

pädagogische hochschule schwyz

Masterarbeit

**Mit neuester Technologie das Netzwerk
kennenzulernen – eine Unterrichtseinheit mit
LoRa und Raspberry Pi**

Autor:

Beat TEMPERLI

beat.temperli@stud.phsz.ch

+41 79 487 46 06

Matrikelnummer 08-910-556

Betreuung:

Prof. Dr.

Mareen GRILLENBERGER

21. Juli 2022

„Ich würde hier gerne einen Witz über UDP hinschreiben, aber ich weiss nicht, ob den überhaupt jemand liest.“

Aus dem WWW

Zusammenfassung

Beat TEMPERLI

Mit neuester Technologie das Netzwerk kennenlernen – eine Unterrichtseinheit mit LoRa und Raspberry Pi

Die Jugendlichen aus dem Zyklus 3 haben heutzutage täglich Berührungen mit den Themen «Internet» und «Netzwerke». Sie nutzen Netzwerke für die tägliche Kommunikation und bewegen sich stetig im Internet. In der Schule ist es ein Thema, wie sich die Lernenden sicher im Internet bewegen oder wie das Internet sinnvoll genutzt werden kann. Wie das Internet und insbesondere die Netzwerke aber genau funktionieren, fehlt im Lehrplan 21. Die technologischen Hintergründe der Netzwerktheorie sollen in einer Unterrichtseinheit behandelt werden und auf ein Niveau heruntergebrochen werden, dass die Schülerinnen und Schüler beispielsweise verstehen, wie das funktioniert, wenn von einem Mobiltelefon eine SnapChat-Nachricht an ein anderes Mobiltelefon gesendet wird. Aufgrund der Breite im Gebiet der Netzwerktheorie soll die Unterrichtseinheit ihren Fokus auf einzelne spezifische Bereiche legen und anhand von Protokollen aufzeigen, wie Nachrichten durchs Internet übermittelt werden und beim ausgewählten Empfänger ankommen.

Die Unterrichtseinheit soll mit Hilfe von moderner Technologie aus dem Bereich von «Physical Computing» unterstützt werden. In diesem Bereich werden Micro-Computer dafür eingesetzt, dass die Lernenden diese Geräte anfassen, zusammenbauen und insbesondere selber programmieren können. Computerprogramme können so direkt umgesetzt werden, Technologie kann angefasst werden. Dieser Bereich hat sich in den letzten Jahren im schulischen Umfeld festgesetzt und wird laufend weiterentwickelt. Dabei kann unterschieden werden zwischen geschlossenen, oft kindergerechten, Systemen, die in sich sehr gut funktionieren, aber nicht beliebig erweiterbar sind, und offenen Systemen, die selber beliebig ergänzt werden können, aber ein grösseres technisches Fachwissen verlangen. Es wird aufgezeigt, dass die offene Technologie rund um den Microcontroller «Raspberry Pi» mit der grafischen Programmieroberfläche «Scratch» verbunden werden kann und dabei einige gängige Hürden wie «Verbindungsschwierigkeit» oder «textbasierte Programmierung» umgangen werden können. In offenen Systemen können modernste Technologien eingesetzt werden, die den Weg noch nicht in das schulische Umfeld oder in geschlossene Umgebungen gefunden haben. So wird in der Unterrichtseinheit die Kommunikation zwischen den Geräten der Lernenden mit der neuen Übertragungstechnologie «LoRa» durchgeführt.

Die vorliegende experimentelle Entwicklungsarbeit soll aufzeigen, dass die Inhalte der Unterrichtseinheit durch den Einsatz von «Physical Computing» enaktiv erlebt werden können. Die Lernenden erfahren dabei an einem möglichst realen Gegenstand, wie ihre eigens programmierten Protokolle funktionieren und ausgeführt werden. Gleichzeitig soll aufgezeigt werden, dass die für Programmiernovizen etwas verständlichere «grafische Programmierung» mit offenen Systemen aus dem Bereich «Physical Computing» verknüpft werden kann und dabei diverse Hürden entfernt oder zumindest verringert werden.

Inhalt

Zusammenfassung	ii
1 Einleitung & Motivation	1
1.1 Kommunikation bei Jugendlichen	1
1.2 Hardware hinter der Kommunikation	2
2 Theoretischer Teil	4
2.1 Netzwerktechnologie	4
2.1.1 «LoRaWAN» x «Internet of Things»	6
2.1.2 Technische Details zu «LoRa»	7
2.1.3 Kommunikation in Netzwerken	9
2.2 Physical Computing	11
2.3 Lerntheoretische Überlegungen	12
2.3.1 Repräsentation von Lerngegenständen	13
2.3.2 Didaktische Reduktion	15
2.4 Netzwerk-Theorie für Sekundarstufe 1	15
2.4.1 Spiralcurriculum	16
2.4.2 Mögliche Themen	16
2.4.3 Vergleich mit internationalen Curricula	18
2.4.4 Annahmen aus den Vereinigten Staaten	19
2.5 Unterrichtsbeispiele zum Thema «Netzwerke» und «Kommunikation» . .	20
2.6 Fehlvorstellungen	21
2.6.1 Fehlerquellen	22
2.7 Ausblick auf die Entwicklung	23
3 Entwicklung	24
3.1 Physical Computing: geeignete Soft- und Hardware	24
3.1.1 Hardware	25
3.1.2 Programmiersprache	26
3.1.3 Auswahl Soft- und Hardware	30
3.2 Kommunikation mit Hilfe von Blockprogrammierung	32
3.2.1 Kommunikation zwischen Scratch und Python	33
3.2.2 1:1-Kommunikation	37
3.2.3 Erweiterungen	38
3.2.4 Weitere Scratch-Blöcke zur Unterstützung	40
3.3 Fehlerquellen beseitigen	40
3.3.1 Adressierung	40
3.3.2 Protokollverarbeitung	41
3.3.3 Parallelität vereinfachen	42
3.3.4 Sichtbarkeit der Datenübertragung	42
4 Unterrichtsszenario	43

4.1	Präsentation Unterrichtsszenario	43
4.1.1	Didaktische Reduktion	44
4.1.2	Erworbenene Kompetenzen	45
4.1.3	Kommentar für Lehrpersonen	46
4.1.4	Aufgabenkarten	46
4.1.5	Protokoll	48
4.1.6	Klassengrösse	50
4.1.7	Zeitaufwand	50
4.2	Erprobung Szenario	50
4.2.1	Technische Probleme	51
4.2.2	Konzeptuelle Probleme	52
4.3	Evaluation Szenario	53
4.3.1	Teil 1: Übersicht	53
4.3.2	Teil 2: Aufgaben & Hilfe	55
4.3.3	Teil 3: Lerninhalte	56
4.4	Überarbeitung Szenario	57
4.4.1	Überprüfung Hardware	57
4.4.2	Verbindungsprobleme Scratch und Python	58
4.4.3	Weiterentwicklung Scratch	58
4.4.4	Detaillierte Auswertung der Unterrichtseinheit	59
4.5	Veröffentlichung Unterrichtsszenario	59
5	Ergebnisse & Diskussion	60
5.1	Zusammenfassung und Ausblick	60
5.1.1	Weiterentwicklung der Ergebnisse	60
5.2	Fazit	61
5.2.1	Neue Technologie	62
5.2.2	Netzwerktechnologie im Unterricht	62
5.2.3	Netzwerke im Zyklus 3	63
5.2.4	Netzwerktechnologie erleben	63
5.2.5	Ausblick und offene Fragen	64
A	Auflistung Hardware	65
A.1	Endgerät «Scratch»	65
A.2	Endgerät «Watcher»	66
B	Auflistung Software	67
B.1	Endgerät «Scratch»	67
B.2	Endgerät «Watcher»	68
C	Unterrichtsmaterialien	69
C.1	Aufgabenkarten & Hilfestellungen	69
C.2	Kommentar für Lehrpersonen	71
D	Auswertung	81
D.1	Rückmeldungen Unterrichtseinheit «Netzwerk»	81
	Literaturverzeichnis	87

Abbildungsverzeichnis

2.1	Übertragung von Daten	4
2.2	OSI-Modell	10
3.1	Schema Verbindung zwischen User-Input/Output ↔ Scratch ↔ Python ↔ LoRa-Input/Output	32
3.2	Leerer Scratch-Block für die Übergabe einer beliebigen Eingabe an die LoRa- Erweiterung	33
3.3	Scratch-Block für die Übergabe einer «Eingabe» an die LoRa-Erweiterung .	33
3.4	Scratch-Block für die Verbindung von einer ganzen Botschaft	37
3.5	«Scratch»-Programm zur Überprüfung, ob eine Nachricht an Gerät A ge- richtet ist	38
3.6	«Scratch»-Programm, das eine Nachricht ausgibt und eine Antwort erfasst	38
3.7	Ausgabe vom «Watcher», Anzeige aller eingegangenen Nachrichten	41
3.8	Scratch-Block für die Verbindung von zwei Texten	41
3.9	Scratch-Block für die Entfernung einer bestimmten Anzahl Zeichen	42
4.1	Aufgabe 1: Anleitung	46
4.2	Aufgabe 1: Hilfestellung	46
4.3	Aufgabe 2: Anleitung	47
4.4	Aufgabe 2: Hilfestellung	47
4.5	Aufgabe 3: Anleitung	47
4.6	Aufgabe 3: Hilfestellung	47
4.7	Aufgabe 4: Anleitung	48
4.8	Aufgabe 4: Hilfestellung	48
4.9	Mögliches Protokoll zur Kommunikation im Netzwerk mit mehreren Geräten.	49
4.10	Material einer einzelnen Gruppe bei der Durchführung	51
4.11	Ein Schüler bei der Bedienung des Endgeräts «Scratch»	51
4.12	Zwei Mädchen beim Entwickeln ihrer Lösungsstrategie	51
4.13	Projektion des Endgeräts «Watcher» für die Darstellung aller versendeten Nachrichten	51
4.14	Auswertung Frage 3	54
4.15	Auswertung Frage 13	55
4.16	Auswertung Frage 14	55
4.17	Auswertung Frage 22	57
C.1	WhatsApp-Icon	69
C.2	SnapChat-Icon	69
C.3	Mobiltelefon mit einigen Apps	69

Tabellenverzeichnis

2.1	Klassifizierung Übertragungstechnologien nach Wang et al. (Wang u. a. 2019).	5
2.2	Interpretation einzelner Kompetenzen aus dem Lehrplan (D-EDK 2016).	17
3.1	Übersicht über die verschiedenen analysierten Umgebungen für die Blockprogrammierung.	28
4.1	Planung Unterrichtseinheit.	50
A.1	Materialliste Endgerät «Scratch».	65
A.2	Materialliste Endgerät «Watcher».	66
B.1	Software Endgerät «Scratch».	67
B.2	Software Endgerät «Scratch-Device».	68

Abkürzungsverzeichnis

dBm	Dezibel Milliwatt
GSM	Global System for Mobile Communications
IANA	Internet Assigned Numbers Association
IoT	Internet of Things
IP	Internet Protocol
LAN	Local Area Network
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LPN	Low Power Network
LTE	Long Term Evolution
RSSI	Received Signal Strength Indication
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WWW	World Wide Web

Für innovative & moderne Technologie in der Schule.

Kapitel 1

Einleitung & Motivation

1.1 Kommunikation bei Jugendlichen

«Der mediale Alltag von Jugendlichen ist schon seit Längerem durch das Handy und das Internet geprägt. In der Schweiz sind kaum noch Jugendliche zu finden, die nicht täglich ein Handy und das Internet nutzen.»

Süss, Waller, Bernath u. a. 2020

Schülerinnen und Schüler in der Schweiz nutzen ihr Mobiltelefon täglich, surfen im Internet und kommunizieren mit ihren Freunden, ihrer Familie, aber auch in der Schule (Keller 2020). Mit den digitalen Möglichkeiten werden Mitteilungen heutzutage online versendet, gelesen - und das nicht wenig (Steil 2022). Die Lernenden dürfen in den Schulen ihre Hausaufgaben online einreichen und Prüfungen digital schreiben (Federli 2022). Nicht von ungefähr zeigen Studien regelmässig (Süss, Waller, Huber u. a. 2014, Süss, Waller, Suter u. a. 2016, Süss, Waller, Külling u. a. 2018), dass die Kinder sich sehr oft im Internet bewegen.

«Man kann nicht *nicht* kommunizieren.»

Watzlawick, Bavelas und Jackson 2011

Schülerinnen und Schüler in der Schweiz kommunizieren also relativ oft. Watzlawick et al. meinen mit diesem Zitat zwar insbesondere die physische Interaktion zwischen zwei Menschen, die in jedem Fall von Kommunikation geprägt ist, auch wenn die beiden Personen eben nicht miteinander sprechen. Mit dem Hintergrund der Nutzung von Internet ist es heute für Jugendliche allerdings fast nicht mehr möglich, **nicht** zu kommunizieren. Das Mobiltelefon ist zum ständigen Begleiter worden (Süss, Waller, Bernath u. a. 2020) und die Schule weiss dies zu nutzen (Keller 2020). Der Umgang mit mobilen Geräten hat seinen festen Platz im Lehrplan 21 (D-EDK 2016) erhalten und im Medien- und Informatikunterricht wird dieser Umgang geschult.

«In einer Zeit, wo das Internet so alltäglich, präsent und vor allem gesellschafts-verändernd ist, muss dieses Themengebiet in der Schule aufgegriffen werden.»

Modrow und Strecker 2016

Und hier beginnen die Probleme: Die Schülerinnen und Schüler nutzen das Internet täglich, doch ist die Technologie dahinter kein Thema im Unterricht. Wie funktioniert das Internet? Wie ist es möglich, dass eine WhatsApp-Nachricht beim Gegenüber auf dem Telefon ankommt?

1.2 Hardware hinter der Kommunikation

Digitale Kommunikation benötigt entsprechende Hardware. In der Schule werden bereits Computer und Tablets eingesetzt (Barras und Petko 2007), eine andere Art von Hardware ist insbesondere im Informatikunterricht unter dem Begriff von «Physical Computing» zu finden. Mit Hilfe von «Physical Computing» werden Computer begreifbar gemacht, die Lernenden können die Geräte selber in die Hand nehmen, zusammenbauen, ihre eigenen Projektideen verwirklichen und die Geräte selber programmieren. Verknüpft man «Physical Computing» mit der Kommunikation, kommt man zum «Internet of Things». In diesem «Internet der Dinge» kommen Netzwerke zum Einsatz, denn die kleinen «Dinge» kommunizieren miteinander und mit dem Internet. Können nun diese Geräte aus dem Bereich des «Physical Computing» miteinander kommunizieren, haben wir die Hardware hinter der Kommunikation, ausgelegt für Lernende in der Schule. Die oft moderne Technologie hinter «Physical Computing», verbunden mit den Möglichkeiten vom «Internet of Things», in Verbindung mit Unterrichtsthemen, die bisher noch gar nicht berücksichtigt wurden, eröffnet uns eine neue Lernwelt für Schülerinnen und Schülern in einer modernen Schule. Moderne Technologien werden in der Industrie erforscht, entwickelt und angewendet, bis diese neuen Errungenschaften im schulischen Umfeld landen, dauert es seine Zeit. Mit dieser vorliegenden Arbeit soll sich das im Bereich der Netzwerktheorie ändern.

Physical Computing ist bereits seit einiger Zeit im Unterricht angekommen, viel Erfahrung konnte gesammelt werden und die Forschung hat bereits erarbeitet, was dabei beachtet werden soll. Diese Grundlagen sollen berücksichtigt und Möglichkeiten gefunden werden, die gängigen Probleme von Beginn an zu beseitigen. So haben sich beispielsweise zwei Varianten durchgesetzt: Geräte, die am Computer angeschlossen werden und mit diesem zweiten Gerät programmiert werden oder eigene kleine Computer, die direkt programmiert werden können.

Bei der ersten Variante befinden wir uns oft in geschlossenen, eingeschränkten Systemen. Ein mögliches Beispiel hierfür wäre der «Lego Mindstorms», der im geschlossenen Lego-System erbaut und mit eigenen Sensoren und Aktoren erweitert werden kann. Die Programmierung findet blockbasiert mit einer grafischen Programmieroberfläche oder textbasiert, mit *echten*¹ Code-Zeilen statt. In beiden Fällen wird das erstellte Programm auf das Gerät geladen und funktioniert - oder auch nicht. Probleme, die in diesen in sich geschlossenen Systemen auftauchen, sind oft Verbindungsschwierigkeiten und dass es umständlich ist, wenn die Geräte für jede Anpassung erneut an den Computer angeschlossen werden müssen. Zudem ist die Erweiterbarkeit oft nur eingeschränkt, sprich mit System-internen Teilen, möglich.

Bei der zweiten Variante sind wir in offenen Systemen, mögliche Geräte aus dem Schulalltag heissen «Arduino» oder «Raspberry Pi». Bei diesen Geräten erfolgt die Programmierung ebenfalls via zweitem Computer, im Fall eines «Raspberry Pi» könnte sie aber auch direkt auf dem Gerät erfolgen. In aller Regel wird hier textbasiert programmiert, was als Einstieg in die Programmierung nicht empfohlen ist (siehe [Unterabschnitt 3.1.2](#)). Diese offenen Systeme können dafür mit beliebigen Sensoren und Aktoren ausgestattet werden, die Erweiterbarkeit für eigene Experimente ist deutlich besser gegeben.

Moderne Technologien erfordern oft die Verwendung von offenen Systemen, da sie - zumindest zu Beginn - nur für solch offenen Systeme entwickelt werden, um möglichst schnell

¹Das soll nicht bedeuten, dass eine graphische Programmierung «unecht» wäre, sondern dass bei Code-Zeilen der gedankliche Schritt hin zu einem komplizierten Computerprogramm viel kürzer ist als er bei einer graphischen Darstellung eines Programms sei. Im Hintergrund werden beide Programmeingaben, textbasiert oder graphisch, nochmals neu kompiliert, damit sie auf einem «Lego Mindstorms» funktionieren.

Verbreitung zu finden. Die offenen Systeme bringen Hürden mit, wie die textbasierte Programmierung, die für den Schulalltag ungeeignet sind. Mit dieser vorliegenden Arbeit soll ein Weg gefunden werden, dass solch offene Systeme mit grafischer Programmierung verknüpft werden können und Schülerinnen und Schüler direkt eine Kommunikation zwischen mehreren Geräten «greifbar» erleben können. Mit einer optimalen Lösung kann bereits im Zyklus 3², wo aktuell die blockbasierte Programmierung empfohlen ist, von den Lernenden selber ein Netzwerk entwickelt werden.

«Physical Computing» kann dafür eingesetzt werden, um die Technologie hinter der Netzwerkkommunikation kennenzulernen. Für Schülerinnen und Schüler aus dem Zyklus 3 stellt diese Verknüpfung eine Möglichkeit dar, die Funktionen des Internets selber zu erforschen.

