen: Widerstand (Bauformen, Farbcodes, Spannungsteiler, Stromteiler), Transistor (als Verstärker, Schalter, FET), Kondensator, Spule, Fotowiderstand, NTC, PTC, LED+Vorwiderstand, Siebensegment, ICs, OPV

Elektrotechnik

Strom-Spannung-Leistung-Verlustleistung, Gleichspannung, Wechselspannung, Konstanthaltung, Frequenz, Bauelemente, Verwendung und Grundsaltungen: Lautsprecher, Piezo, Motoren, Schrittmotor, Servomotor (Modellbau), „Dynamos“, Lüftermotoren, Relais (NO, NC), Taster (Taster entprellen), Schalter (Umschalter, Polwendeschalter), Elektromagnet, Messung von Strom und Spannung, Pulsweitenmodulation PWM

Technische Informatik

Controller

Namenskonventionen für Variablenbenennung (führender Buchstabe, Typkennzeichen usw.) Programmiertechniken, PAP, Struktogramm Datentypen und deren Darstellung im dig. System int, double, float ..., Tabellen, Matrizen Grundalgorithmen, Programmablaufsteuerung, rekursiv, sequentiell, parallel, Ablauf eines linearen Programms, IF-IF-IF=vollständige Alternative?, Interrupt, Semaphortechnik: bestes Beispiel Funk-crash-Melder (zwei identische Programme mit verschiedenen Aufgaben – wie macht man das?) Programmiersprachen Bedienoberfläche MakeCode gegen Nepo (eigentlich ist nur MakeCode geeignet!) Schnittstellen, PAP/Struktogramm, E/A: 1. Initialisieren der GPIO, was Ausgang, was Eingang, was digital/analog 2. Definieren und Vorbelegen der Variablen (malloc) 3. HMI-Programmierung: Falscheingaben abfangen, Falschausgaben abfangen 4. Debugging-Möglichkeiten unbedingt nutzen (Testpunkte setzen, Variablen beobachten) Bits und Bytes, Wiki erstellen, Apps

Sport

Bewegungsabläufe kontrollieren (Spaß an der Bewegung für Kinder)



8. Software-Engineering

Erlernen des Engineerings eines IT-Systems durch

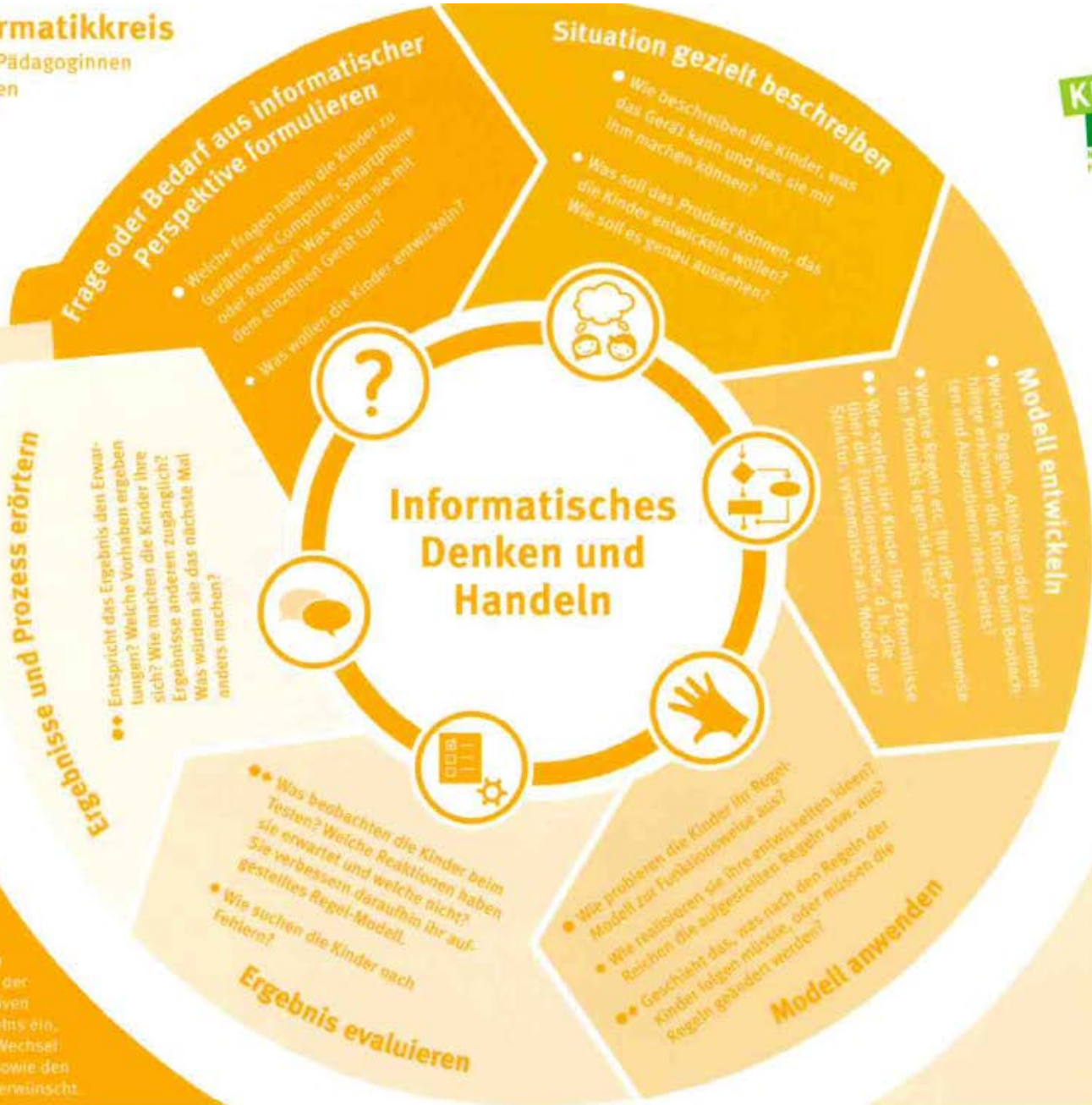
a) Dekonstruktion b) Rekonstruktion:

1. Problem verbalisieren, „Lastenheft“
2. Abstraktion (nur Wesentliches bleibt), dadurch Verringerung der Komplexität, Modellierung, Algorithmisierung, PAP, Struktogramm
3. Modell gedanklich durchspielen/vorevaluieren, Zeitregime simulieren
4. Strukturierung, Modularisierung
5. Modulweise Digitalisierung (Programmierung), Fehlersuche (Debugging)
6. Fehlertoleranz
7. Kompilierung
8. Evaluierung, Verifizierung
9. Dokumentation
10. Wiederverwenden, teilen mit anderen, Beachtung rechtlicher Aspekte bei Veröffentlichung



Der Informatikkreis

Hinweise für Pädagoginnen und Pädagogen



Neben dem grundlegenden Prozess des Erkundens (●) bestehender Informatiksysteme ist das Gestalten (◆) zentraler Bestandteil der Informatik. Nehmen Sie je nach Interesse oder Anliegen der Kinder eine dieser beiden Perspektiven informatischen Denkens und Handelns ein, um den Informatikkreis zu nutzen. Wechsel zwischen Erkunden und Gestalten sowie den einzelnen Phasen sind erlaubt und erwünscht.