Die vorliegende experimentelle Entwicklungsarbeit soll die folgenden Fragen klären:

- Welche neuen Ansätze für die Erschaffung von Netzwerken als Unterrichtsgegenstand gibt es aufgrund des stetigen Wandels im Bereich der Technologie?
- Welche Bereiche aus der Netzwerktechnologie sind sinnvoll für den Unterricht?
- In welcher Form sollen Schülerinnen und Schüler des Zyklus 3 die Netzwerktechnologie kennenlernen?
- Wie kann die Netzwerktechnologie vereinfacht und eingebettet werden, so dass der Unterrichtsgegenstand enaktiv erlebt werden kann?

²Der Zyklus 3 umfasst 3 Schuljahre und entspricht der Oberstufe (Schuljahre 7, 8 und 9), wo Jugendliche im Alter von 12 bis 15 Jahren unterrichtet werden.

Kapitel 2

Theoretischer Teil

2.1 Netzwerktechnologie

Um in einem Netzwerk Daten zu übertragen, braucht es drei Elemente: einen Sender, einen Empfänger sowie einen Kommunikationskanal. Einfache Übertragungen funktionieren nach einem simplen Schema, das Meinel und Sack (Meinel und Sack 2009) genauer erläutert haben: Eine Information wird umgewandelt, so dass eine übermittelbare Nachricht entsteht, die in ein übertragbares Signal umgewandelt werden kann. Die Nachricht wird nun mit Hilfe von einzelnen Signalen über einen Kommunikationskanal vom Sender an den Empfänger übertragen. Beim Empfänger werden die übertragenen Signale als Nachricht gesammelt und dann wieder zurückgewandelt, so dass die Information vom Sender beim Empfänger verstanden wird. Der Ablauf wird in *Abbildung 2.1* schematisch dargestellt.

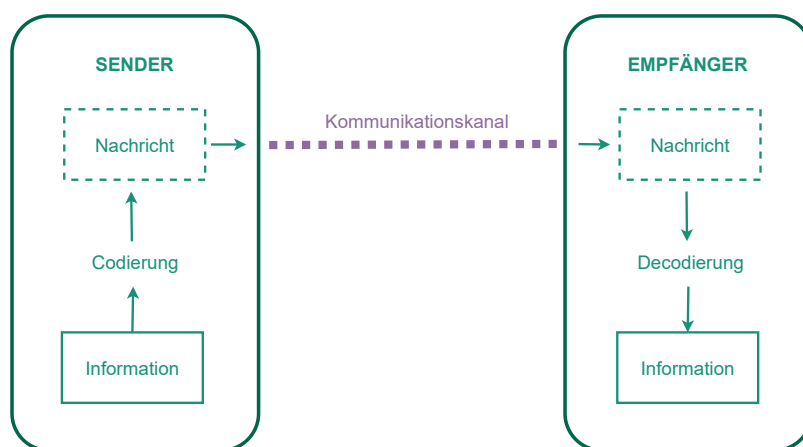


ABBILDUNG 2.1: Übertragung von Daten Quelle: Eigene Darstellung basierend auf Meinel und Sack 2009

Zwischen 1960 und 1970 wurden in den Vereinigten Staaten zum ersten Mal Computer miteinander verbunden (Janet Ellen Abbate 1994). Ein erstes Netzwerk entstand, um einzelne Computer von verschiedenen Hochschulen miteinander zu verbinden, das ARPANET. Um das Netzwerk zu nutzen, wurden erste Protokolle entwickelt, mit Hilfe derer innerhalb des Netzwerks kommuniziert werden konnte. Die Protokolle geben vor, wie einzelne Informationen verarbeitet werden, bevor sie übertragen werden, und wie kontrolliert werden kann, dass die Übertragung erfolgreich war.

Die Computer wurden damals analog miteinander verbunden oder konnten, relativ schnell nach der Entwicklung des ARPANETs, über Satellitenverbindung miteinander kommunizieren. Die Signale wurden dementsprechend entweder via Funk übertragen oder durch ein Kabel versendet. Im Prinzip hat sich daran bis heute nichts geändert. Es wurden neue

Protokolle entwickelt und neue Kommunikationskanäle erschaffen, aber das Schema aus [Abbildung 2.1](#) hat nach wie vor Bestand. Die folgenden Kommunikationskanäle wurden neu entwickelt und haben sich in der heutigen Netzwerkwelt durchgesetzt.

- GSM für die Telefonie via Mobilfunknetz, ab ca. 1990 (Eberspächer, Vögel und Bettstetter 2013)
- WLAN für die drahtlose Übertragung des LAN-Signals, ab ca. 1997 (Hiertz u. a. 2010)
- Bluetooth für die nahe Übertragung von Informationen via Funk, ab ca. 1999 (Bisdikian 2001)

In den letzten Jahren gab es nun vermehrt Entwicklungen, die spezifisch auf das «Internet of Things»³ ausgerichtet waren. Wang et al. (Wang u. a. 2019) haben verschiedene dieser neuen Übertragungstechnologien aus dem Bereich analysiert und unter Berücksichtigung von Energieverbrauch und Distanz zusammengefasst. Die Klassifizierung wird in [Tabelle 2.1](#) aufgeführt.

Low Power Tiefer Energieverbrauch			High Power Hoher Energieverbrauch	
Kurzdistanz	Mitteldistanz	Langdistanz	Kurz- /Mitteldistanz	Langdistanz
BLE WiSun ZigBee ...	Dash7 Ingenu/PRMA Weightless(N/P/U) ...	LoRa NB-IoT NB-Fi Sigfox Telensa ...	Wi-Fi ...	LTE LTE-A 5G ...

TABELLE 2.1: Klassifizierung Übertragungstechnologien nach Wang et al. (Wang u. a. 2019).

Alle diese Übertragungstechnologien sind im Bereich «IoT» einsetzbar. Vorteile ergeben sich aber, sobald mit geringem Energieverbrauch über weite Distanzen kommuniziert, beziehungsweise Informationen versendet werden kann. Mit diesem Hintergrund ragen die Technologien «LoRa», «NB-IoT», «NB-Fi», «Sigfox» und «Telensa» aus der Analyse von Wang et al. hervor. Wang et al. konnten bei den gewählten Technologien sichere Übertragungs-Distanzen im Mittel (Median-Werte) zwischen 1046m («LoRa») und 2061m («NB-IoT») messen. Diese Langdistanztechnologien übertragen Daten auf einer Frequenz von 868 oder 900 MHz.

³Das «Internet of Things» verbindet einzelne Geräte mit dem Internet und lässt die Geräte automatisch miteinander kommunizieren und handeln. Die Entwicklung hinter dem IoT verspricht, verschiedene Aspekte unseres Lebens zu transformieren. Neue IoT-Geräte kommen beispielsweise aus dem Bereich der Energieverwaltung oder Heimautomation und bringen uns hin zum «Smart Home», lassen uns sicherer leben und Energie effizienter nutzen. Tragbare Fitness-Tracker oder Gesundheits-Applikationen ändern unser Bewusstsein für unsere Gesundheits-Dienste. Systeme von verknüpften Fahrzeugen, intelligenten Verkehrsleitsystemen oder Sensoren in Brücken und Strassen bringen uns hin zu «Smart Cities».

Hinweis

Im Hinblick auf die Entwicklung einer IoT-Anwendung (siehe Kapitel 3), bei der Daten via Langdistanztechnologie übertragen werden, stellt sich im Rahmen dieser Arbeit die Frage nach der Verfügbarkeit der jeweiligen Technologie, sprich wie viele Adapter und Programme verfügbar sind. Für den Entwicklungsprozess in Kombination von «Physical Computing» sind deutlich mehr Adapter für «LoRa» als für die anderen Technologien erhältlich⁴. Da ein Teil der Arbeit daraus besteht, die Technologie auch in der Realität umzusetzen, wird bereits zu Beginn darauf geachtet, dass die neue Netzwerktechnologie mit den gängigen Technologien aus dem Bereich «Physical Computing» umgesetzt werden kann. Aus diesem Grund wird die Technologie «LoRa» gewählt und hier weiter vorgestellt.

⁴Vergleich Google-Suche nach «„LoRa“ development board» und «„sigfox“ development board» ergibt 60 Mal mehr Suchresultate für den LoRa-Suchbegriff: 11'200'000 zu 193'000.

2.1.1 «LoRaWAN» x «Internet of Things»

Die «LoRa»-Technologie wurde von SemTech⁴ für den Einsatz als Low Power Wide Area Network entwickelt. «LoRa» ist eine Funktechnologie, die stromsparend über weitreichende Distanzen Daten übertragen lässt. «LoRaWAN» umschreibt das ganze Netzwerk rund um «LoRa», das die Kommunikation via «LoRa» erst ermöglicht. Im «Internet of Things» kommt die Technologie rund um LoRa und LoRaWAN klassischerweise mit drei Elementen zur Geltung: LoRa Sensoren, LoRa Gateways und LoRa Server.

LoRa Sensor

LoRa Sensoren oder «Nodes» sind kleine Geräte, meistens mit einer Batterie versehen und deshalb unabhängig vom Stromnetz. Die «Nodes» senden Daten, die sie meist direkt aus der näheren Umwelt (Temperatur, Wetter, Heimautomation) gesammelt haben, via «LoRaWAN» an einen Gateway in der Nähe.

LoRa Gateway

LoRa Gateways sammeln übers LoRaWAN gesendete Daten von «Nodes» ein und leiten diese weiter ins Internet, entweder via Mobilfunk, sprich GSM beziehungsweise LTE, oder direkt via LAN.

LoRa Server

Der LoRa Server trägt die gesammelten Daten zusammen und lässt eine Auswertung zu. Ebenfalls können hier die einzelnen Nodes verwaltet werden, so dass nur berechtigte oder «eigene» Nodes Daten sammeln können.

Zurück zu «LoRaWAN». Ein LoRaWAN kann mit den oben genannten drei Elementen selber erstellt und betrieben werden. Da das «Internet of Things» wenn möglich ortsunabhängig funktionieren soll, ist eine flächendeckende Verbreitung von Gateways essentiell. Einzelne Nodes können von überall Daten senden, die Daten müssen aber jeweils von einem Gateway empfangen werden. Bei Distanzen von wenigen hundert Metern ist das noch nicht relevant, bei kilometerweiter Übertragung benötigt man für ein funktionierendes Netzwerk

⁴Firma «SemTech Corporation», Kalifornien: www.semtech.com/lora

auch entsprechend verteilte Gateways. In der Schweiz gibt es mehrere Initiativen, die ein Netzwerk von Gateways erstellen, das dann von Firmen und Privatpersonen genutzt werden kann. So muss für den Einsatz von Nodes nicht immer gleich ein ganzes Gateway-Netzwerk erstellt werden. Die Swisscom baut ein kostenpflichtiges Low Power Network - basierend auf LoRaWAN⁵ - auf, daneben gibt es zwei Projekte, die von offenen Gemeinschaften betrieben werden: «The Things Network» und «LORIoT».

Das «The Things Network»⁶ ist ein offenes Gemeinschaftsprojekt von Privatpersonen, die ihre eigenen Gateways betreiben, die dann vom Netzwerk zusammengefasst werden. Gleichzeitig wird eine Server-Infrastruktur für die Verarbeitung der empfangenen LoRa-Daten zur Verfügung gestellt, damit eine Auswertung oder weitere Verarbeitung der Daten ermöglicht wird. Bei «LORIoT»⁷ werden die LoRa-Server zur Verfügung gestellt, entweder kostenpflichtig für Unternehmen oder gratis für die IoT-Gemeinschaft. Die Gateways und Nodes müssen selber organisiert werden.

Um die «LoRa»-Technologie nutzen zu können braucht man in erster Linie Sender und Empfänger. Für eine Kommunikation über diese neue Funkübertragung müssen die Geräte in der Lage sein, Daten zu senden und gleichzeitig auch zu empfangen, nur so kann eine bidirektionale Kommunikation stattfinden. Sobald ein grösseres System benötigt wird, wo einzelne Geräte Daten sammeln, diese weiterleiten und an einem dritten Ort die Daten verarbeitet werden, nutzt man optimalerweise ein LoRaWAN, also ein ganzes Netzwerk, das bereits für diesen Zweck entwickelt oder gar bereits erstellt wurde.

2.1.2 Technische Details zu «LoRa»

LoRa basiert auf einem physikalischen Layer, der die Kommunikation über Long Range, also lange Distanzen, ermöglicht (Devalal und Karthikeyan 2018). Im LoRaWAN ist genau definiert, wie die Kommunikation erfolgen muss, es sind Standards definiert und die ganzen Spezifikationen sind inzwischen öffentlich verfügbar und werden von der LoRa-Allianz⁸ weiterentwickelt. Devalal und Karthikeyan führen weiter die folgenden technischen Details auf.

Chirp Spread Spectrum Modulation

Die Chirp-Spread-Spectrum-Modulation trägt dazu bei, dass die Daten über eine längere Distanz transportiert werden können, wie Devalal und Karthikeyan in ihrer Übersicht aufzeigen. Grundsätzlich wird bei der Funktechnologie ein Signal über eine bestimmte Bandbreite in einer bestimmten Frequenz übertragen. Durch die Modulation, die über die Sendezeit aktiv hervorgerufen wird, vergrössert oder verkleinert sich die Frequenz während dem Senden. Diese Modulation trägt dazu bei, dass Rauschen und Interferenzen reduziert werden.

⁵Die Swisscom erstellt das LPN LoRaWAN in eigener Initiative und stellt es Unternehmen kostenpflichtig zur Verfügung (siehe Swisscom (Schweiz) AG 2021). In Zusammenarbeit mit SBB und Post wird es laufend ausgebaut und deckt gemäss eigenen Angaben inzwischen 97% der Schweiz ab. Es ist dafür gedacht, dass Firmen ihre IoT-Nodes schweizweit einsetzen und vernetzen können.

⁶thethingsnetwork.org

⁷www.loriot.io

⁸Die «LoRa-Alliance» ist eine offene und nicht-profit-orientierte Organisation, die «LoRa» weltweit bewirbt, weiterentwickelt und die Personen und Unternehmen dahinter unterstützt und zusammenbringt. lora-alliance.org

Link Budget

Die Übertragungsbilanz (oder englisch: Link Budget) im «LoRa»-Netz ist sehr hoch. In die vereinfachte Bilanz fließen vier Punkte ein: die Sendeleistung, die Empfängerempfindlichkeit, der Antennengewinn⁹ sowie die Freiraumdämpfung¹⁰ (siehe Aref und Sikora 2014). Verglichen mit anderen Technologien hat «LoRa» die beste Übertragungsbilanz, meinen Devalal und Karthikeyan (Devalal und Karthikeyan 2018).

Spreading Factor

Durch den Spreading Factor, der beim Senden manuell beeinflusst werden kann, wird beim Erhöhen einerseits die Distanz vergrößert, aber gleichzeitig auch die übertragene Bitrate verkleinert. Ein grosser Spreading Factor bedeutet zwar längere Distanzen, aber verkleinert die Datenpakete, die übertragen werden können. Ein kleiner Spreading Factor ermöglicht grössere Pakete über kleinere Distanzen. Bei der Konfiguration von «LoRa»-Geräten kann der Spreading Factor zwischen 7 und 12 selber gewählt werden.

RSSI

Die *Received Signal Strength Indication* (RSSI) bezeichnet die Signalstärke, in welcher ein LoRa-Signal beim Empfänger ankommt. Die Signalstärken liegen bei Datenübertragungen in der Theorie zwischen -130 dBm und 0 dBm - je höher der Wert desto stärker der Empfang und desto erfolgreicher kommt das jeweilige Datenpaket beim Empfänger an (Petäjälä et al. 2017). In der Praxis bewegen sich die Werte zwischen -50 dBm und -140 dBm, wie Tan et al. (Tan u. a. 2019) in ihrem Versuch aufzeigen. Dabei spielen die Faktoren «Umgebung» (ländlich, suburban oder urban) und «Distanz» eine grosse Rolle. In ihrem Versuch konnten sie aufzeigen, dass eine Übertragung in ländlichen Gebieten bis zu einer Distanz von 9.7 Kilometern und in städtischen Gebieten bis 4.6 Kilometern möglich ist. Bei diesen weiten Distanzen kommen die RSSI-Werte bei -140 dBm zu liegen.

Lebenszeit Batterie

«LoRa» verbraucht von allen gängigen Übertragungstechnologien am wenigsten Energie (Devalal und Karthikeyan 2018). Das führt dazu, dass LoRa-Anwendungen mit einer Batterie sehr lange überleben können. Die Kommunikation von batteriebetriebenen Endgeräten erfolgt deshalb optimalerweise über «LoRa». Einer der Gründe dafür ist, dass bei «LoRa» keine stetige Synchronität erwartet wird, die Endgeräte benötigen also nur Energie, wenn Daten versendet werden und nicht von Zeit zu Zeit, um sicherzustellen, dass auf der Empfängerseite ein Gerät verbunden ist.

⁹Antennen mit hohem Antennengewinn sind stark richtungsempfindlich, sprich die beiden Antennen müssen möglichst genau aufeinander gerichtet sein, um Daten zu übertragen. Antennen mit tiefem Antennengewinn sind unabhängiger von der Senderichtung.

¹⁰Mit der Freiraumdämpfung ist gemeint, wie stark Signale in der Umwelt gedämpft werden. Optimalerweise bei der Übertragung im freien Raum, aber genauso durch Gebäude, Bäume oder andere Gegenstände, die den Weg der Übertragung kreuzen.

Sicherheit

Die Kommunikation im Netzwerk findet verschlüsselt über die *AES*¹¹-Verschlüsselung statt. Die Verschlüsselung findet dabei auf zwei Ebenen statt: Erstens wird die Netzwerk-kommunikation verschlüsselt, damit man nur innerhalb des Netzwerkes überhaupt Zugriff auf den Inhalt erhalten kann - sofern man den Schlüssel besitzt. Zweitens wird die Kommunikation für eine einzelne Applikation im System von Nodes und Gateways separat verschlüsselt, so dass der Netzwerkbetreiber keine Inhalte mitlesen kann - das ist nur für den Applikationsbesitzer möglich, der den entsprechenden Schlüssel generiert.

2.1.3 Kommunikation in Netzwerken

Kommunikation in Netzwerken ermöglicht es, Datenpakete zwischen verschiedenen Computern zu übermitteln. Dazu müssen einerseits die Computer miteinander verbunden sein, Datenpakete oder Nachrichten müssen vorhanden sein und die Computer benötigen eine Regel, wie sie miteinander kommunizieren können.

Im Internet, wie es heute existiert, existieren verschiedene Möglichkeiten, wie ein solches Netzwerk aufgebaut werden kann. Die Computer können drahtlos oder via Kabel miteinander verbunden werden. Damit sich beide Computer auch tatsächlich verstehen, ist es wichtig, dass diese in der gleichen Sprache kommunizieren, der eine Computer muss ja verstehen, was er vom anderen Computer erhält.