9. Wie lernt man das Programmieren?

1. Grundkenntnisse Mikrocontroller und Peripherie
2. Allgemeine Strukturierung **aller** Programme kennenlernen:
 - Auswahanweisungen (if then else ...), Wiederholungsanweisungen, bool. Funktionen/Operatoren, Prozeduren, Funktionen, Rekursion, Klassen, Objekte, Vererbung, Rechte, Thread, Prozesssynchronisation/Sperren, Scheduling, Thread-Kommunikation, Deadlocks
 - Datentypen und typabh. Operatoren
 - Bibliotheken
3. Analyse → Denken wie ein Computer: Algorithmen formal beschreiben (PAP, Struktogramm)
4. Lexikalik (welche reservierten Worte gibt es), Syntax (wie zusammensetzen) und Semantik (was macht die Anweisung) einer **speziellen** Programmiersprache „spielend-forschend“ erlernen
5. Programmierumgebung kennenlernen
6. Programme schreiben (implementieren)
7. Testen, Debuggen
8. Dokumentieren

10. Editoren



MAKECODE

Mit Hilfe des MakeCode-Editors kann der Calliope mini vielfältig gesteuert werden. Entweder durch das Anordnen von Programmier-Blöcken oder mit JavaScript.

makecode.calliope.cc

Die aktuelle Betaversion (mit vielen Neuerungen und der Möglichkeit zwischen Blöcken, JavaScript und Python zu wechseln) findet sich unter: makecode.calliope.cc/beta

Eine offline Version findet sich unter: drive.google.com

The screenshot displays the Calliope mini programming environment. On the left, there is a category menu with the following items: Grundlagen, Eingabe, Musik, LED, Funk, Schleifen, Logik, Variablen, Mathematik, Motoren, Fortgeschritten, Funktionen, Arrays, Text, Spiel, Bilder, Pins, Seriell, Steuerung, and Erweiterungen. The workspace contains a 'beim Start' block followed by a 'dauerhaft' loop block. Inside the loop, there are two 'zeige LEDs' blocks, each followed by a 'pausiere (ms) 1000' block. The project name 'Flashing Heart' is visible at the bottom left, and a 'Herunterladen' button is at the bottom left. The top navigation bar includes 'CALLIOPE mini', 'Startseite', 'Teilen', 'Blöcke', 'JavaScript', and the Microsoft logo.

The screenshot displays the Calliope mini web interface. At the top, there is a navigation bar with 'CALLIOPE mini', 'Startseite', and 'Teilen' on the left, and 'Blöcke', 'JavaScript', and 'Microsoft' on the right. The main area is divided into three sections:

- Left Panel:** A large image of the Calliope mini board. Below it is an 'Explorer' dropdown menu and a grid of icons representing various components like sensors, actuators, and communication modules.
- Middle Panel:** A search bar 'Suche...' followed by a vertical list of categories: Grundlagen, Eingabe, Musik, LED, Funk, Schleifen, Logik, Variablen, Mathematik, Motoren, and Fortgeschritten.
- Right Panel:** A code editor showing JavaScript code for a 'Flashing Heart' project. The code is as follows:

```
1 basic.forever(function () {  
2   basic.showLeds(`  
3     # . # . #  
4     . . . . .  
5     # . . . #  
6     . . . . .  
7     # . # . #  
8   `)  
9   basic.pause(1000)  
10  basic.showLeds(`  
11    . . . . .  
12    . . . . .  
13    . . . . .  
14    . . . . .  
15    . . . . .  
16  `)  
17  basic.pause(1000)  
18 })  
19
```

At the bottom, there is a 'Herunterladen' button on the left, a 'Flashing Heart' label with a refresh icon in the center, and a set of control icons (undo, redo, stop, play) on the right.

JavaScript-Referenz

Webtechnologien für Entwickler > JavaScript > JavaScript-Referenz

Deutsch ▾

Auf dieser Seite

- Globale Objekte
- Anweisungen
- Ausdrücke und Operatoren
- Funktionen
- Zusätzliche Referenzseiten

Verwandte Themen

JavaScript

Tutorials:

- ▶ Einleitend
- ▶ JavaScript Guide
- ▶ Fortgeschritten
- ▶ Erweitert

Referenzen:

- ▶ Standardobjekte
- ▶ Ausdrücke & Operatoren
- ▶ Anweisungen & Deklarationen

Dieses Kapitel des JavaScript-Teils auf MDN dient als JavaScript-Sprachreferenz. Mehr über diese Referenz.

Globale Objekte

Dieses Kapitel dokumentiert alle JavaScript-Standardobjekte, zusammen mit ihren Methoden und Eigenschaften.

Anweisungen

Dieses Kapitel dokumentiert alle JavaScript-Anweisungen und -Deklarationen.

Für eine alphabetische Auflistung, siehe Sidebar auf der linken Seite.

Kontrollfluss

Block

Anweisungsblöcke werden genutzt um ein oder mehrere Anweisungen zu gruppieren. Der Block wird von geschweiften Klammern umfasst.

break

Beendet die aktuelle Schleife, Switch, oder mit Label versehene Anweisung und das

The screenshot displays the Calliope mini programming interface. On the left, there is a visual representation of the Calliope mini board with various components labeled. Below it, an 'Explorer' panel shows a list of categories: Grundlagen, Eingabe, Musik, LED, Funk, Schleifen, Logik, Variablen, Mathematik, Motoren, and Fortgeschritten. The main workspace shows a Python script for a 'Flashing Heart' project. The script is as follows:

```
1 def on_forever():
2     basic.show_leds("""
3         # . # . #
4         . . . . .
5         # . . . #
6         . . . . .
7         # . # . #
8         """,)
9     basic.pause(1000)
10    basic.show_leds("""
11        . . . . .
12        . . . . .
13        . . . . .
14        . . . . .
15        . . . . .
16        """,)
17    basic.pause(1000)
18    basic.forever(on_forever)
```

At the bottom of the interface, there is a 'Herunterladen' button and a 'Flashing Heart' project name with a refresh icon.

8.2. The `while` statement

The `while` statement is used for repeated execution as long as an expression is true:

```
while_stmt ::= "while" assignment_expression ":" suite  
            ["else" ":" suite]
```

This repeatedly tests the expression and, if it is true, executes the first suite; if the expression is false (which may be the first time it is tested) the suite of the `else` clause, if present, is executed and the loop terminates.

A `break` statement executed in the first suite terminates the loop without executing the `else` clause's suite. A `continue` statement executed in the first suite skips the rest of the suite and goes back to testing the expression.

8.3. The `for` statement

The `for` statement is used to iterate over the elements of a sequence (such as a string, tuple or list) or other iterable object:

```
for_stmt ::= "for" target_list "in" expression_list ":" suite  
           ["else" ":" suite]
```

The expression list is evaluated once; it should yield an iterable object. An iterator is created for the result of the `expression_list`. The suite is then executed once for each item provided by the iterator, in the order returned by the iterator. Each item in turn is assigned to the target list using the standard rules for assignments (see

10. Editoren



OPEN ROBERTA LAB®

Im „Open Roberta Lab“ lernen selbst Neulinge mit der grafischen Programmiersprache NEPO® intuitiv programmieren! Die Open-Source-Plattform beruht auf der langjährigen Erfahrung der Fraunhofer-Initiative „Roberta® – Lernen mit Robotern“.
lab.open-roberta.org

10. Editoren



CALLIOPE MINI SWIFT PLAYGROUND

Der Calliope mini Swift Playground kann ohne Vorkenntnisse eingesetzt werden und eignet sich hervorragend zum Einstieg ins textbasierte Programmieren mit dem **Mac** oder dem **iPad**. Die **Programmiersprache Swift** wird in den ersten Kapiteln durch die Eingabe und Anpassung von Parametern leicht verständlich nahegebracht. Zum Calliope mini Playground gibt es auch eine großartige Handreichung, welche dabei hilft spannende Projekte (mit Swift) im Unterricht umzusetzen.