Die Internet Assigned Numbers Authority¹² [IANA] ist eine Behörde, die die Kommunikation im Internet regelt. Dazu definiert sie einzelne Protokolle für die jeweiligen Kommunikationsarten im Internet. Die aktuelle Liste an bestätigten Internetprotokollen umfasst mehr als 140 Einträge von Protokoll-Variationen, die jeweils einen bestimmten Nutzen abdecken (IANA 2021). Zu den bekannteren Protokollen gehört das Internet Protokoll, das heute in den Varianten IPv4 (Perkins 1996) und IPv6 (Deering und Conta 1998) existiert. In diesen beiden Protokollen wird definiert, wie einzelne Rechner in einem Netzwerk erreichbar sind, einmal in der vierten Version mit theoretisch ungefähr 4 Milliarden möglichen Rechner-Adressen, einmal in der neueren sechsten Version, welche für deutlich mehr mögliche Rechner-Adressen vorbereitet wurde.

Ein weiteres Protokoll ist das Transmission Control Protocol [TCP] (Postel 1981). Mit dem TCP können zwei Endpunkte in einem Netzwerk eine bidirektionale Verbindung herstellen, mit der beide Geräte miteinander kommunizieren können. Dazu muss von beiden Endgeräten die IP-Adresse (basierend auf dem *Internet Protokoll*) bekannt sein sowie ein Port definiert sein, über den die Verbindung erstellt wird. Eine klassische TCP-Verbindung wird wie folgt aufgebaut:

- Gerät **A** sendet **SYN** mit einer beliebigen Zahl zur Identifizierung an Gerät **B**
- Gerät **B** antwortet mit **SYN-ACK**, der empfangenen Identifizierung und einer neuen beliebigen Zahl an Gerät **B**
- Gerät **A** bestätigt mit **ACK** und der empfangenen Identifizierung an Gerät **B**

¹¹Advanced Encryption Standard [AES] ist eine Verschlüsselungstechnologie, die ein symmetrisches Verschlüsselungsverfahren anbietet, das - Stand heute - als sicher eingestuft wird. Beim symmetrischen Verfahren kann man mit einem Schlüssel einen Text verschlüsseln und auch wieder entschlüsseln.

¹²Die IANA ist zuständig für die globale Koordination von DNS-Einträgen, IP-Adressierung und weiteren Internet-Protokollen.

- Die Verbindung zwischen den beiden Geräten ist aufgebaut, Daten können nun übertragen werden.

Verglichen mit TCP ist das User Datagram Protocol [UDP] (Postel 1980) eine Alternative mit Vor- und Nachteilen. Das UDP wurde entwickelt, um damit Sprachmitteilungen zu übermitteln. Bei solchen Datenübertragungen spielt es nicht per se eine Rolle, dass alle Daten ankommen, sondern dass diese schnell und ohne Verzögerung ankommen. Wenn unterwegs Teile der Übertragung verloren gehen, kann man trotzdem das wichtigste mithören. Der Vorteil ist hier, dass die Verbindung nicht zuerst aufgebaut werden muss sondern Daten direkt von Gerät A zu Gerät B übertragen werden können. Der Nachteil, der sich daraus ergibt ist, dass die Daten einfach verschickt werden und nicht überprüft wird, ob diese auch tatsächlich beim Gegenüber ankommen¹³. Eine bidirektionale Kommunikation ist so nicht möglich, aber in einigen Fällen ist das auch nicht erwünscht.

Diese einzelnen Protokolle erfordern alle immer eine bestehende Verbindung. Die beiden Geräte, die miteinander kommunizieren, müssen dazu auf physischer Ebene miteinander verbunden sein. Die «International Telecommunication Union» hat dazu ein Modell entwickelt, das aufzeigen soll, wie die Geräte genau miteinander kommunizieren können und wie sie physisch dazu in überhaupt erst in der Lage sind: das OSI-Modell (International Telecommunication Union 1994).

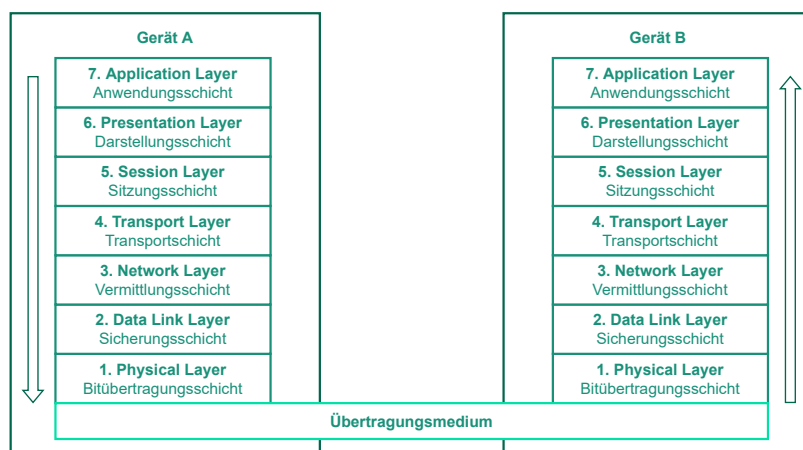


ABBILDUNG 2.2: OSI-Modell Quelle: Eigene Darstellung

In Abbildung 2.2 wird aufgezeigt, wie dieses OSI-Modell grundsätzlich funktioniert. Zwei Endgeräte, die über ein Übertragungsmedium miteinander verbunden sind, kommunizieren über verschiedene Ebenen hinweg miteinander. Die Daten folgen den beiden Pfeilen: in der Anwendungsschicht (7) wird die Verbindung eingeleitet, die Daten queren alle Layer im Gerät A und werden dabei so aufbereitet, dass sie am Ende als einzelne Bits über das Übertragungsmedium übertragen werden können. Die übertragenen Pakete werden anschliessend im Gerät B wieder so aufbereitet, dass aus den einzelnen übertragenen Bits die benötigten Daten für die Anwendungsschicht zusammengefügt werden.

Eine TCP-Verbindung bildet auf der Transportschicht (4) die einzelnen Pakete und initiiert die Verbindung zum anderen Gerät. In der Vermittlungsschicht (3) wird die Adressierung vorgenommen, so dass die Daten auch wissen, zu welchem Gerät diese gelangen sollen. Auf den unteren beiden Schichten Sicherung (2) und Bitübertragung (1) geschieht die Umwandlung von der IP-Adresse zur MAC-Adresse und anschliessend werden die einzelnen Bits übertragen.

¹³Damit sollte nun auch das Zitat von Seite i erklärt sein.

Die UDP-Verbindung wird ebenfalls auf der Transportschicht (4) eingeleitet und folgt den Prinzipien der TCP-Übertragung. Das UD-Protokoll unterscheidet sich aber darin, dass die Transportschicht auf dem zweiten Gerät nicht kontrolliert, dass alle Pakete empfangen wurden. Die Verbindung ist auch nur ein-direktional, so dass Pakete nicht auch gleich noch in die andere Richtung zurückgeschickt werden, wie das bei TCP der Fall ist.

2.2 Physical Computing

Der Begriff «Physical Computing» wird im schulischen Umfeld verbreitet verwendet und umfasst die Planung und Umsetzung von Projekten mit interaktiven Objekten oder Installationen (Romeike und Przybylla 2014). Schülerinnen und Schülern wird ermöglicht, konkrete, greifbare Produkte der realen Welt zu erbauen, die ihrer Phantasie entspringen.

In der Schule ist Physical Computing je länger je mehr ein Bestandteil vom Informatikunterricht oder anderen MINT-Fächern¹⁴, der allerdings den Weg in den Lehrplan noch nicht gefunden hat, wobei im Lehrplan grundsätzlich Kompetenzen und nicht die Methoden, die zu den Kompetenzen führen, erwähnt werden.

Die edukative Arbeit mit programmierbaren Bauteilen, unter anderem von Lego, startete bereits ums Jahr 1985 herum und wurde stetig weiterentwickelt (siehe Katterfeldt, Dittert und Schelhowe 2015). Die Bauteile wurden günstiger, besser verknüpfbar, und die Programmierung wurde einfacher. Vorteile haben sich etabliert, Nachteile wurden abgebaut. Physical Computing verbreitete sich rund um den Globus. Schon früh wurden Ansätze von «Verstehen» anstatt dem klassischen «Lernen», der Ansatz «Learning-by-doing» oder eben Ausprobieren, Planen, Scheitern und erneut Starten im Bereich von «Physical Computing» gelebt und gelehrt.

Lego hat 1998 die erste Version des «Mindstorms»-Roboters auf den Markt gebracht (The LEGO Group 2022), ein einfach zu programmierender Baustein, der mit Sensoren und Motoren versehen werden konnte und eingehende Sensordaten verarbeiten konnte, so dass der Roboter autonom darauf reagieren konnte. Der einfache Roboter vom Spielzeughersteller «LEGO» fand den Weg schnell in die Privathaushalte und wurde auch schon bald in der Schule eingesetzt. Lernende konnten selber einen kleinen Computer programmieren und ihn Dinge ausführen lassen, auf Signale von Sensoren mit Hilfe diverser Aktoren reagieren. Der «Mindstorms» wird dabei blockbasiert und in den späteren Versionen auch textbasiert programmiert. Goldweber, Fagin und McNally diskutierten bereits im Jahr 2006, aufgrund von diversen verfassten Studien zum Thema «Mindstorms in Education» darüber, ob dieser Lego-Roboter eine Zukunft hat im Unterricht (McNally u. a. 2006). Sie kamen zum Schluss, dass der Unterricht wohl durch den Einsatz nicht schlechter wird, die Studien aber auch keine klare Verbesserung aufzeigen würden. Schon 10 Jahre später, 2017, fassten Afari und Khine zusammen (siehe Afari und Khine 2017, dass die Lego-Roboter positive Veränderungen im Unterricht hervorrufen würden, sowie die Verknüpfung von mehreren Fachrichtungen (Mathematik, Informatik, Programmieren, etc.) eine Leistungssteigerung bei den Lernenden hervorrufen würde.

Neben dem «Mindstorms» sind viele weitere Tools in den Schulen erfolgreich eingesetzt worden, darunter die folgenden:

- Raspberry Pi Microcomputer, programmiert mit Python (Ariza und Gil 2022)
- Arduino Microcomputer, programmiert mit C++ (Sari u. a. 2022)

¹⁴MINT: Mathematik, Informatik, Naturwissenschaften und Technik.

- micro:bit, programmiert mit diversen, meist blockbasierten, Programmierumgebungen (Hodges u. a. 2020)
- ...

Die Liste kann beliebig weitergeführt werden, wie auch Przybylla und Romeike bereits ausgeführt haben (Przybylla und Romeike 2015). Dabei unterscheiden sie die kommerziell verfügbaren Geräte aus dem Bereich «Physical Computing» in folgende Kategorien:

- Verpackte Elektronik ohne Programmierung.
- Verpackte Elektronik mit Programmierung, oft via Computer oder Mobiltelefon programmierbar und mit Batterien betrieben (beispielsweise «LEGO Mindstorms»).
- Peripherie-Geräte mit einem Computerchip, die an einem Computer angeschlossen werden für die Programmierung und den Betrieb.
- Geräte mit einem Computerchip, die für die Programmierung an einen Computer angeschlossen werden müssen, aber ohne Computer betrieben werden können (beispielsweise «micro:bit» oder «Arduino»).
- Eigenständige Geräte mit einem Computerchip, die direkt programmiert werden können (beispielsweise «Raspberry Pi»).

Bei den letzten beiden Kategorien werden die Anwendungsmöglichkeiten riesig, da die Geräte oft beliebig erweitert werden können und analog einem eigenständigen Computer frei programmiert werden können. Je freier die Möglichkeiten, desto komplexer wird aber deren Ansteuerung und Programmierung, da nicht einfach wie beispielsweise bei einem «Mindstorms» nach einer einfachen Anleitung vorgegangen werden kann. Je eigenständiger das Gerät und je grösser die Möglichkeiten, desto stärker kommt das oben erwähnte «Ausprobieren, Planen, Scheitern und erneut Starten» zum Tragen. Die Geräte können und müssen frei programmiert werden, bis die Idee umgesetzt ist und funktioniert.

Ein wichtiger Punkt bei der Auswahl der richtigen Hardware ist hier sicher die Unterscheidung bei den letzten Kategorien, ob die Geräte grundsätzlich eigenständig sind und direkt programmiert werden können oder ob diese mit einem Computer verbunden werden müssen, damit sie programmiert werden können. Bei eigenständigen Geräten ergibt sich der Vorteil, dass die Verbindung zwischen Programmierung und Gerät direkt erfolgt und nicht erst hergestellt werden muss. Eine mögliche Fehlerquelle wird direkt eliminiert.

2.3 Lerntheoretische Überlegungen

Ausprobieren. Testen. Ideen umsetzen. Scheitern. Neu starten. Ideen verwerfen. Sich selber motivieren, weiter zu machen. Kreativ sein. Erfolgreich sein. Etwas lernen. Unbemerkt etwas lernen. Lernen durch Verstehen.

«Tell me and I forget, teach me and I may remember, involve me and I learn.»

Benjamin Franklin¹⁵

¹⁵Dieser Satz wird Benjamin Franklin zugeschrieben. Der Satz basiert auf einem chinesischen Sprichwort, welches wiederum Konfuzius zugeschrieben wird. Beides kann aber nicht mit Sicherheit überprüft werden und könnte auch einfach eine Legende aus den Weiten des Internets sein.

Dieser Satz soll aufzeigen, dass aus lerntheoretischer Sicht eine lernende Person möglichst nah am Unterrichtsgegenstand sein soll; der zu lernende Inhalt soll nicht einfach erzählt oder unterrichtet werden, sondern direkt erlebt und damit auch erlernt werden. Durch die direkte Einbindung der Lernenden verspricht man sich gemäss dem obenstehenden Zitat, dass die gelernten Inhalte eben tatsächlich erlernt und nicht einfach direkt danach wieder vergessen werden.

Ein geeignetes Feld für diese Lernmethoden bildet der Konstruktivismus ab. Begründet wurde dieser schon in der ersten Hälfte des zwanzigsten Jahrhunderts durch Piaget (unter anderem Piaget 1947). 2005 fasst Siebert die Theorie wie folgt zusammen: Lernen nach dem Konstruktivismus basiert darauf, dass man beim Lernen nicht die Realität erkennt wie sie ist, sondern sich diese konstruiert und sie so wahrnimmt, wie man damit auch umgehen kann (Siebert 2005). So funktioniert das Lernen als selbstgesteuerte Aktivität, die von aussen zwar angeregt werden könne, aber eben nicht einfach von aussen konstruiert vorgelegt werde. Die lernenden Personen erbauen sich ihr eigenes Umfeld, organisieren den Lernprozess selber und steuern soviel wie möglich selber. Von aussen werden einzelne Punkte hervorgehoben und angeregt, die Organisation des Lernens liegt bei den Lernenden selber. Etwas später fasst Siebert zudem einige Lernstile von Kolb (Kolb 1976) zusammen und erwähnt nun das «aktive Experimentieren» und erklärt den Term mit der Bereitschaft, neue Lösungen und Methoden zu erproben, meist nach dem Prinzip von «Versuch und Irrtum» (Siebert 2014). Er ergänzt, dass Lernen nur bedingt nach dem wissensvermittelnden Dialog mit dem Sender-Empfänger-Schema abläuft, sondern in erster Linie ein selbstgesteuerter Prozess ist, der von der lernenden Person ausgeht.

Mandl hat im gleichen Zeitraum zentrale Aspekte des konstruktivistischen Lernens zusammengefasst (Mandl 2006) und nennt unter anderem, dass Lernen ein aktiver Prozess ist, der erst durch die aktive Beteiligung der Lernenden möglich ist. Ein zweiter Punkt ist, dass Lernen ein selbstgesteuerter Prozess ist, der von den lernenden Personen ausgeht. Streller et al. fassen die Erkenntnisse dann soweit zusammen, dass eine Lernumgebung, die dem konstruktivistischem Ansatz folgt, der problemorientierte Unterricht sei (Streller u. a. 2019). Sie empfehlen reale Problemstellungen in möglichst authentischen Umgebungen und ein forschendes Lernen. Sie bringen weiter ein, dass Lernumgebungen eine kognitive Aktivierung beinhalten sollen. Lernende sollen vertieft über den Lerngegenstand nachdenken und sich aktiv mental damit auseinandersetzen.

Spezifisch schlagen sie unter anderem vor, dass Lernaufgaben die folgenden Punkte erfordern:

- Die Aufgaben erfordern spezifische Problemlöseprozesse.
- Die Aufgaben sind komplex und bestehen aus mehreren Komponenten.
- Die Aufgaben erfordern die Verknüpfung von mehreren bekannten Sachverhalten oder die Anwendung dieser auf neue Situationen.
- Die Aufgaben haben mehrere Lösungsmöglichkeiten.
- Die Aufgabe stellen nicht sämtliche Informationen bereit, sondern implizieren, dass die Lösungen von Lernenden gefunden werden müssen und auch gefunden werden können.

2.3.1 Repräsentation von Lerngegenständen

Bruner hat bereits 1966 aufgezeigt, dass Lernen in verschiedenen Formen geschieht, allerdings ohne bereits eine Wertung reinzubringen: enaktiv, sprich handelnd, ikonisch, sprich mit Hilfe von bildlichen Darstellungen oder symbolisch mit verbaler oder formalen Hilfen

(Bruner 1966). Diese drei verschiedenen Arten von Repräsentationen vom effektiven Inhalt sollen ermöglichen, Unterrichtsinhalte aus verschiedenen Perspektiven anzuschauen und je nach dem die beste Möglichkeit auszuwählen, um entsprechend gute Resultate beim Lernen zu erzielen. Mit der Repräsentation von Lerngegenständen wird nicht der Gegenstand an sich erlernt, sondern eine Repräsentation davon. Wir bewegen uns beim repräsentativen Lernen in der Nähe vom Modelllernen, der kognitiven Lerntheorie nach Bandura, die besagt, dass wir anhand von einem Vorbild durch Nachahmung lernen (Bandura und Kober 1976).

Mit diesen Repräsentationsmodi haben sich Hartmann et al. hinsichtlich des Lernens im Informatikunterricht beschäftigt und dabei zwei weitere Unterscheidungen eingeführt (Hartmann, Näf und Reichert 2006):

- Die **semi-enaktive** Methode, bei der eine Lehrperson etwas vorzeigt, also enaktiv tätig ist, die Lernenden aber nur zuschauen.
- Die **virtuell-enaktive** Methode, bei der Lernende enaktive Vorgänge in einer computergesteuerten Simulation erproben und erlernen. Die entsprechende Tätigkeit wird nur virtuell simuliert.