[Calliope mini Swift Playground](http://www.calliope.cc)

10. Editoren



ABBOZZA! CALLIOPE

abbozza! Calliope ist eine grafische Entwicklungsumgebung für den Calliope mini, die für die Verwendung in Schulen und für Maker aller Altersstufen entwickelt wird. abbozza! wird am Lehrstuhl Didaktik der Informatik der Uni Osnabrück entwickelt.

[abbozza! Calliope](https://www.abbozza.cc)

10. Editoren



TIGERJYTHON 4 KIDS

Eine exemplarische Einführung ins Programmieren mit Python für Junge und Junggebliebene. Der Lehrgang ist in Anlehnung an die Programmiersprache LOGO spielerisch aufgebaut und wendet sich an ProgrammieranfängerInnen ohne informatische Vorkenntnisse, außer etwas Übung auf dem Niveau einer einfachen Textverarbeitung.

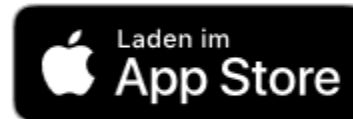
[TigerJython4Kids](#)

10. Editoren



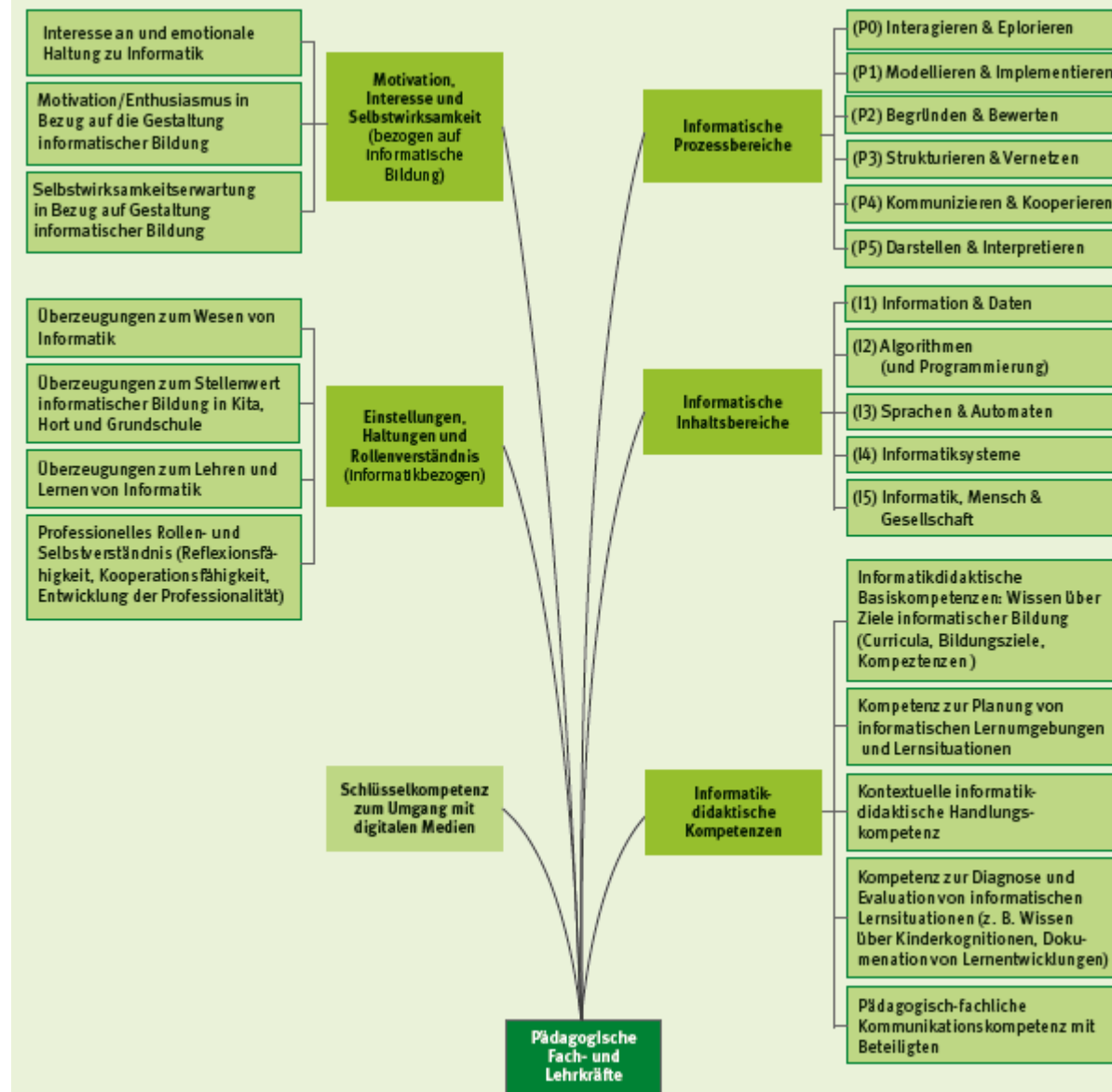
CALLIOPE MINI APP

Der Calliope mini kann auch mobil programmiert werden! Mit der iOS oder Android App können die Editoren direkt vom Smartphone oder Tablet genutzt werden.

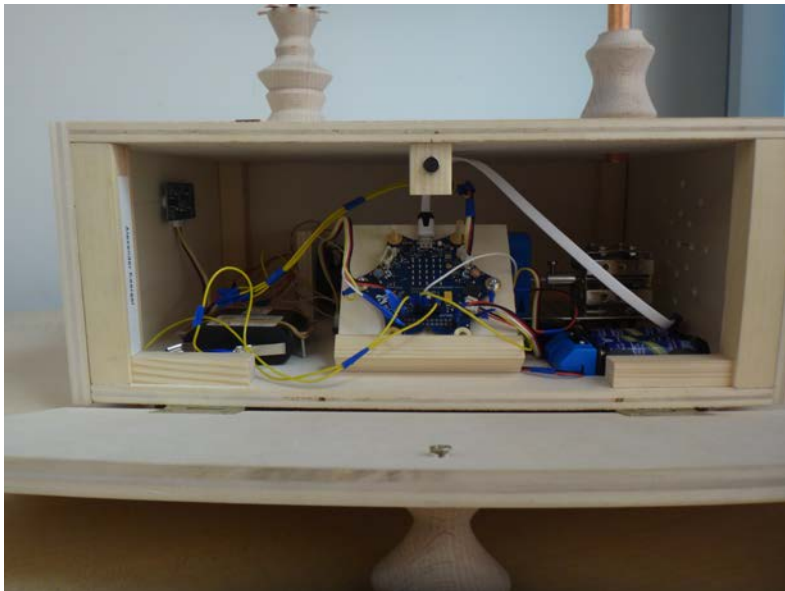


11. Ansatz in der Lehrer- ausbildung

Zieldimensionen informatischer
Bildung auf Ebene LuL
(HdkF Bd. 9, Bergner et al.
Frühe informatische Bildung,
S. 344)

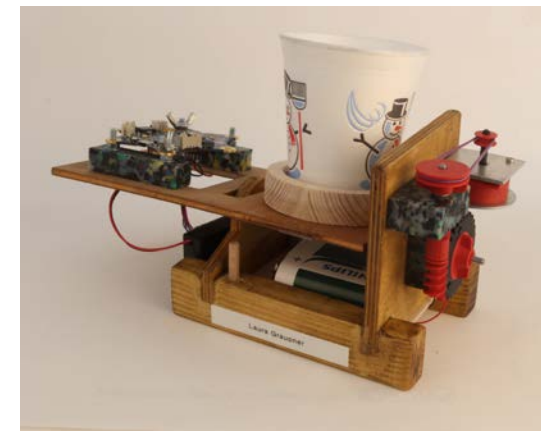
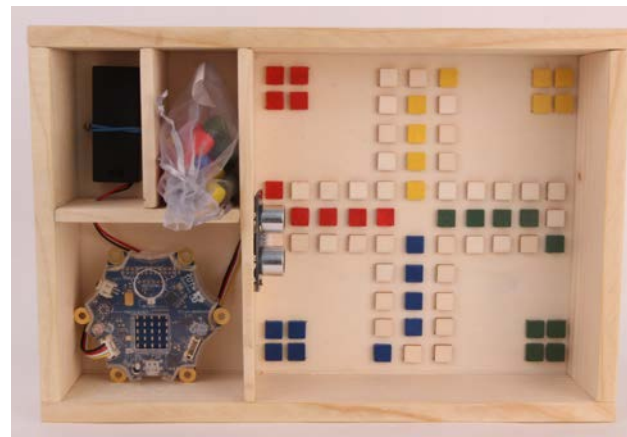
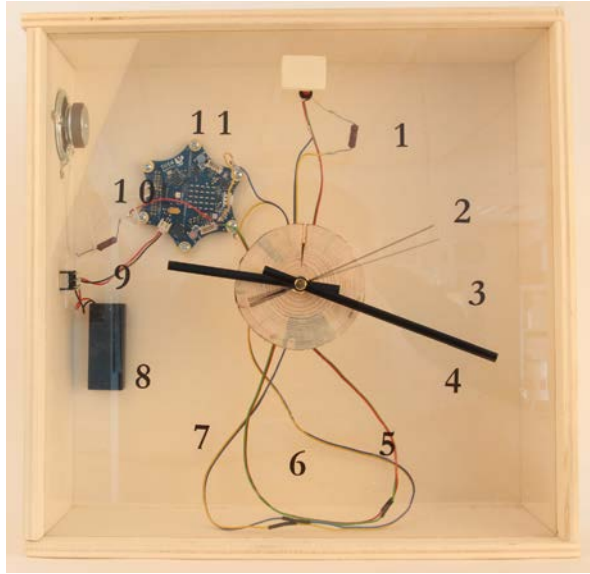


11. Ergebnisse der Projektarbeit in der Werklehrausbildung





11. Ergebnisse der Projektstätigkeit in der Werklehrausbildung





11. Praktische Erfahrungen

Es gab „kleine“ Probleme der letzten Übungen ...

- Akustische Rückkopplung, Einfluss Sonnenlicht
- Ausgänge sind nicht wie in Dokumentation; reichen nicht aus
- Ungelöste elektrische Probleme wurden hinausgeschoben (Licht-Spannungsteiler)
- Ungelöste mechanische Probleme
- Ausschalter nach außen führen, Ausschalter anbringen
- Wie können Bedienung und Rückmeldung mit zwei Tasten und zwei Leuchtdioden und/oder Ton sicher realisiert werden?
- Probleme beim Wechsel von USB auf Batterie
- Motorschnittstelle verhält sich undefiniert

und große strukturelle Probleme.

Designkompetenz:

- Wie werden komplexe Aufgabenstellungen Schritt für Schritt gelöst?
- Mangelhafte handwerkliche Fähigkeiten: Kiste mit Deckel bauen, Türen
- Fehlende Kenntnisse aus den Kursen des Masch.baus und der Elektrotechnik machen sich stark bemerkbar

Digitalkompetenz:

- Bedienung des Editors Makecode nicht intuitiv, die Möglichkeiten zum Up-/Download des eigenen Programms wurde von vielen nicht erkannt. Hochladen der Dokumentation in das Netz ?
- Studierende haben Verständnisprobleme, wie ein Programm im Controller (Calliope) abläuft
- Die Grundlagen des strukturierten Programmierens sind unbekannt, ebenso Automatenzustandsdiagramm
- Mehrere Studierende waren nicht in der Lage über Snipping-Tool oder Drucktaste das Programm abzufotografieren (Digital-Na[t]iv?)