Die drei Autoren sprechen sich aber klar für die enaktive Variante aus, um neue Themen im Unterricht zu erlernen und geben gleich Hilfestellungen dazu. Enaktive Repräsentation kann nicht spontan geschehen sondern muss entsprechend vorbereitet werden, meist ist das aber im schulischen Umfeld und dann auch mit den vorhandenen Mitteln möglich. Bei aufwändigen Repräsentationen, die auch noch schwierig herzustellen sind, weil die Materialien nicht einfach vorhanden sind, rechtfertigt sich der Aufwand erst, wenn die Materialien dann wiederholt verwendet und eingesetzt werden können.

Modrow und Strecker, die sich in jüngster Zeit mit der Didaktik in der Informatik befasst haben, sind einig mit den Aussagen von Hartmann et al., und zeigen ebenfalls auf, dass die Klassifizierung von Bruner durchaus seine Berechtigung hat, klassifizieren dann aber beispielsweise für die Simulation eines fallenden Körpers bereits die enaktive Methode, worin sie sich von den neu eingeführten Unter-Klassen von Hartmann et al. distanzieren (Modrow und Strecker 2016). Gleichzeitig plädieren sie aber auch für ein Modelllernen, beschreiben es als Prozess, um Systeme umzusetzen, die mehr als Simulationen der Realität zu verstehen sind. Die Lernenden bilden in eigenen Modellen oft Prozesse ab, die dann aber nur in der Theorie umgesetzt werden, da sie wohl nicht voll funktionsfähig wären und in der Realität Bestand hätten. Hier nähern sie sich wiederum der virtuell-enaktiven Methode von Hartmann et al. an.

Krey und Schwanewedel fassen für den naturwissenschaftlichen Bereich verschiedene weitere Studien zusammen und zeigen auf, inwiefern die Repräsentation beim Lernen wichtig ist und wie diese eingesetzt werden kann (Krey und Schwanewedel 2018). Sie kommen zum Ergebnis, dass Lernen optimalerweise mit multiplen externen Repräsentationen geschieht, die sich teilweise wiederholen, überschneiden oder komplett unterscheiden. Sehr oft werden Repräsentationen gemäss ihren Quellen eingesetzt, um schwierige Sachverhalte verschieden darzustellen und dabei verschiedene Sinne anzusprechen (beispielsweise Texte und Bilder mit den gleichen Informationen für unterschiedliche Lernverhalten von Schülerinnen und Schülern). So kann auch erreicht werden, dass Vorlieben und Routinen von Lernenden berücksichtigt werden. Ebenfalls wechseln Krey und Schwanewedel die Perspektive und erläutern, dass nicht nur die Lehrperson vorgeben muss, wie eine Repräsentation ausschauen soll, sondern auch Lernende sollen in der Lage sein, komplexe Konstrukte aus der Realität vereinfacht in einer Repräsentation darzustellen.

2.3.2 Didaktische Reduktion

Die Netzwerktheorie umfasst verschiedene komplexe Themenbereiche, wovon in den vorangehenden Unterkapiteln ein paar wenige aufgezeigt wurden. Diese komplexen Themen sollen für den Unterricht vereinfacht und neu aufbereitet werden, sofern sie überhaupt für den Unterricht ausgewählt werden. Lehner definiert das als «didaktische Reduktion» (Lehner 2020). Dabei stehen zwei wichtige Vorgehensweisen im Fokus: Auswahl der Inhalte, so dass der Umfang der Lerninhalte bei der Unterrichtsplanung verringert werden kann; Konzentration und Vereinfachung der Inhalte, so dass die elementaren und fundamentalen Inhalte ausgearbeitet werden.

Mit der Verringerung der Inhalte darf es aber gemäss Streller et al. nicht passieren, dass dabei eine quantitative Verminderung von den Inhalten geschieht (Streller u. a. 2019), sondern es soll eine Reduktion auf die wesentlichen und elementaren Inhaltspunkte geschehen, wie das Lehner im zweiten Punkt erwähnt.

Streller et al. fassen zusammen, dass bei der didaktischen Reduktion die Abstraktheit eines Sachverhalts und dessen Komplexität auf das Wesentliche zurückgeführt wird, so dass die Lernenden den tatsächlichen Inhalt überschauen und begreifen können (Streller u. a. 2019; Bleichroth 1991; Rösler 1996; Lehner 2020). Sie erwähnen abschliessend, dass vereinfachte Aussagen aber wissenschaftlich zulässig bleiben müssen.

Zusammenfassend ergibt sich aus den lerntheoretischen Überlegungen, dass die einzelnen Bereiche aus der Netzwerktheorie so ausgewählt und aufbereitet werden, dass diese einerseits repräsentativ für andere Bereiche stehen und andererseits reduziert und vereinfacht werden. Ziel soll aber immer sein, dass die Elemente enaktiv im Unterricht eingesetzt werden können und die Lernenden beim Erarbeiten der einzelnen Bereiche involviert sind.

2.4 Netzwerk-Theorie für Sekundarstufe 1

Hartmann et al. erwähnen schon früh, dass für sie das Thema Kommunikation zu den fundamentalen Themen für den Informatikunterricht gehört, und nennen als Beispiel die synchrone und asynchrone Kommunikation zwischen zwei Geräten (Hartmann, Näf und Reichert 2006).

Modrow und Strecker stimmen zehn Jahre später dieser Meinung zu und schreiben:

«In einer Zeit, wo das Internet so alltäglich, präsent und vor allem gesellschafts-verändernd ist, muss dieses Themengebiet in der Schule aufgegriffen werden.»

Modrow und Strecker 2016

Sie relativieren aber auch, dass viele Bereiche aus der Netzwerktheorie zu komplex für den allgemeinbildenden Unterricht sind, und zählen beispielsweise «Routingverfahren», detaillierte Kenntnisse von Protokollen oder auch das «Schichtenmodell» als ungeeignet für diesen allgemeinen Unterricht auf (Modrow und Strecker 2016).

Zendler und Spannagel haben etwas allgemeiner untersucht, welche Themen den Weg in die verschiedenen Lehrpläne aus dem Bereich der Informatik gefunden haben. Kommunikation und Netzwerke scheinen gemäss ihrer Untersuchung tatsächlich weit verbreitet zu sein. Sie beurteilen die Netzwerk-Theorie aber nicht als Konzept, das in den meisten Unterdisziplinen der Informatik einen Platz findet, aber dort, wo Netzwerke auftauchen, seien diese durchaus relevant (Zendler und Spannagel 2008).

Im neusten deutschschweizerischen Lehrplan (D-EDK 2016) tauchen Netzwerke dann auch auf:

- MI 2.3.j: SuS können lokale Geräte, lokales Netzwerk und das Internet als Speicherorte für private und öffentliche Daten unterscheiden.
- MI 2.3.m: SuS können das Internet als Infrastruktur von seinen Diensten unterscheiden (z.B. WWW, E-Mail, Internettelefonie, Soziale Netzwerke).

Aber auch verbundene Themen aus dem Bereich Netzwerk und Kommunikation finden ihren Platz:

- MI 2.3.b: SuS können sich mit eigenem Login in einem lokalen Netzwerk oder einer Lernumgebung anmelden.
- MI 2.3.n: SuS können die Risiken unverschlüsselter Datenübermittlung und -speicherung abschätzen.

Eine Interpretation dieser vier Kompetenzen ergibt einzig beim letzten Punkt relevante Informatikinhalt aus der Netzwerk-Theorie, wie in der [Tabelle 2.2](#) sichtbar gemacht wird. Beim letzten Punkt (MI 2.3.n) kann beispielsweise angeschaut werden, inwiefern offene Kommunikation in Netzwerken von anderen Netzwerkteilnehmenden mitgelesen werden kann und wie mit einer geeigneten Verschlüsselung das Mitlesen von Inhalten und Daten verhindert werden kann.

2.4.1 Spiralcurriculum

Inhalte aus dem Lehrplan können mit den realen Theorien und der Praxis aus der Industrie verknüpft werden. Die Inhalte werden aber natürlich entsprechend angepasst und heruntergebrochen. Viele Inhalte tauchen nach der Schule im Berufsleben oder im Studium wieder auf und können entsprechend vertieft werden. So bewegen sich in einer optimalen Schulumgebung die Lernenden gemäss Bruner in einer Spirale und lernen Dinge immer wieder und jedes Mal etwas vertiefter (Bruner 1970). In der Praxis ist die Netzwerktheorie sehr komplex und die Theorie dahinter geht schnell in die Tiefe. Gemäss Modrow und Strecker spricht das dafür, das Thema im Unterricht auf Sekundarstufe 1 nicht einfach so zu behandeln.

Im Spiralcurriculum folgt nach der Sekundarstufe 1 für einige Lernenden die Sekundarstufe 2, das Gymnasium. Themen dort können, wenn man Bruner folgt, in einer vereinfachten Form bereits in der Sekundarstufe 1 behandelt werden und es ist sinnvoll, dass die Themen tatsächlich bereits angesprochen werden.

Der Rahmenlehrplan fürs Gymnasium (Schweizerische Konferenz der kantonalen Erziehungsdirektoren 2017) gibt allerdings nur wenig spezifische Inhalte aus dem Bereich «Netzwerk» vor:

- Den modularen Aufbau von Computersystemen aus Komponenten und Schnittstellen kennen
- Die wichtigsten technischen Hintergründe von Computernetzwerken, bzw. des Internets durchschauen

Die Maturitätsschulen erstellen ihren jeweiligen Lehrplan basierend auf diesem Rahmenlehrplan und adressieren Themen wie «einfache Netzwerkarchitektur», «Protokolle» oder «Übermittlung und Adressierung» (Regierungsrat Kanton Glarus 2019).

2.4.2 Mögliche Themen

Dem Spiralcurriculum folgend, können die folgenden drei Themen aus dem Bereich «Netzwerk» für die Sekundarstufe 1 übernommen werden: «einfache Netzwerkarchitektur», «Protokolle» und «Adressierung von Geräten».

Kompetenz	Beschreibung	Interpretation
MI 2.3.j	SuS können lokale Geräte, lokales Netzwerk und das Internet als Speicherorte für private und öffentliche Daten unterscheiden.	Unterscheidung von verschiedenen Speicherorten. Diese können lokal oder online sein. Zugriffe auf einzelne Daten können eingeschränkt werden, gespeicherte Daten können automatisch synchronisiert werden oder kopiert beziehungsweise verschoben werden.
MI 2.3.m	SuS können das Internet als Infrastruktur von seinen Diensten unterscheiden (z.B. WWW, E-Mail, Internettelefonie, Soziale Netzwerke).	Unterscheidung von verschiedenen Applikationen und ihren Anwendungen. Im Browser können Webseiten besucht werden und daraus Informationen herausgesucht werden, in einem Emailprogramm können Emails als Kommunikationsmittel verwendet werden.
MI 2.3.b	SuS können sich mit eigenem Login in einem lokalen Netzwerk oder einer Lernumgebung anmelden.	Wissen um die Geheimhaltung von Passwörtern, Anwendung von Applikationen in Form von Webseiten und Webservices, mit einem Benutzerlogin kann man sich anmelden und abmelden. Verhalten und Schutz im Umgang mit persönlichen Daten.
MI 2.3.n	SuS können die Risiken unverschlüsselter Datenübermittlung und -speicherung abschätzen.	Bei der unverschlüsselten Datenübermittlung und -speicherung können unbefugte Dritte die Daten auslesen oder mitlesen.

TABELLE 2.2: Interpretation einzelner Kompetenzen aus dem Lehrplan (D-EDK 2016).

Einfache Netzwerkarchitektur

Eine einfache Netzwerkarchitektur ist dann gegeben, wenn 2 Geräte direkt miteinander verbunden sind, was aber noch nicht wirklich ein Netz darstellt. Eine andere Variante ist die, dass alle Geräte miteinander verbunden sind und miteinander kommunizieren können. Dabei ist von Vorteil, wenn die Geräte direkt und offen miteinander kommunizieren können.

Protokolle

Mit Hilfe von Protokollen wird der Datenstrom strukturiert, so dass andere Geräte die übermittelten Daten verarbeiten können, da diese einem vorgegebenen Schema folgen. Die Protokolle aus dem Internet definieren meist gleich noch diverse Abfolgen von Nachrichten, um beispielsweise Schlüssel für die Verschlüsselung der nachfolgenden Nachrichten zu übermitteln, damit ist aber bereits eine weitere Komplexitätsstufe erreicht.

Adressierung von Geräten

Wie werden Geräte im Netzwerk erkannt? In heutigen Netzwerken werden die Geräte mit einer individuellen Nummer, die durch den Hersteller vergeben wird, markiert¹⁶. Mit dieser Nummer kann jedes Gerät identifiziert werden. In Netzwerken wird aber oft mit der IP-Adresse gearbeitet, diese wird immer wieder neu verteilt, so dass freie Plätze wieder besetzt werden. Die IP-Adresse ist ebenfalls gemäss einem spezifischen Protokoll aufgebaut und genau definiert. Eine viel einfachere Adressierung wäre beispielsweise eine Durchnummerieren aller Geräte im Netzwerk, entweder numerisch von 0-9 oder alphabetisch von A-Z.

2.4.3 Vergleich mit internationalen Curricula

In der Schweiz sind Netzwerke in den Lehrplänen der Sekundarstufe 1 nur beschränkt zu finden. Stichprobenartig wurden verschiedene Lehrpläne aus verschiedenen Ländern genauer analysiert. Die Auswahl der Länder entspricht den Kriterien «ähnliche kulturelle Hintergründe wie die Schweiz» (Deutschland), deren Lage auf verschiedenen Kontinenten (Australien, Japan, Grossbritannien) oder deren technologische Fortschrittlichkeit (China, Vereinigte Staaten). Ebenfalls wurde berücksichtigt, dass die Lehrpläne online verfügbar sind.

In den untersuchten Ländern sieht es aktuell sehr ähnlich aus, wie die folgende Übersicht aufzeigt. Eine Ausnahme bilden die Vereinigten Staaten.

Australien

Verschiedene Technologien werden dafür eingesetzt, «Computational Thinking» zu erlernen (Australian Curriculum, Assessment and Reporting Authority [ACARA] 2021). Dabei wird nicht genauer definiert, welche Technologien gemeint sind. Netzwerke werden nirgends explizit genannt und die erwähnten Projektideen deuten auf die fachgerechte Anwendung von verschiedenen Technologien hin und nicht auf die Verständnisförderung der Hintergrundtechnologie.

«Investigate and make judgements on how the characteristics and properties of materials, systems, components, tools and equipment can be combined to create designed solution»¹⁷

Australian Curriculum, Assessment and Reporting Authority [ACARA] 2021

China

In China gibt es neben dem Pflichtfach Informatik sechs verschiedene Wahlpflichtfächer, eines davon ist «Netzwerk Grundlagen» (Liu, Li und Han 2021). Das Netzwerk-Modul¹⁸ ist Grundlage für die akademische Abschlussprüfung, die alle Lernenden in China absolvieren.

Deutschland

Am Beispiel vom Lehrplan von Berlin und Brandenburg (Senatsverwaltung Abteilung Bildung Berlin 2015) sind Netzwerke ein Thema, aber relativ ähnlich wie in der Schweiz:

¹⁶Die MAC-Adresse ist jedem Gerät einzeln durch den Hersteller zugewiesen und ist nicht veränderbar.

¹⁷Untersuchen und beurteilen, wie die Merkmale und Eigenschaften von Materialien, Systemen, Bauteilen, Werkzeugen und Geräten kombiniert werden können, um konstruktive Lösungen zu schaffen.

¹⁸An dieser Stelle konnte nicht evaluiert werden, inwiefern der Inhalt des Moduls mit den Beispielen aus dieser Arbeit (unter anderem Kommunikation in Netzwerken verstehen) übereinstimmt.

- das Zusammenwirken von Hardware, Software und Netzwerk anhand der schulischen Computer beschreiben (C Informatik: 2.3.D/E)
- das Zusammenwirken von Hardware, Software und Netzwerk anhand technischer Alltagsgegenstände erläutern (C Informatik: 2.3.F)

Grossbritannien

Ähnlich wie in Australien geht es darum, dass mit Hilfe von Technologien Projekte aus dem Bereich Physical Computing umgesetzt werden (Department for Education 2014). Es wird nicht genauer definiert, welche Technologien verwendet werden, aber beispielhaft werden «Sensoren», «Aktoren» oder «Microcontroller» genannt.

«apply computing and use electronics to embed intelligence in products that respond to inputs and control outputs using programmable components»¹⁹

Department for Education 2014

Japan

In Japan gibt es im Bereich «Information» an der High School zwei Wahlmodule, eines davon müssen die Lernenden besuchen (Liu, Li und Han 2021). Das Modul «Social and Information» beinhaltet unter anderem das Teilmodul «information and communication networks and dissemination»²⁰²¹.

Vereinigte Staaten von Amerika

In den Vereinigten Staaten hat das Thema «Netzwerke» den Weg in den Lehrplan bereits gefunden (K-12 Computer Science Framework Steering Committee 2016). Im «K-12 Computer Science Framework» erhält das Thema «Networks and the Internet» ein eigenes Kapitel und wird von der 2. bis zur 12. Klasse separat berücksichtigt.

«Computers send and receive information based on a set of rules called protocols. Protocols define how messages between computers are structured and sent.»²²

Auszug aus Grade 8 (K-12 Computer Science Framework Steering Committee 2016)

Die Lernenden werden in ihrer schulischen Laufbahn neben den «Protokollen» auch mit den Bereichen «Netzwerktopologie», «Skalierung von Netzwerken», «Senden und Empfangen von Daten», «Netzwerke als Kommunikationsmittel» konfrontiert.

2.4.4 Annahmen aus den Vereinigten Staaten

Im Bereich «Networks and the Internet» bildet der Lehrplan aus den Vereinigten Staaten den folgenden Ablauf von Inhalten im Bereich «Netzwerk-Kommunikation und -Organisation» ab (Auszug aus K-12 Computer Science Framework Steering Committee 2016):

¹⁹Anwendung von Computern und Einsatz von Elektronik zur Erstellung von intelligenten Produkten, die mit programmierbaren Komponenten auf Eingaben reagieren und entsprechend Ausgaben steuern.

²⁰Informationsnetze und Kommunikationsnetzwerke und deren Verbreitung

²¹An dieser Stelle konnte nicht evaluiert werden, inwiefern der Inhalt des Moduls mit den Beispielen aus dieser Arbeit übereinstimmt

²²Computer senden und empfangen Informationen auf der Grundlage einer Reihe von Regeln, die als Protokolle bezeichnet werden. Protokolle legen fest, wie Nachrichten zwischen Computern strukturiert und gesendet werden.

1. am Ende vom **zweiten** Schuljahr

Netzwerke verbinden Menschen mit anderen Menschen und Informationen, Orte, sowie Ideen. Weltweit können Menschen miteinander verbunden werden.

2. am Ende vom **fünften** Schuljahr

Über eine physische Verbindung (Kabel oder Funk) können Informationen versendet und empfangen werden. Einige Verbindungen sind dabei stärker als andere. Informationen werden in kleine Pakete zerteilt, versendet und beim Empfänger wieder zur ganzen Information zusammengesetzt. «Router» und «Switches» helfen dabei, die Pakete an den richtigen Empfänger zu übermitteln.