11. Praktische Erfahrungen

Erkenntnisse aus den Fragebogen zum Calliope-Projekt

- LuL können sich den Einsatz von Calliope in der GS mehrheitlich nicht vorstellen weil sie selbst nicht fit genug sind/ sie Calliope für ungeeignet halten; Lego WeDo sei besser
- LuL sehen die Notwendigkeit einer solchen Vorbereitung auf Digitalisierung überwiegend nicht ein
- LuL haben keinen Spaß am Programmieren da notwendiges Wissen fehlt, sie sind bis auf wenige Ausnahmen *nicht Flow-fähig*. Calliope ist ihnen teilweise zu schwierig.
- LuL wollen vor der Arbeit mit Calliope grundlegend programmieren lernen und mit den Editoren beschäftigt werden (wahrscheinlich kinofilmartig)
- Notwendige Lernstunden zählen sie mit zu den 10 AS für das Prüfungswerkstück (die meisten Programme waren unter 20 Zeilen!)
- LuL können mehrheitlich mit offenen Aufgaben schwer umgehen (haben sie in der Schule nicht gelernt); ein Werkstück selbst zu konstruieren ist einigen zu schwierig
- LuL wünschen sich mehr grundschulfertige Übungen
- Das 4K-Modell des Lernens im 21. Jhdt. (Kommunikation, Kollaboration, Kreativität, kritisch Denken) ist unbekannt

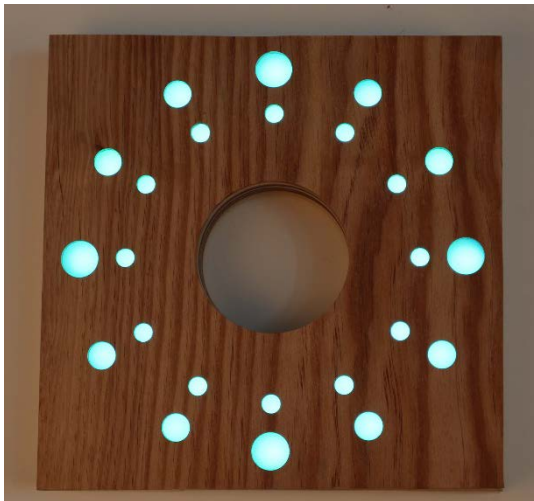
Aber: Mein subjektiver Eindruck: 90% waren am Ende zufrieden.



12. Ausblick

Anspruchsvolle Programmierprojekte mit Calliope

- Uhr (mit Weckfunktion), Zeitdarstellung!
- Drehzahlmesser mit Magnet
- Gymnastikprogramm



```
dauerhaft
wenn [digitale Werte von Pin P0] = 1
dann
  ändere [Taster 1] auf 1
  zeige LEDs
  pausiere (ms) 50
```

```
beim Start
ändere [Taster 1] auf 0
ändere [Taster 2] auf 0
ändere [Taster 3] auf 0
```

```
dauerhaft
wenn [digitale Werte von Pin P1] = 1
dann
  ändere [Taster 2] auf 1
  zeige LEDs
  pausiere (ms) 50
```

```
dauerhaft
wenn [Taster 1] = 1 und [Taster 2] = 1 und [Taster 3] = 1
dann
  Beginne Melodie [Dadadum] Wiederhole einmal
  pausiere (ms) 5000
  ändere [Taster 1] auf 0
  ändere [Taster 2] auf 0
  ändere [Taster 3] auf 0
  zeige LEDs
```

```
dauerhaft
wenn [digitale Werte von Pin P2] = 1
dann
  ändere [Taster 3] auf 1
  zeige LEDs
  pausiere (ms) 50
```

Hausarbeit S. Precht



Ein paralleles Programm mit Calliope
Achte auf das Zeitmanagement!

Vielen Dank für Ihre Aufmerksamkeit!