3. am Ende vom **achten** Schuljahr

Computer senden und empfangen Informationen auf der Grundlage einer Reihe von Regeln, die als Protokolle bezeichnet werden. Protokolle legen fest, wie Nachrichten zwischen Computern strukturiert und gesendet werden. Unter Berücksichtigung von Sicherheit, Geschwindigkeit und Zuverlässigkeit wird der beste Pfad zum Senden und Empfangen von Daten ermittelt.

4. am Ende vom **zwölften** Schuljahr

Die jeweilige Netztopologie bestimmt, wie viele Geräte im Netzwerk unterstützt und erreicht werden können. Jedem Gerät wird eine Adresse zugewiesen, die es im Netz eindeutig identifiziert. Im Internet bestimmen Hierarchie und Redundanz in den Netzen, inwiefern Verbindungen skaliert werden und wie zuverlässig diese erstellt werden können. So kann bestimmt werden, wie gut einzelne Geräte im Internet erreichbar sind.

In der Kategorisierung vom Lehrplan K-12 entspricht bis und mit fünftes Schuljahr der «Elementary School», danach folgt die «Middle School» mit den Schuljahren sechs bis 8 und die «High School» mit den Jahren Neun bis Zwölf. Der Zyklus 3 aus der Schweiz ist folglich vergleichbar mit der «Middle School», die dazugehörigen Inhalte sind die Ziele bis und mit Schuljahr Acht, die ersten drei Punkte aus der vorherigen Liste. Insbesondere der Bereich «Protokolle» überschneidet sich mit den Empfehlungen basierend auf dem Spiralcurriculum aus dem schweizerischen Schulsystem (siehe [Abschnitt 2.4.2](#)).

2.5 Unterrichtsbeispiele zum Thema «Netzwerke» und «Kommunikation»

Die folgende Liste soll Auskunft darüber geben, wie Andere das Thema «Kommunikation» und insbesondere «Netzwerke» im Unterricht behandelt haben und welche Schwerpunkte sie dabei gesetzt haben.

- E-Mail (nur?) für dich! (Gramm, Hornung und Witten 2011)

Über verschiedene Stationen wird in dieser Unterrichtseinheit die Kommunikation in Netzwerken, insbesondere die Kommunikation via Email genauer angeschaut. Dabei wird ein eigenes Protokoll entwickelt, das insbesondere die Übertragung der Daten im Fokus hat, also wie die Signale und Daten übertragen werden. Anschliessend wird der Fokus auf die Verschlüsselung gelegt, so dass die Kommunikation nicht von unberechtigten Personen mitgelesen werden kann. Die Unterrichtseinheit wird am

Computer durchgeführt, dabei sind diverse Programme und Add-Ons²³ zu installieren. Die Computer benötigen einen Internetzugang.

- Filius (Freischlad 2022)

Filius ist eine Lernsoftware für Netzwerke. Mit Filius können Netzwerke digital simuliert werden und die Kommunikation in den jeweiligen Netzwerken getestet und beobachtet werden. Dabei kann ziemlich exakt nachgebaut und eruiert werden, wie die einzelnen Internetprotokolle funktionieren und wie Daten in einem Netzwerk zwischen einzelnen Geräten ausgetauscht werden.

- Wie funktioniert das Internet? (Borowski u. a. 2011)

Kinder - diese Unterrichtseinheit ist für Schülerinnen und Schüler der dritten und vierten Klasse - erleben das Internet und seine Funktionen mit Hilfe von Figuren aus Pappe. Sie sehen die einzelnen Abläufe, beispielsweise von einer Authentifizierung mit Benutzernamen und Passwort. Dabei sehen sie direkt, welche Kommunikationsschritte dabei unternommen werden und welche Geräte aus dem Internet welche Funktion übernehmen.

- Informatik erLernen: Einheit «Netze» (Mittermeir und Bischof 2022)

In dieser Einheit erlernen die Jugendlichen in fünf Schritten, was es für funktionierende Netze alles benötigt:

1. Übertragung ist fehleranfällig, Fehler müssen korrigiert werden
2. Für die Kommunikation braucht es Regeln, sprich Protokolle
3. Für die Übertragung von Gerät zu Gerät benötigt jedes Gerät eine eigene Adresse
4. Die Adressen (3.) und Protokolle (2.) müssen miteinander harmonieren, damit die Daten fließen können
5. Das Versenden von Daten kann mit Hilfe von Algorithmen optimiert werden: Stichwort «Routing»

Die Lernenden erfahren in Planspielen einzelne Funktionen von Netzwerken und sehen, dass diese miteinander funktionieren müssen, damit Daten am gewünschten Ziel eintreffen.

Die verschiedenen Beispiele aus dem schulischen Umfeld zeigen inhaltlich ähnliche Schwerpunkte auf, die bei Netzwerken und Kommunikation im Unterricht wohl zwangsläufig auftauchen. Die Herangehensweise ist sicher nicht die gleiche - «Physical Computing» kommt in keiner der erwähnten Unterrichtseinheit vor - aber die Themen sind relativ ähnlich wie die ausgearbeiteten Themen aus [Abschnitt 2.4](#).

2.6 Fehlvorstellungen

Beim Programmieren wurde bereits mehrfach (beispielsweise Kaczmarczyk u. a. 2010, Kohn 2017 oder Swidan, Hermans und Smit 2018) untersucht, dass Fehlvorstellungen bei Lernenden auftauchen können und dass diese auch mit der richtigen Herangehensweise vermieden werden können. Im Bereich Netzwerktechnologie lassen sich direkt noch keine entsprechenden Untersuchungen finden, dafür wurde bereits untersucht, was für Vorstellungen Schülerinnen und Schüler vom Internet haben (siehe Diethelm und Zumbrägel 2010) sowie

²³Erweiterungen zu bestehenden Programmen

die Vorstellung von einzelnen Funktionen von Mobiltelefonen (siehe Brinda und Braun 2017).

In beiden Studien wurde unter anderem untersucht, wie Daten in einem Netzwerk übertragen werden. Gemäss Brinda und Braun sind sich die Kinder bewusst, dass Smartphones in einem Netzwerk miteinander verknüpft sind. Dass für die Erreichbarkeit eine genaue Adressierung ermöglicht werden muss, darüber waren sich die Kinder nicht mehr einig. Diethelm und Zumbrägel sind zu ähnlichen Resultaten gekommen. Bei ihnen wurde klar, dass einzelne Geräte im Internet eine eindeutige Adresse besitzen müssen, aber nur wenige konnten ein System erklären, das dem der IP-Adressen nahe kommt. Bei der Übertragung von Daten konnten dann nur noch wenige der teilnehmenden Kinder erklären, dass es wohl eine physikalische Verbindung über Funk oder über eine Leitung benötigt. Ähnliche Resultate erhielten auch Brinda und Braun, Daten werden als Signale von Masten (hier nannten die befragten Kinder aber auch Satelliten) zu den Mobiltelefonen übertragen.

2.6.1 Fehlerquellen

Neben den bereits genannten Fehlvorstellungen kann es auch ganz konkrete Ursachen geben, die Fehler bei der Erstellung des eigenen Netzwerkes verursachen können. Fehler können den Lernprozess unterbrechen oder gar fehlleiten, deshalb sollen die Lernenden diese gar nicht erst begehen. Die folgende Auflistung soll helfen, mögliche Fehlerquellen anzugehen und von Beginn an zu verhindern.

Fehlerquelle: Adressierung

Welches Gerät erhält welche Nachricht? Wie kann ein Gerät eine Nachricht so identifizieren, dass die Nachricht für das Gerät bestimmt ist?

In einem offenen Netzwerk mit einem offenen Versand von Nachrichten erhalten alle Geräte alle Nachrichten. Die Lernenden müssen die Nachrichten so verschicken, dass diese bei einem Gerät angezeigt werden und von den anderen Geräten ignoriert werden. Sie können nicht einfach alle Nachrichten verarbeiten oder anzeigen lassen. Diese Selektionierung der Nachrichten ist wichtig, insbesondere wenn tatsächlich alle versendeten Nachrichten beim Gerät ankommen.

Eine präventive Massnahme gegen dieses Problem könnte sein, dass die Identifizierung der Geräte ziemlich früh im Lernprozess erlernt wird und die Lernenden früh im Prozess merken, dass alle Geräte alle Nachrichten erhalten, aber diese nur vereinzelt - immer dann, wenn ihr Gerät angesprochen wird - verarbeiten sollen.

Fehlerquelle: Protokollverarbeitung

Das Protokoll wird vorgeben, wie Nachrichten ergänzt werden müssen, aber auch wie Nachrichten aus einer übermittelten Botschaft herausgelesen werden können. Aus einer übermittelten Datenmenge kann also beispielsweise der Empfänger, der Absender und die eigentliche Nachricht herausgelesen werden. Genauso müssen Nachrichten vor dem Senden mit den entsprechenden Informationen versehen werden. Diese Verarbeitung, das Zusammensetzen und Auseinandernehmen der übermittelten Botschaften, kann Schwierigkeiten verursachen, da technisch im Programm gelöst werden muss, was zuvor miteinander im Team abgesprochen wurde. Je nach Protokoll könnte beispielsweise der erste Buchstaben aus dem übermittelten String dem Absender entsprechen. So muss zu Beginn der Nachricht der Absender als einzelner Buchstabe ergänzt und beim Empfänger entsprechend wieder ausgelesen und entfernt werden.

Eine präventive Massnahme gegen dieses Problem kann sein, dass die Programmierumgebung bei der Verarbeitung von Texten unterstützt. Die konkreten Schritte für die Prävention der möglichen Fehlerquellen folgen in [Abschnitt 3.3](#).

Probleme mit der Parallelität

Beim Problem der Parallelität geht es nicht per se um eine Fehlvorstellung, aber um ein Problem, welches beim Programmieren und Arbeiten mit Computern oft auftauchen kann. Das Programm arbeitet zu langsam, um auf alle Ereignisse zu reagieren. Nachrichten werden stetig im Netzwerk verschickt, ein einfaches Programm wird aber nur eine Nachricht nach der anderen verarbeiten und überprüfen. Soll über ein Netzwerk kommuniziert werden, benötigen wir Eingaben von Usern und müssen Nachrichten den Usern anzeigen. Die Eingabe dauert seine Zeit, in welcher ein Programm warten wird. Die Ausgabe muss eine Zeit lang bestehen - oder gar ewig angezeigt werden, damit der User die Nachricht auch tatsächlich lesen kann.

Probleme mit der Parallelität können verringert werden, in dem beispielsweise Wartelisten im Hintergrund geführt werden und die Geräte die eingehenden Nachrichten speichern und in der eingetroffenen Reihenfolge an Scratch übermitteln. So muss das Scratch-Programm der Lernenden nicht diese Arbeit übernehmen. Scratch bietet zudem einen Parallel-Modus an, so dass die verschiedenen Programme ohne eine individuell programmierte Lösung bereits parallel laufen. Eine Schwierigkeit wird dann aber nach wie vor sein, dass die verschiedenen Programmstränge aufeinander hören und untereinander Ereignisse und Nachrichten austauschen können.

Fehlerquelle: Übertragung der Daten

Kinder können sich nur schwer vorstellen, wie Daten übertragen werden. Insbesondere bei der unsichtbaren Übertragung via Funk, bei der keine sichtbare physische Verbindung zwischen zwei Geräten benutzt wird, ist das auch durchaus verständlich. Ebenso ist bei sichtbaren Kabelverbindungen nicht sichtbar, wie die Daten genau übertragen werden. Sofern möglich, ist es durchaus in Betracht zu ziehen, die Datenübertragung in irgendeiner Form sichtbar zu machen.

2.7 Ausblick auf die Entwicklung

Schülerinnen und Schüler des Zyklus 3 sollen die Netzwerktechnologie aktiv selber erleben können, damit sie verstehen und erlernen, wie die Kommunikation zwischen einzelnen Geräten funktioniert. Das soll mit Hilfe der modernen Netzwerktechnologie «LoRa» in Kombination von Physical Computing in Form von eigenständigen Geräten, die direkt programmiert werden können. Dabei sollen die Lernenden vor komplexe Lernaufgaben gestellt werden, die sie eigenständig lösen können und bei denen sie verschiedene Lösungswege gehen können. Mögliche Inhalte der einzelnen Aufgaben sind die Netzwerkarchitektur, Protokolle sowie die Adressierung von Geräten. Diese Inhalte sollen aber weder klar vorgegeben sein, noch sollen sie Punkt für Punkt abgearbeitet werden. Viel wichtiger ist, dass die Lernenden die Inhalte während den Aufgaben selber herleiten und die Funktionen im Netzwerk direkt erleben können.

Im folgenden [Kapitel 3](#) wird aufgezeigt, wie das geschieht.

Kapitel 3

Entwicklung

3.1 Physical Computing: geeignete Soft- und Hardware

Physical Computing basiert auf dem Einsatz von Microcontrollern, meist Einplatinencomputer oder Kleinstcomputer. Kombiniert mit Sensoren oder Aktoren und im Einsatz mit der richtigen Software können sich damit realitätsnahe Computerumgebungen erstellen lassen. Sensordaten können gemessen und verarbeitet werden, Aktoren können bei Bedarf angesteuert werden. Dazu wird auf der einen Seite Hardware benötigt, die ihren Zweck erfüllt, und Software, die entweder bereits die Bedingungen für den Einsatz erfüllt, oder aber für ihren Einsatz optimiert werden kann. Ebenfalls muss beachtet werden, in welchem Bereich Physical Computing eingesetzt wird: in der Schule, allenfalls draussen, allenfalls bei den Schülerinnen und Schülern zuhause. In diesen Umgebungen, aber auch allgemein beim Einsatz von Physical Computing in der Schule, sollten ein paar Punkte speziell berücksichtigt werden. So können schon von Beginn an Fehlerquellen verhindert und unnötige Frustrationen im Unterricht vermieden werden. Aus anderen Projekten im Bereich vom «Physical Computing» ergeben sich die folgenden Punkte:

- Gleichzeitige Verarbeitung von verschiedenen Eingaben und Ausgaben. Sensoren messen Daten und gleichzeitig muss auf äussere Umstände reagiert werden. Die Hard- und Software muss in der Lage sein, parallel mehrere Dinge zu verarbeiten. (Przybylla 2018).
- Softwarefehler im Programm der Schülerinnen und Schüler verhindern oft ein sauberes Funktionieren des Programms. Fehlererkennung und -vermeidung kann durch Ausprobieren, durch die Kontrolle der Lehrperson, aber auch durch partnerschaftliche Kontrolle sowie Pair Programming²⁴ innerhalb der Zielgruppe erhöht werden (Chakarov u. a. 2021). Mit der richtigen Software können Syntax-Fehler²⁵ verhindert werden, so können solche Probleme beispielsweise bei der Blockprogrammierung nicht auftauchen, da in diesem Fall keine Syntax selber erschaffen werden muss, sondern vom Programm bereits korrekt vorgegeben wird.

²⁴Pair Programming ist eine Methode, um zu zweit an einem Gerät Programme zu schreiben. Dabei programmiert die eine Person, während die andere Person nur kontrolliert, berät und beobachtet.

²⁵Syntax-Fehler sind Fehler in Programmen, die dann auftauchen, wenn die Logik des gewünschten Programms falsch notiert wurde. Beispielsweise können im Programmcode einzelne Satzzeichen falsch eingesetzt werden und das Programm funktioniert deshalb nicht korrekt, obwohl abgesehen von diesen kleinen Flüchtigkeitsfehlern alles einwandfrei wäre. Abgesehen von Syntax-Fehlern gibt es auch bei den visuellen Programmiersprachen einige Misskonzepte, die bei Schüler:innen in verschiedenen regelmässig auftauchenden Fehlerquellen resultieren, wie Sirkkiä und Sorva aufzeigen konnten (Sirkkiä und Sorva 2012)

- Verbindungen von Kabeln sollen möglichst stark sein. Im Projekt von Chakarov et al. haben sich die verwendeten «Alligatoren»-Clips von Zeit zu Zeit gelöst und es kam eine unbekannte und meist unsichtbare Fehlerquelle hinzu, die vermieden werden könnte (Chakarov u. a. 2021).
- Allgemeine Robustheit hilft, dass Geräte nicht gleich einen Defekt aufweisen, nur weil sie kurz unsorgfältig behandelt wurden.

Dazu kommen Erfahrungen aus der vorliegenden Projektarbeit:

- Beim Einsatz von «headless» Raspberry Pi²⁶ muss man sich jeweils über ein bestehendes Netzwerk mit dem Gerät verbinden. Dafür ist in jedem Fall die IP-Adresse vom Gerät nötig. Erstens muss dafür das Gerät im Netzwerk eingebunden sein, andererseits muss die IP-Adresse herausgelesen werden können. Die einfachste Lösung ist die Verbindung via Ethernet, so müssen keine WLAN-Daten angepasst werden. Verbundene Geräte in einem Netzwerk können mit Zugriff über den Router²⁷ einfach ausgelesen werden.

3.1.1 Hardware

Die Theorie zeigt auf, dass Hardware aus den Kategorien «Geräte mit einem Computerchip» oder besser «Eigenständige Geräte» (siehe [Abschnitt 2.2](#)) eingesetzt werden soll.

Bei den eigenständigen Geräten hat sich der «Raspberry Pi» der «Raspberry Pi Foundation» durchgesetzt. Die Stiftung hat sich mit der Entwicklung des Einplatinencomputers unter anderem zum Ziel gesetzt, dass Schulen die Geräte einsetzen und Schülerinnen und Schülern damit ermöglichen, digitale Technologien selber umzusetzen (Raspberry Pi Foundation 2022). Der «Raspberry Pi» ist ähnlich bedienbar wie ein normaler Computer, kann mit einer Desktop-Umgebung versehen werden und mit Peripherie wie Maus und Tastatur ergänzt werden. Das gängige Betriebssystem basiert auf Linux und unterstützt die verschiedenen Programmiersprachen, unter anderem «Python», einer beliebten Programmiersprache für Einsteiger (unter anderem Zuckarelli 2021).

Der «Raspberry Pi» kann mit weiterer Hardware ergänzt werden, als Peripherie kann ein Bildschirm mit Touch-Funktion angeschlossen werden, so dass auf eine Tastatur und eine Maus zur Bedienung verzichtet werden kann. Auf einem Raspberry Pi, ausgerüstet mit Touchscreen, können graphische, browserbasierte Programmierumgebungen installiert und genutzt werden. Mit Linux im Hintergrund können auch alle anderen Programmierumgebungen (textbasiert oder blockbasiert), die für Linux verfügbar sind, eingesetzt werden. Browserbasierte Anwendungen können mit einer digitalen Tastatur über den Browser gesteuert werden, für andere Anwendungen gibt es weitere digitale Tastaturen für die Betriebssystem-Umgebung von Linux. Genauso kann aber auch eine normale Tastatur verwendet werden.

Für den «Raspberry Pi» stellt Adafruit den LoRa-Adapter «Adafruit LoRa Radio Bonnet with OLED» zur Verfügung, der direkt mit dem Raspberry Pi verbunden werden kann. Für die Programmierung via «Python» werden von Adafruit Bibliotheken zur Verfügung gestellt, so dass der Hardware-Adapter direkt angesteuert werden kann und Daten einfacher übers LoRa-Netzwerk versendet und empfangen werden können. Adafruit hat sich zum Ziel gesetzt, solche Erweiterungen zu produzieren, die im schulischen Umfeld für technologische

²⁶«headless» Raspberry Pi bedeutet, dass das Raspberry Pi ohne Bedienungshilfen wie Tastatur oder Maus oder Bildschirm eingesetzt wird. Ein Zugriff ist nur via Netzwerk möglich.

²⁷Das Netzwerkgerät, welches die einzelnen Geräte im Netzwerk miteinander verbindet und Dinge im Netzwerk steuert, wie beispielsweise die Verteilung der individuellen IP-Adressen.

Experimente verwendet werden, aber genauso auch in anderen Bereichen für die Entwicklung von Prototypen verwendet werden können (Adafruit Industries, LLC 2022).

Aus der Kategorie «Geräte mit einem Computerchip» ist der Microcontroller «Arduino» weit verbreitet. Dieser kann über eine eigene Software programmiert werden, dazu muss der «Arduino» mit einem Computer verbunden werden, dieser benötigt die entsprechende Software. Die dabei verwendete Software ist textbasiert. Erweiterungen für blockbasierte Programmierumgebungen wie Scratch (in diesem Fall «Scratch for Arduino») oder Snap (in diesem Fall «Snap for Arduino») sind ebenso möglich. Ein «Arduino» kann wie der «Raspberry Pi» mit diverser Hardware ausgestattet, beispielsweise mit einem Bildschirm. Auch die Erweiterung mit LoRa-fähigen Adaptern ist gegeben, hier kann beispielsweise das «Dragino LoRa Shield» (bastelgarage.ch 2022) verwendet werden. Da der «Arduino» nicht selber gesteuert wird, wie das beim «Raspberry Pi» der Fall ist, benötigen wir den Bildschirm nur für allfällige Text-Ausgaben vor, während und nach dem Versenden von Nachrichten. Um Nachrichten zu übermitteln, müssen wir diese auch dem Gerät übergeben können, dazu ist eine digitale oder analoge Tastatur nötig, die direkt am «Arduino» angeschlossen werden kann.

Für die Programmierung des «Arduino» benötigen wir einen zweiten Computer, der mit dem Gerät verbunden wird. Darüber wird das Programm vorbereitet und anschliessend auf den Microcontroller geladen. Dies hat den Vorteil, dass geräteunabhängig programmiert werden kann und das gewünschte Programm *vorbereitet* werden kann. Ein Nachteil ist, dass man den «Arduino» jeweils zuerst mit dem Computer verbinden muss, um das neue Programm auf dem «Arduino» zu speichern. Anschliessend an die graphische Programmierung wird das erstellte Programm in ein eigenständiges Programm umgewandelt, das nun direkt auf den Arduino geladen wird. Dies geschieht automatisch und bei der Umwandlung der graphischen Blöcke wird auch die entsprechende Kommunikation zwischen dem LoRa-Netzwerk und dem «Arduino» vorbereitet. Eine interne Verknüpfung der Programmierumgebung mit der Hardware, wie sie beim «Raspberry Pi» nötig ist, ist hier überflüssig.

3.1.2 Programmiersprache

Auch im Bereich der Programmiersprache muss entschieden werden, welche Programmiersprache gewählt werden soll. Als erstes gilt es, eine Wahl zu treffen zwischen blockbasierter, visueller Programmierung oder reiner textbasierter Programmierung. Davon ausgehend, dass die Schülerinnen und Schüler wenig oder keine Erfahrung²⁸ mitbringen, ist die Blockprogrammierung die bessere Wahl. Bestehende Lehrmittel für den Zyklus 3 setzen bereits auf die Blockprogrammierung (beispielsweise «connected 4», Fischer u. a. 2021), aber auch Steinbach und Zitzler empfehlen 2017, blockbasiert zu programmieren - und empfehlen gleich die Umgebung «Scratch» (Steinbach und Zitzler 2017). Reine Informatiklehrmittel fürs Gymnasium auf Stufe Zyklus 3 setzen hingegen auf eine textbasierte Programmierung (beispielsweise «einfach Informatik», Hromkovic und Kohn 2018)), finden dafür aber nur mässig Unterstützung, wie Döbeli Honegger von der Pädagogischen Hochschule Schwyz

²⁸Im Zyklus 1 und 2 beinhaltet der Lehrplan 21 (D-EDK 2016) in erster Linie das Kennenlernen von Anweisungen und Programmabläufen, insbesondere dem «Beschreiben», «manuell Ausführen» und «Lesen». Erst gegen Ende kommt das Verständnis, dass «Programme eine Abfolge von vordefinierten Anweisungen» sind und als letzte Kompetenz folgt das «Schreiben und Testen» von Programmen mit Schleifen, bedingten Anwendungen und Parametern. Mit Ausnahme von diesem letzten Punkt sind die Kompetenzen darauf ausgelegt, die Kompetenzen physisch auszuprobieren und nicht digital umzusetzen. Bis zur Oberstufe bringt der Lehrplan nur wenig Programmieren mit sich.

darüber urteilt: «Bei textbasierten Programmiersprachen stellen Syntaxfehler eine erste Einstiegshürde dar, die blockbasierte Sprachen verhindern²⁹» (Döbeli Honegger 2021).

Für die blockbasierte Programmierung haben sich ein paar wenige Programmierumgebungen durchgesetzt:

- Scratch (Resnick u. a. 2009)
- Snap! (Harvey und Mönig 2010)
- Blockly (Fraser 2015)
- Raspblocks³⁰
- Microblocks (Cabrera, Maloney und Weintrop 2019)
- MakeCode (Devine u. a. 2018)

In [Tabelle 3.1](#) findet sich eine Übersicht, wie die Programmierumgebungen allgemein hinsichtlich dem Einsatz im Unterricht eingeschätzt werden, und insbesondere, ob damit eine Verbindung mit der Hardware aus dem vorangehenden Teil [Unterabschnitt 3.1.1](#) möglich ist. Diese Verbindung mit dem LoRa-Node ist essentiell und kann nur aufgebaut werden, wenn neue Blöcke dem System hinzugefügt werden können. Diese neuen Blöcke sollen so erstellt werden, dass im Hintergrund eine Verbindung zwischen dem Computer und dem LoRa-Node aufgebaut wird und im Vordergrund Daten aufgenommen werden können, welche vom jeweiligen Block an die LoRa-Node weitergeleitet werden. Alternativ könnte ein bestehender Block verwendet werden, der bereits eine passende Konfiguration für die Kommunikation zwischen Computer und LoRa-Node beinhaltet, entsprechende Blöcke sind aber zu diesem Zeitpunkt nicht vorhanden.

²⁹Dabei ist anzumerken, dass es spezifische textbasierte Programmiersprachen für Einsteiger gibt, die solche Syntaxfehler reduzieren möchten, in dem möglichst viele spezifische Syntax gar nicht benötigt wird. Im Falle von «einfach Informatik» ist die empfohlene Sprache «tigerJython», die zwar nicht ohne Syntax auskommt, aber die grundlegenden Syntax-Fehler kennt und Einsteigerinnen und Einsteiger auf solche Fehler hinweist.

³⁰Raspblocks ist eine wenig verbreitete Programmierumgebung, ist aber insbesondere erwähnenswert, da sie spezifisch für die Blockprogrammierung von «Raspberry Pi»'s erschaffen wurde.

Name	Link	Qualität ¹	Verbindung Raspberry Pi	Eigene Blöcke	Kommentar
Scratch 2.0	scratchx.org/#extensions	4	Ja: ScratchX	Ja	veraltete Version, ersetzt
Scratch 3.0	github.com/LLK/scratchvm/blob/develop/docs/extensions.md	5	Entweder via Desktop-Version oder als Web-Anwendung im Browser	Ja	Browserbasiert: JavaScript & Websockets
Snap!	snap.berkeley.edu/snap/snap.html	5	Gemäss User-Forum machbar ²	Ja	Browserbasiert: JavaScript & Websockets.
Blockly	developers.google.com/blockly	4	Keine Live-Verbindung, Python-Code muss exportiert & importiert werden	Ja	Code muss separat verarbeitet werden.
Raspblocks	raspblocks.com/coding/demos/code/coding.html	3	Keine Live-Verbindung, Python-Code muss exportiert & importiert werden	nicht möglich	Code muss separat verarbeitet werden.
Microblocks	microblocks.fun	5	Nur Raspberry Pi Zero via separater Plattform (PlatformIO) möglich	Ja	Unklar, ob das via Websockets möglich wäre.
micro:bit, MakeCode	makecode.microbit.org	-	Keine Verbindung mit Raspberry Pi möglich.	wäre möglich	Keine passenden LoRa-Nodes für micro:bit bekannt.

TABELLE 3.1: Übersicht über die verschiedenen analysierten Umgebungen für die Blockprogrammierung.

¹Eigene Einschätzung mit Punkten zwischen 1 (schlecht) und 6 (super) aufgrund von Aussehen, Handling, Programmierung, Komplexität und Dokumentation²Eintrag im User-Forum: forum.snap.berkeley.edu/t/raspberry-pi-gpio-snap-extension/2305/23

Die beiden verbreiteten Programmiersprachen «Snap!» und «Scratch» sind beides browserbasierte Software-Anwendungen, die abgesehen von den oberflächlichen Browserfunktionen keine enge Verbindung mit der Gerätehardware herstellen können. Andere visuelle Programmiersprachen wie beispielsweise «Raspblocks» arbeiten gar nicht in Echtzeit, das heisst, bei diesem Programm wird zuerst das gewünschte Programm aufgezeichnet, dann wird dieses kompiliert und in ein ausführbares Programm-Script umgewandelt. Dieses Script kann anschliessend direkt auf der gewünschten Hardware gestartet werden. Um ein interaktives Erlebnis zu ermöglichen, soll das Programm interaktiv funktionieren und immer direkt, sprich ohne Umwege, getestet beziehungsweise gestartet werden können. Aus der Liste [Tabelle 3.1](#) bleiben so nur noch «Scratch» und «Snap!» in der engeren Auswahl.

Beide Blockprogrammierungen sind als browserbasierte Applikationen nicht direkt mit der Hardware verbunden. Die Kommunikation mit Hilfe von einem Netzwerk-Adapter (LoRaWAN oder ähnliches) wird aber direkt via Hardware gesteuert und mit Hilfe der GPIO-Kontakten³¹ auf dem jeweiligen Microcontroller angesteuert. Damit das funktioniert, wird eine Schnittstelle zwischen Hardware und Browser eingesetzt: eine auf «WebSocket»³² basierende Verbindung. Die Verbindung soll ermöglichen, dass die webbasierte Applikation als «Client» via «WebSocket»-Protokoll direkt mit dem jeweiligen Gerät als «Host» kommunizieren kann und die Webapplikation «Scratch» beziehungsweise «Snap!» Daten direkt an den Host übermitteln können, so dass diese zu einem Netzwerksignal umgewandelt werden können.

Wenn beide Programmiersprachen mit einer ähnlichen Lösung eingesetzt werden können, stellt sich die Frage, welche nun ausgewählt werden soll. Die Popularität spricht hier klar für «Scratch», welche bis im November 2021 bereits über 80 Millionen Personen³³ zu einer Nutzung der Programmiersprache registrieren konnte. «Snap!» hingegen verzeichnete bis im März 2021 nur gerade eine halbe Million Nutzerinnen und Nutzer³⁴. Von der Grösse lässt sich auf die Popularität schliessen, aber auch auf die Verbreitung und den allgemeinen Bekanntheitsgrad. Je verbreiteter und bekannter die gewählte Anwendung, desto eher können wir darauf schliessen, dass Kinder damit umgehen können, da die Anwendung bereits bekannt ist, oder aber auch dass die geplante Unterrichtseinheit erfolgreich eingesetzt werden kann, da die Anwendungen per se bereits bekannt sind.

Beide Programmierumgebungen können individuell erweitert werden, eignen sich also für die Verknüpfung mit einer eigenen «WebSocket»-Umgebung.

³¹General Purpose Input/Output sind die Standardkontakte auf einem Raspberry Pi, um weitere Aktoren oder Sensoren mit dem Controller zu verbinden.

³²Das WebSocket-Protokoll ist eine bidirektionale Verbindung zwischen einem Web-Client (also zum Beispiel einer Browser-Anwendung) und einem Host, beispielsweise einem Server. Die Verbindung basiert auf dem TCP-Protokoll und ermöglicht eine einfache Zwei-Weg-Kommunikation, bei der einmalig die Verbindung aufgebaut wird und anschliessend beliebig oft Daten ausgetauscht werden können (Fette und Melnikov 2011).

³³Statistik von Scratch: 80'976'811 registrierte Nutzerinnen und Nutzer mit 90'103'505 geteilten Projekten [Stand: 23. November 2021, abgerufen von scratch.it.edu/statistics]

³⁴Auskunft von Snap-Entwickler «Jens» im Snap-Forum, dass bis zum 18. März 2021 511'000 Nutzer im System zu finden sind [Stand: 23. November 2021, abgerufen von forum.snap.berkeley.edu/t/how-many-snap-users-are-there/5598/8]

3.1.3 Auswahl Soft- und Hardware

Hardware

Als Grundlage entscheiden wir uns für einen «Raspberry Pi», den wir mit den nötigen Adaptern und Erweiterungen ausstatten. Dazu gehört ein Touchscreen für die Bedienung sowie der LoRa-Adapter von Adafruit.

Die Entscheidung fiel auf den «Raspberry Pi», da hier eine direkte Programmierung auf dem Gerät möglich ist. So ist einerseits schnell sichtbar, was direkt auf dem Gerät passiert. Auf der anderen Seite werden mögliche Verbindungsprobleme zwischen Programmiercomputer und Gerät direkt vermieden. Der «Arduino» hingegen ist immer nur *ausführendes* Gerät, die Programmierung erfolgt auf einem zweiten Gerät, das in den Schulzimmern zwar sicher vorhanden wäre (Barras und Petko 2007), aber noch entsprechend eingerichtet werden müsste. Mit der direkten Programmierung auf dem jeweiligen «Raspberry Pi» ist keine weitere Hardware nötig und die Geräte können direkt im Unterricht eingesetzt werden.

Die komplette Liste der eingesetzten Hardware ist unter [Anhang A](#) aufgelistet.

Software

Von den beiden noch zur Auswahl stehenden Programmiersprachen wird für die Umsetzung der finalen Benutzeroberfläche auf die visuelle, blockbasierte Programmiersprache «Scratch» gesetzt.

Die Verknüpfung zwischen «Scratch» und dem «Raspberry Pi» wird mit Hilfe eines vorbereiteten «WebSocket»-Servers hergestellt, welcher auf der Programmiersprache «Python» basiert. Für die finale Unterrichtseinheit soll aber nur die Programmierumgebung «Scratch» mit den nötigen Erweiterungen im Vordergrund stehen, alles weitere soll automatisch im Hintergrund geschehen.

Für die Verknüpfung zwischen dem «WebSocket»-Server und «Scratch» werden auf beiden Seiten Schnittstellen erstellt, damit Signale in beide Richtungen übertragen werden können. Der Server kann die eintreffenden Signale ins LoRa-Netz übermitteln und ankommende Nachrichten aus dem LoRa-Netz an Scratch weiterleiten. In Scratch müssen dafür eigenen Blöcke erstellt werden, beispielsweise ein «Sende»-Block, der Nachrichten an den «WebSocket»-Server leitet. Für die andere Richtung braucht es einen «Empfangen»-Block, der jeweils aktiviert wird, sobald eine LoRa-Nachricht eintrifft. Dafür wird dann auch zusätzlich ein «Nachrichten»-Block nötig sein, der die jeweilige Nachricht in «Scratch» auch ausgeben kann.

Eine solche Erweiterung von «Scratch» mit eigenen Blöcken ist möglich. «Scratch» ist eine offene Softwarelösung, die hauptsächlich vom Scratch-Team am MIT³⁵ entwickelt wurde und laufend erweitert wird. Der Quellcode der einzelnen Applikations-Teile ist öffentlich und

³⁵Massachusetts Institute of Technology

kann unter bestimmten Voraussetzungen³⁶ selber verwendet und weiterentwickelt werden. Die einzelnen Software-Teile sind unter Github³⁷ veröffentlicht. Mit den Applikationen «scratch-vm», das die virtuelle Maschine für den Betrieb der Webapplikation zur Verfügung stellt, und «scratch-gui», das die browserbasierte Nutzeroberfläche erstellt, kann die Grundfunktion von Scratch selber erstellt und beliebig erweitert³⁸ werden.

Die komplette Liste der verwendeten Software ist unter [Anhang B](#) aufgelistet.

³⁶Die Software wurde unter der «BSD 3-Clause License» veröffentlicht. Die Software darf privat oder kommerziell veröffentlicht werden, angepasst werden und weiterverbreitet werden, sofern die folgenden drei Bedingungen erfüllt sind:

- Die neue Software (Quellcode) muss unter der gleichen Lizenz verbreitet werden
- Das neue Programm muss unter der gleichen Lizenz verbreitet werden
- Die Hersteller (Name, Firma, weitere an der Software beteiligte Personen) dürfen nicht genutzt werden, um für die neue Applikation zu werben.

³⁷github.com/llk

³⁸Die Scratch-Anwendung dieser vorliegenden Arbeit mit den hier beschriebenen Anpassungen sind unter github.com/btemperli/scratch-raspberry veröffentlicht.

3.2 Kommunikation mit Hilfe von Blockprogrammierung

Die Verbindung zwischen den einzelnen Usern (*User Input / Output*) und dem LoRa-Signal im Netzwerk (*LoRa Input / Output*) wird in Abbildung 3.1 aufgezeigt.

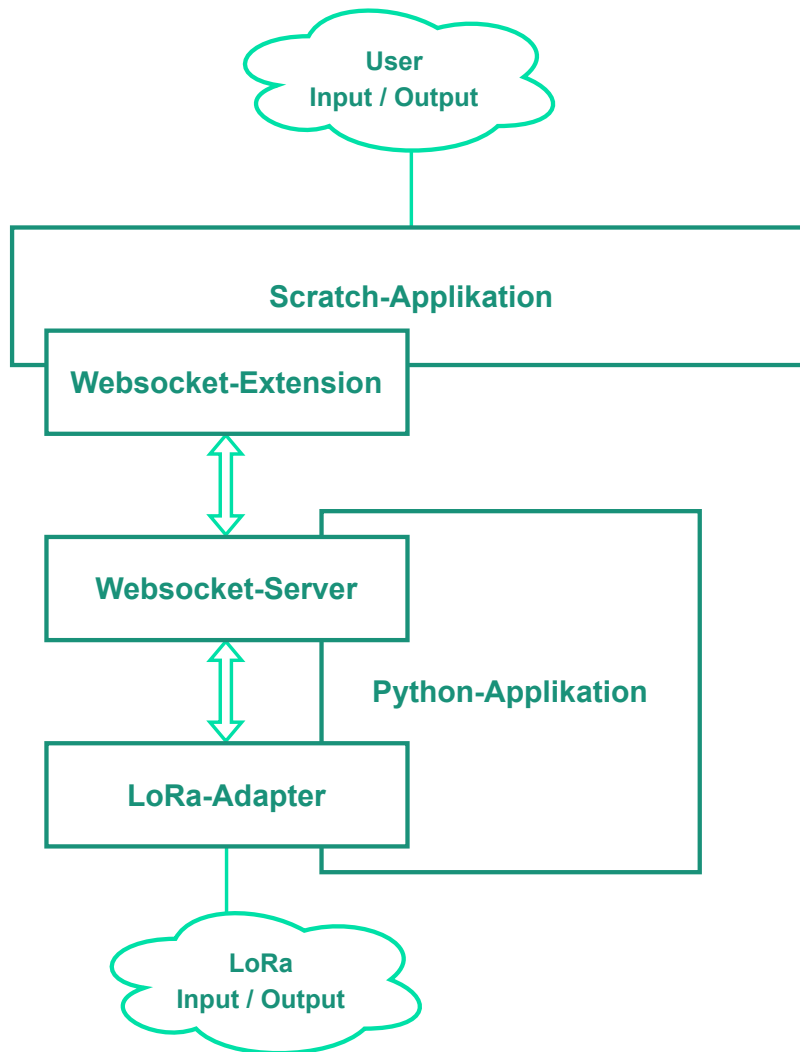


ABBILDUNG 3.1: Schema Verbindung zwischen User-Input/Output ↔ Scratch ↔ Python ↔ LoRa-Input/Output Quelle: Eigene Darstellung

Die Handlungen vom User geschehen direkt in der Scratch-Applikation auf dem Bildschirm des physischen Moduls. Die Scratch-Applikation ist mit einer «WebSocket»-Extension ausgestattet, so dass eine bidirektionale «WebSocket»-Verbindung mit dem «WebSocket»-Server hergestellt werden kann. Diese «WebSocket»-Server-Applikation ist Teil der Python-Applikation, die auch die Kommunikation mit dem LoRa-Adapter³⁹ übernimmt.

³⁹Adafruit LoRa Radio Bonnet mit OLED-Bildschirm

3.2.1 Kommunikation zwischen Scratch und Python

Anhand von zwei Beispielen soll aufgezeigt werden, wie die Kommunikation zwischen der Scratch-Oberfläche (*Blockprogrammierung*) und der Python-Umgebung mit der Anbindung an den LoRa-Sender/-Empfänger funktioniert.

Scratch zu LoRa

Mit Hilfe von einem eigenen Scratch-Block⁴⁰ kann eine beliebige Eingabe⁴¹ an die eigene Scratch-Erweiterung übergeben werden.



ABBILDUNG 3.2: Leerer Scratch-Block für die Übergabe einer beliebigen Eingabe an die LoRa-Erweiterung Quelle: Eigene Darstellung

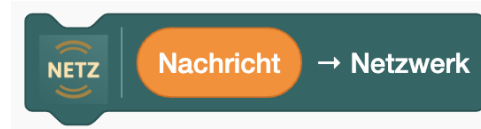


ABBILDUNG 3.3: Scratch-Block für die Übergabe einer «Eingabe» an die LoRa-Erweiterung Quelle: Eigene Darstellung

Damit in der Scratch-Oberfläche ein eigener Block genutzt werden kann, muss dieser in einer eigenen Erweiterung definiert und geladen werden.

```

1 return {
2   id: 'raspylora',
3   blocks: [
4     {
5       opcode: 'messageSendLoRa',
6       blockType: BlockType.COMMAND,
7       text: '[MESSAGE] Network',
8       arguments: {
9         MESSAGE: {
10          type: ArgumentType.STRING,
11          defaultValue: 'text'
12        }
13      }
14    }
15  ]
16 }

```

CODE-AUSSCHNITT 3.1: Definition für einen Scratch-Block für die Übergabe einer beliebigen Eingabe, in diesem Fall vom Typ `String`

⁴⁰«Command» Block aus Scratch

⁴¹Zahl, String, Liste

Dieser einzelne «Scratch»-Block verarbeitet beim Aufruf im Hintergrund die Eingabe: Zuerst wird überprüft, ob eine Verbindung zum «WebSocket-Host»⁴² besteht, wenn nötig wird diese aufgebaut. Sofern die Verbindung vorhanden ist, wird ein JSON⁴³-Objekt an die Python-Applikation übermittelt.

```

1 function messageSendLoRa (args) {
2   if (!connected) {
3     if (!connectionPending) {
4       this.connect();
5       connectionPending = true;
6     }
7   }
8
9   // run the "send" command
10  if (connected) {
11    const n = args.MESSAGE;
12    msg = {send: message.toString()};
13    msg = JSON.stringify(msg);
14    window.socketr.send(msg);
15  } else {
16    const callbackEntry = [this.numberWrite.bind(this), args];
17    wait_open.push(callbackEntry);
18  }
19 }

```

CODE-AUSSCHNITT 3.2: Ausschnitt aus dem Javascript-Code: JSON-Objekt wird an die Python-Applikation gesendet

Die Kommunikation zwischen der «Scratch»-Erweiterung und dem «WebSocket»-Server in der Python-Applikation geschieht via JSON-Objekten.

```

1 {
2   "send": "hi"
3 }

```

CODE-AUSSCHNITT 3.3: Ein mögliches JSON-Objekt mit vordefinierten Schlüssel `send` und dem Inhalt `hi`

Die Python-Applikation reagiert auf das erhaltene Objekt und erkennt anhand des Schlüssels `send`, dass die enthaltene Nachricht via «LoRa»-Adapter ins Netzwerk verschickt wird.

```

1 class LoRa:
2   def __init__(self, logger):
3     self.environment = os.environ.get('environment')
4     self.available_lora = self.environment == 'raspberrry'
5     if self.available_lora:
6       # Configure LoRa Radio
7       CS = DigitalInOut(board.CE1)
8       RESET = DigitalInOut(board.D25)
9       spi = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)

```

⁴²In diesem Fall stellt der Python-Server eine «WebSocket»-Umgebung bereit und fungiert als «Host», mit dem sich andere Geräte verbinden können. Die «Scratch»-Umgebung im Browser kann anschliessend eine «WebSocket»-Verbindung zum Server oder eben «Host» herstellen.

⁴³JavaScript Object Notation ist eine weit verbreitete Möglichkeit, Datenobjekte via Programmierschnittstelle zu übertragen.

```

10     self.rfm9x = adafruit_rfm9x.RFM9x(spi, CS, RESET, 868.1)
11     self.rfm9x.tx_power = 23
12
13     def send(self, message):
14         message_data = bytes(message, encoding="utf-8")
15
16         if not self.available_lora:
17             self.logger.print('send text: ' + message)
18             self.logger.print('send bytes: ' + str(message_data))
19             return
20
21         self.rfm9x.send(message_data)
22
23     class Server:
24     def handle_json_message(self, json_message):
25         try:
26             if "send" in json_message:
27                 send_message = json_message['send']
28                 self.lora.send(send_message)
29         except:
30             self.log.print('problems with json input...')
31             self.log.print(traceback.format_exc())

```

CODE-AUSSCHNITT 3.4: Der «WebSocket»-Server verarbeitet die eingehenden JSON-Objekte

LoRa-Nachricht zu Scratch

Nachrichten, die über das LoRa-Netz beim Endgerät ankommen, werden dort via Server gesammelt. Der Server speichert dabei alle eingehenden Nachrichten in einer Liste ab und entfernt die doppelten Einträge in der Liste⁴⁴.

```

1 def add_lora_message(self, message):
2     self.received_messages.append(message)
3     # remove duplicates from received message:
4     self.received_messages = list(dict.fromkeys(self.received_messages))
5
6 # Read from the LoRa-network
7 # ---
8 # Saves all the messages to a local array.
9 def read_from_lora(self):
10    packet = self.rfm9x.receive()
11
12    if packet is None:
13        # do nothing, no message read.
14        return
15
16    # @see Watcher: def handle_packet(self, packet)
17    packet_text = str(packet, "utf-8")
18    pattern = r'\[[a-z0-9:]*\]'
19    matches = re.match(pattern, packet_text)
20

```

⁴⁴Die Tests der Unterrichtseinheit haben ergeben, dass die Lernenden beim Ausprobieren sehr oft die gleiche Nachricht übermitteln. Scratch muss nun all diese Nachrichten verarbeiten, dies hat allerdings keinen Mehrwert und bedeutet nur, dass beim Programmieren viel Zeit verstreicht, während die uninteressanten Test-Nachrichten überprüft werden. In einer neuen Version der Server-Software werden deshalb doppelte Nachrichten eliminiert, so dass nur eine einzelne zur Verarbeitung übrig bleibt. Werden gleiche Nachrichten mehrfach empfangen und jeweils direkt verarbeitet, fällt die Eliminierung entsprechend gar nicht auf.

```

21     # Address is available in the packet
22     if matches:
23         packet_text = re.sub(pattern, '', packet_text)
24
25         self.prev_packet = packet_text
26         self.add_lora_message(packet_text)
27         return
28
29 # Get the latest message coming from the network.
30 def get_latest_message(self):
31     if len(self.received_messages):
32         message = self.received_messages[0]
33         print('received message: ' + message)
34         self.received_messages.pop(0)
35         return message
36     else:
37         return None
38
39 # Check the LoRa-Net for incoming messages
40 async def broadcast_lora_message(self):
41     while True:
42         message = self.lora.get_latest_message()
43         if message:
44             for c in self.websocket_clients:
45                 await c.send(message)
46         await asyncio.sleep(1)

```

CODE-AUSSCHNITT 3.5: Der «WebSocket»-Server speichert alle eingehenden Nachrichten ab

Via «Scratch» kann man nun die älteste Nachricht aus der Liste vom «WebSocket»-Server abfragen. Dies geschieht mit dem Reporter-Block `LoRaMessage`. In den folgenden Code-Zeilen finden sich die Javascript-Anweisungen für den Block, für den Ablauf im Block, da wird jeweils die letzte Nachricht vom Server, falls diese existiert, zurückgegeben, ansonsten wird eine Nachricht in «Scratch» angezeigt, dass aktuell keine Nachrichten verfügbar sind. In einem zweiten Teil reagiert «Scratch» auf ankommende Nachrichten von der «WebSocket»-Verbindung, in dem die Nachrichten jeweils in der Liste `lastMessages` zwischengespeichert werden.

```

1  loraMessageReporter (args) {
2      let receivedMessage = '';
3
4      if (lastMessages.length) {
5          receivedMessage = lastMessages.shift();
6      } else {
7          receivedMessage = 'No Message received';
8      }
9
10     return receivedMessage;
11 };
12
13 // reporter messages from the board
14 window.socketr.onmessage = function (message) {
15     msg = null;
16     try {
17         msg = JSON.parse(message.data);
18     } catch (error) {
19         try {
20             // try without parsing.
21             msg = message.data;

```

```

22     } catch (error) {
23         log.info(error);
24     }
25 }
26
27 lastMessages.push(msg);
28 lastMessageReceived = true;
29 log.info(msg);
30 };

```

CODE-AUSSCHNITT 3.6: Funktion vom Reporter-Block `LoRaMessage` und dem Eintreffen von neuen Nachrichten.

3.2.2 1:1-Kommunikation

Damit eine Kommunikation zwischen zwei Geräten stattfinden kann, müssen die Nachrichten so gekennzeichnet werden, dass diese von einem Sender an einen Empfänger übermittelt werden. Die Kommunikation erfordert meist eine Antwort, deshalb muss der Empfänger wissen, von wem er die Nachricht erhalten hat. Viel wichtiger in einem Netzwerk aus verschiedenen Geräten ist aber, dass ein einzelnes Gerät weiss, an wen eine eingehende Nachricht gerichtet ist. Nur so weiss ein Empfänger, dass er die Nachricht auch tatsächlich erhalten sollte, insbesondere wenn in einem offenen Netzwerk mit mehreren Netzwerkteilnehmenden Nachrichten offen verschickt werden und jedes Gerät die Nachrichten empfangen kann.

Nachrichten sollen also mit dem Absender und dem Empfänger gekennzeichnet werden. Dafür reicht ein einfaches Protokoll. Dabei werden die zu übermittelnden Botschaften wie folgt aufgebaut:

«Empfänger» «Trennzeichen» «Absender» «Trennzeichen» «Nachricht»

Beispiel: B-A-Hallo!

Wie in «Scratch» eine Botschaft zusammengesetzt werden kann, zeigt [Abbildung 3.4](#). Dabei müssen wir den Absender beispielsweise beim Bediener abfragen und speichern diesen in einem «Variablen-Block» ab.

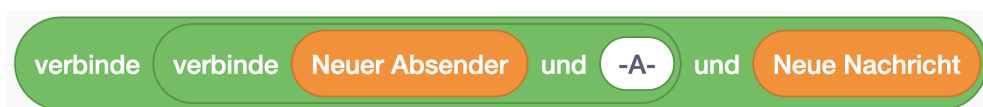


ABBILDUNG 3.4: Scratch-Block für die Verbindung von einer ganzen Botschaft
Quelle: Eigene Darstellung

Damit in «Scratch» nun Botschaften an das eigene Gerät ausgelesen werden können, muss in zwei Schritten vorgegangen werden. Zuerst muss überprüft werden, ob die Botschaft an das Empfänger-Gerät `A` (im Beispiel ist das Gerät mit dem Empfänger-Buchstaben «A» erreichbar) gerichtet ist. Dabei wird schlicht das erste Zeichen abgefragt und überprüft, ob es einem `A` entspricht. Diese Abfrage zeigt [Abbildung 3.5](#).

In einem zweiten Schritt wird die Nachricht aus der Botschaft herausgelöst und dem User angezeigt. Die Anzeige der Nachricht ist in der [Abbildung 3.6](#) damit verbunden, dass der User direkt eine Antwort eingeben kann und diese Antwort zurückgesendet wird.

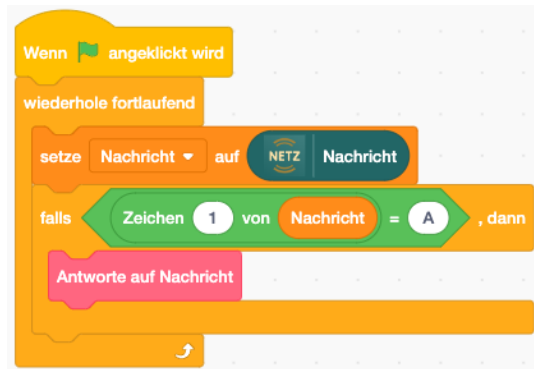


ABBILDUNG 3.5: «Scratch»-Programm zur Überprüfung, ob eine Nachricht an Gerät A gerichtet ist Quelle: Eigene Darstellung

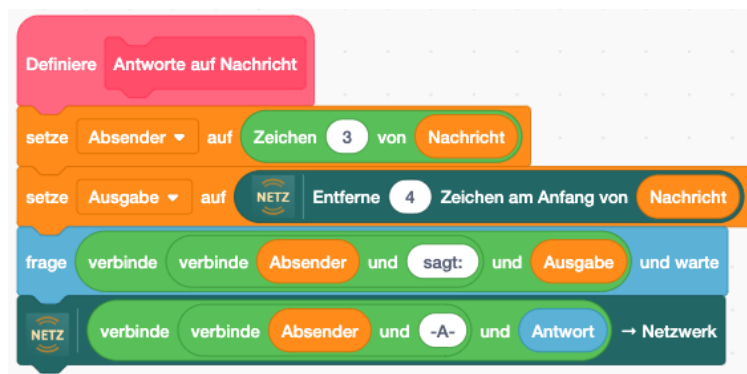


ABBILDUNG 3.6: «Scratch»-Programm, das eine Nachricht ausgibt und eine Antwort erfasst Quelle: Eigene Darstellung

Zu beachten gilt es nur, dass das Trennzeichen nicht in der Nachricht vorkommt. Separiert muss definiert werden, wie die Geräte identifiziert werden, sprich wie der «Absender» beziehungsweise «Empfänger» gekennzeichnet wird. Die Geräte können entweder numerisch von 0 bis 9 gekennzeichnet werden oder alphabetisch mit Buchstaben adressiert werden. In diesem Fall braucht es eine Absprache beim Programmieren, welche Nummer beziehungsweise welcher Buchstabe welchem Gerät zugeordnet wird. Solange die Adressierung einstellig ist, kann das Protokoll auch ohne Trennzeichen auskommen: das erste Zeichen in der Botschaft kennzeichnet den Absender, das zweite Zeichen kennzeichnet den Empfänger, ab dem dritten Zeichen beginnt die Nachricht. Wenn die Adressierung mehrstellig ist, benötigt man das Trennzeichen, um die Adressierung auszulesen.

3.2.3 Erweiterungen

Nach der «1:1-Kommunikation» und der Kommunikation von mehreren Geräten mit Hilfe des erarbeiteten Protokolls sind weitere mögliche Entwicklungen angedacht und teilweise bereits vorbereitet.

Verschlüsselung

Als Vorbereitung für allfällige weitere Schritte in der Netzwerkkommunikation, soll es möglich sein, Nachrichten verschlüsselt zu übermitteln⁴⁵. So kann verhindert werden, dass

⁴⁵Nachrichten im LoRa-Netzwerk werden grundsätzlich verschlüsselt übertragen, insbesondere in den bekannten LoRaWAN Anwendungen wie «The Things Network». In der erarbeiteten Umgebung müssen die

die anderen Geräte im Netzwerk die Kommunikation mitlesen. Die Verschlüsselung passiert in der «Scratch»-Umgebung, mit Hilfe von einem Passwort kann ein beliebiger String mit dem AES-Standard (siehe auch [Abschnitt 2.1.2](#)) verschlüsselt werden. Die verschlüsselte Nachricht kann anschliessend via LoRa-Netzwerk versendet werden. Hierbei handelt es sich um eine symmetrische Verschlüsselung, das heisst das Passwort muss auf anderem Weg übertragen werden.

Routing

Mit Hilfe der «Scratch»-Geräte und einem funktionierenden Netzwerk lässt sich nicht nur kommunizieren, sondern auch weitere Daten übertragen. Dafür ist oft eine grössere Distanz nötig, beispielsweise können in der Umgebung vom Schulhaus stunden- oder tageweise Temperaturdaten gesammelt werden. Dafür müssen die Geräte in der Lage sein, über weite Distanzen zu kommunizieren. Mit den «LoRa»-Sendern lassen sich in der Theorie Mitteilungen über mehrere Kilometer übertragen. In der Realität entscheiden viele Faktoren mit, über welche Distanz tatsächlich eine Verbindung hergestellt werden kann. Die Distanz kann aber mit den eigenen Geräten erweitert werden, in dem ein Routing implementiert wird: Die verschiedenen Geräte kommunizieren nicht nur untereinander, sie leiten auch Nachrichten weiter. Wenn Gerät **A** eine Nachricht an Gerät **C** senden möchte, **C** aber nicht erreichbar ist, kann das Gerät **A** einen Aufruf machen, dass doch andere Geräte versuchen, Gerät **C** zu erreichen. Falls Gerät **B** zwischen den beiden Geräten liegt und *vermitteln* kann, spricht die Nachrichten von **A** an **B** weiterzuleiten und umgekehrt, ist man in der Lage, Mitteilungen von Gerät **A** via **B** nach **C** zu übertragen. Dieses *vermitteln* oder eben Routing sollte mit den vorbereiteten Geräten umgesetzt werden können.

Netzwerkaufbau

In einem Netzwerk können weitere Themen angeschnitten werden, die mit Hilfe der entwickelten Geräte umgesetzt werden können. Beim Netzwerkaufbau braucht es im Netzwerk ein zentrales Gerät, das zusätzliche Funktionen übernimmt und als «Router» die Datenströme zwischen den einzelnen Geräten im Netzwerk leitet. Dafür müssen weitere Funktionen eingebaut werden, auf Seiten der einzelnen Geräte im Netzwerk und auch auf dem zentralen Gerät, das als Router fungiert. Die einzelnen Blöcke für die jeweiligen Funktionen wurden noch nicht implementiert, die folgenden Aufgabenteile wurden aber bereits angedacht.

Kommunikation im Netzwerk bedingt, dass die einzelnen Geräte im Netzwerk unter einer bestimmten Adresse erreichbar sind. Die Adressverteilung im Netzwerk erfolgt normalerweise automatisch über den Router. Dabei müssen sich neue Geräte im Netzwerk identifizieren und beim Router anmelden. Dabei geben sie ihre fixe MAC-Adresse preis und erhalten eine IP-Adresse, über welche sie im Netzwerk erreicht werden können. Der Router sorgt dafür, dass die IP-Adresse eindeutig ist, sprich jedes Gerät erhält eine andere Adresse und verknüpft gleichzeitig die jeweiligen MAC-Adressen mit den neuen IP-Adressen. So weiss der Router stets, welche Geräte im Netzwerk sind und wie diese erreicht werden können. Geräte aus dem Netzwerk können beim Router anfragen, welche Geräte unter einer bestimmten IP-Adresse erreichbar sind oder ob ein bestimmtes Gerät, das mit einer MAC-Adresse identifiziert wird, sich im Netzwerk befindet. Geräte im Netzwerk haben auch die Möglichkeit, einen Aufruf mit der Suche nach einer IP-Adresse zu starten. Das Gerät mit der bestimmten IP-Adresse

Nachrichten aber spätestens bei der Anzeige für die weiteren Geräte im Netzwerk wieder entschlüsselt werden, so dass - auch wenn die Übertragung verschlüsselt wäre - trotzdem alle Geräte mitlesen können. In der simplen Übertragung via die vorbereiteten Python-Server, die auf den jeweiligen Endgeräten installiert wurden, werden die Nachrichten nicht verschlüsselt.

sollte sich nun beim suchenden Gerät zurückmelden. So finden die Geräte auch ohne den Umweg über den «Router» heraus, ob ein bestimmtes Gerät erreichbar ist.

3.2.4 Weitere Scratch-Blöcke zur Unterstützung

«Scratch» bietet bereits in der Standardanwendung viele Blöcke, um den Nutzerinnen und Nutzern die Programmierung so einfach wie möglich zu machen. Neben den spezifischen Blöcken, die für die Kommunikation zwischen der Python-Umgebung und der «Scratch»-Anwendung (siehe [Abschnitt 3.2](#)) verwendet werden, hat sich ergeben, dass weitere Blöcke benötigt werden, um die Kommunikation im Netzwerk unterstützen.

- **Text abschneiden**

Das Protokoll für die Kommunikation ist textbasiert. Die Nachricht wird also mit zusätzlichen Angaben wie Empfänger und Absender ergänzt. Um die Nachricht doch als Nachricht zu erhalten, muss der Text entweder vor der Nachricht oder nach der Nachricht entfernt werden. Beispielsweise wird aus `Absender-Empfänger-Nachricht` die Nachricht extrahiert, indem `Absender-Empfänger` aus dem übermittelten Text entfernt wird. Das ist mit «Scratch» zwar möglich (siehe [Scratch Wiki Community 2018](#)), aber umständlich und herausfordernd. Neu gibt es einen Block, mit dem vom Beginn einer Zeichenabfolge eine bestimmte Anzahl Zeichen entfernt werden kann. Ergänzt wird dieser Block mit einem weiteren, mit dem vom Ende der Zeichenkette ebenfalls eine bestimmte Anzahl Zeichen entfernt werden können.

- **Text ver- und entschlüsseln**

Im Netzwerk werden die Nachrichten von den Geräten offen übertragen. Alle Geräte, die eingehende Nachrichten verarbeiten können, sind entsprechend in der Lage, die Nachrichten auch zu lesen und anzuzeigen. Damit die Nachrichten auch verschlüsselt übermittelt werden können, wurden zwei Blöcke gestaltet, einer um eine rohe Textkette mit Hilfe eines Passwortes zu verschlüsseln und einer, um die verschlüsselte Textkette mit einem Passwort wieder zu entschlüsseln.

3.3 Fehlerquellen beseitigen

Bereits bei der Entwicklung der Unterrichtseinheit soll Rücksicht auf mögliche Fehlvorstellungen und Fehlerquellen genommen und versucht werden, diese von Beginn an zu beseitigen oder zumindest einzuschränken. Für die vier möglichen Fehlerquellen aus dem vorangehenden Kapitel (siehe [Unterabschnitt 2.6.1](#)) wurden die folgenden Massnahmen umgesetzt.

3.3.1 Adressierung

Die einzelnen Geräte versenden offene Nachrichten ins Netzwerk. Diese Nachrichten erhalten per se keinen Absender noch einen Empfänger, sondern nur die jeweilige Nachricht. Als Hilfsmittel, damit die Lernenden erkennen, ob ihre Nachrichten versendet werden und dass diese auch gleich für alle sichtbar sind, wird neben den «Scratch»-Geräten auch ein Endgerät «Watcher» erstellt. Dieses Gerät beobachtet den gesamten Datenverkehr und listet alle empfangenen Nachrichten aus dem Netzwerk auf. Die Nachrichten werden dabei direkt einem Gerät zugeordnet und gruppiert.

Damit das möglich ist, werden alle Nachrichten im Hintergrund mit der MAC-Adresse versehen. Der «Watcher» kann so die eingehenden Nachrichten sortieren und einem Sender-Gerät zuordnen. Die Lernenden können so einerseits überprüfen, ob ihre Nachrichten

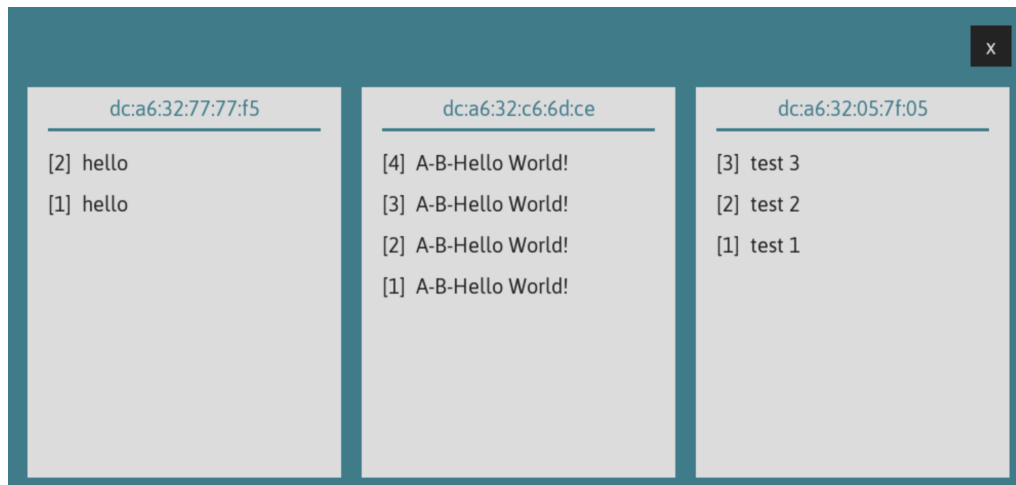


ABBILDUNG 3.7: Ausgabe vom «Watcher», Anzeige aller eingegangenen Nachrichten Quelle: Eigene Darstellung

auch tatsächlich ankommen und sehen auch die anderen Nachrichten, die im Netzwerk versendet werden. Für die Verarbeitung in «Scratch» wird allen eingehenden Nachrichten die hinzugefügte MAC-Adresse wieder entfernt. Die Lernenden erhalten mit dieser Massnahme nicht eine direkte Lösung für das Problem. Mit der Gruppierung der Nachrichten sehen sie aber direkt, dass Nachrichten von verschiedenen Geräten versendet werden. Durch die Anzeige der MAC-Adresse erhalten sie zudem einen Hinweis, dass die einzelnen Geräte einen internen Code - eben die MAC-Adresse - besitzen, um die Nachrichten zuzuordnen.

Im Aufgabenprozess werden sie nach dem Einstieg mit «Senden» und «Empfangen» darauf hingewiesen, dass Nachrichten mit einer Sender-Adresse und einer Empfänger-Adresse versehen werden sollen.

3.3.2 Protokollverarbeitung

Im Protokoll, das von den Lernenden entwickelt wird, besteht eine Hauptaufgabe darin, dass Nachrichten mit dem Sender und Empfänger versehen werden und diese für die weitere Verarbeitung wieder entfernt werden müssen. Dazu müssen die Programme in der Lage sein, einzelne Zeichen einem Text hinzuzufügen, sowie einzelne Zeichen von einem Text wieder zu entfernen. Das Zusammenfügen von bestehenden Texten mit neuen Text-Elementen ist in «Scratch» bereits vorbereitet und mit dem Block «verbinde» (siehe [Abbildung 3.8](#)) möglich.



ABBILDUNG 3.8: Scratch-Block für die Verbindung von zwei Texten Quelle: Eigene Darstellung

Wie bereits unter [Unterabschnitt 3.2.4](#) aufgezeigt, ist das Entfernen von gewissen Zeichen in «Scratch» nicht vorbereitet und nur über eine komplexe Programmierung (siehe [Scratch Wiki Community 2018](#)) machbar. Deshalb wurde der Block «entferne Zeichen am Anfang» (siehe [Abbildung 3.9](#)) beziehungsweise «entferne Zeichen am Ende» erstellt. Damit können die Lernenden beispielsweise die ersten vier Zeichen einer Nachricht entfernen.



ABBILDUNG 3.9: Scratch-Block für die Entfernung einer bestimmten Anzahl Zeichen Quelle: Eigene Darstellung

3.3.3 Parallelität vereinfachen

Die möglichen Probleme der Parallelität werden bereits in [Abschnitt 2.6.1](#) erläutert. Verhindert werden kann wohl nicht, dass die Lernenden Schwierigkeiten haben werden, wenn die Programme verschiedene Ausführungen gleichzeitig durchführen müssen. Trotzdem können zwei Dinge berücksichtigt werden:

Im Quellcode von [Code-Ausschnitt 3.6](#) wird ersichtlich, was passiert, wenn keine neuen Nachrichten eingetroffen sind: Das Programm auf dem Server versendet die Nachricht `'No Message received'`. So kann das «Scratch»-Programm zumindest überprüfen, ob aktuell gerade Nachrichten da sind, die verarbeitet werden müssen.

Im Quellcode von [Code-Ausschnitt 3.5](#) werden die eingehenden Nachrichten gespeichert. Dabei werden doppelte Nachrichten ersatzlos gelöscht. Das Programm muss so nicht gleiche Nachrichten mehrmals überprüfen und dabei warten, bis die Nachricht eingetroffen ist, die eigentlich erwartet wird. Doppelte Nachrichten tauchen auf, wenn ein Gerät zur Funktionsprüfung des Sendens im Sekundentakt eine bestimmte Nachricht verschickt. So können die Lernenden immer sicher sein, dass aktuell gerade Nachrichten verschickt werden können. Erwartete Nachrichten hingegen sind Nachrichten, die nach dem abgemachten Protokoll formatiert sind und an das eigene Gerät gerichtet sind. Hier muss das Programm überprüfen können, dass eine einkommende Nachricht an das Gerät gerichtet ist. Diese *relevanten* Nachrichten sollen nicht unnötig von anderen Nachrichten blockiert werden.

3.3.4 Sichtbarkeit der Datenübertragung

Die Daten werden im LoRaWAN unsichtbar via Funkwellen übertragen. Mit vertretbarem Aufwand ist es wohl nicht machbar, diese Wellen sichtbar zu machen. Die Verwendung des Endgerät «Watcher» kann allenfalls dazu beitragen, dass die Schülerinnen und Schüler *sehen*, welche Nachrichten übertragen wurden. Die Übertragung an sich ist zwar nach wie vor unsichtbar, aber das Ergebnis der Übertragung wird auf einem zweiten Gerät einsehbar. Nun liegt es an der Vorstellungskraft, dass die Lernenden sehen, wie ihre Botschaften drahtlos bei anderen Geräten im Netzwerk ankommen.

Kapitel 4

Unterrichtsszenario

4.1 Präsentation Unterrichtsszenario

Aus der vorangehenden Entwicklung ergibt sich für das Unterrichtsszenario ein schrittweiser Aufbau in die eigene Programmierung eines Netzwerkes, beziehungsweise der Umsetzung einzelner Netzwerkkomponenten:

1. Kennenlernen von Scratch⁴⁶.
2. Kennenlernen der neuen Scratch-Blöcke zum Senden von Nachrichten.
3. Kennenlernen der neuen Scratch-Blöcke zum Empfangen von Nachrichten.
4. Nachrichten spezifisch an ein anderes Gerät senden, bilaterale Kommunikation aufbauen. Dazu muss ein einfaches Protokoll entwickelt werden.
5. Das Netzwerk wird so erweitert, dass mit verschiedenen Geräten kommuniziert werden kann. Dazu muss das Protokoll aus den vorangehenden Aufgaben allenfalls erweitert werden.
6. (Erweiterung A) Die Kommunikation zwischen zwei Geräten wird verschlüsselt, damit die anderen Geräte im Netzwerk die private Kommunikation nicht mitlesen können⁴⁷ (siehe [Abschnitt 3.2.3](#)).
7. (Erweiterung B) Routing einbauen, so dass über mehrere Stationen kommuniziert werden kann. Einzelne Pakete der Kommunikation werden von den verschiedenen Geräten weitergeleitet, andere nicht⁴⁸ (siehe [Abschnitt 3.2.3](#)).
8. (Erweiterung C) Netzwerk mit einem Router erweitern, der die An- und Abmeldung von Geräten im Netzwerk übernimmt und Adressen verteilt. Ebenfalls vermittelt der Router zwischen einzelnen Geräten und weiss, welche Geräte im Netzwerk erreichbar sind (siehe [Abschnitt 3.2.3](#)).

⁴⁶In der Durchführung des hier entwickelten Unterrichtsszenarios wird davon ausgegangen, dass die Lernenden bereits Erfahrung mit Scratch haben. Die Schulklassen wurden bewusst so ausgesucht.

⁴⁷In dieser Erweiterung könnte das Thema Verschlüsselung eingebaut werden und beispielsweise ein System für eine RSA-Verschlüsselung programmiert werden.

⁴⁸Beispielsweise könnten gesammelte Daten über die ganze Schulgemeinde von den einzelnen Kinderzimmern bis ins Schulzimmer übertragen werden. Hier würde die Distanz von LoRaWAN mit einem einzelnen Empfänger wahrscheinlich nicht ausreichen.

Das Unterrichtsszenario ist so aufgebaut, dass die Lernenden in vier Gruppen mit je einem Endgerät arbeiten können. Die Geräte sind ausgerüstet mit der fertigen Software sowie den vorgegebenen Hardware-Komponenten⁴⁹. Jedes Gerät kann via «LoRa»-Übertragung mit den anderen Geräten kommunizieren und wird direkt via Scratch programmiert.

4.1.1 Didaktische Reduktion

Die besprochenen Inhalte für die Lernenden aus dem Zyklus 3 (siehe Abschnitt 2.4), kombiniert mit den Überlegungen der didaktischen Reduktion (siehe Unterabschnitt 2.3.2) ergeben Reduktionen in den folgenden Bereichen:

Einfache Netzwerkarchitektur

Die normale Kommunikation in unserer Einheit funktioniert auf der gleichen hierarchischen Ebene für alle Geräte. Keines der Geräte (abgesehen von den Erweiterungen) übernimmt Kontrollfunktionen oder vergibt Zugänge oder Adressen im Netzwerk. Die Geräte sind untereinander gleichberechtigt und können direkt miteinander kommunizieren.

Protokolle

Protokolle definieren, wie die Kommunikation in Netzwerken funktioniert. Dabei werden verschiedene Abläufe definiert, aber auch wie der Verbindungsaufbau, -abbruch oder -unterbruch gehandhabt wird. Neben den Abläufen wird auch definiert, wie Daten in einzelne Pakete zerlegt werden und wie diese übertragen werden. Dies sind nur um ein paar Beispiele, die vielen verschiedenen Netzwerkprotokolle unterscheiden sich grundlegend und decken die gesamten Kommunikationsmöglichkeiten ab.

Die Reduktion in diesem Bereich ist dementsprechend am grössten, die Protokolle werden stark vereinfacht oder gleich ganz weggelassen. In der Unterrichtseinheit soll erreicht werden, dass Nachrichten von einem Gerät zu einem anderen Gerät übermittelt werden. Das Protokoll muss also neben der Nachricht nur die Informationen zum Absender und Empfänger beinhalten. Alles weitere, wie etwa die Zerteilung in verschiedene Pakete oder die Codierung für die Übertragung, übernimmt das LoRa-Protokoll im Hintergrund. Die Lernenden entwickeln nur das Protokoll für die reine Kommunikation zwischen zwei Geräten.

Adressierung von Geräten

Im Netzwerk gibt es neben der MAC-Adresse das eigene Protokoll der IP-Adressen, in welchem definiert wird, wie die Adressen von IPv4⁵⁰ oder IPv6⁵¹ aufgebaut sind, wie sie verteilt werden und wie sie im Zusammenspiel mit anderen Geräten im Netzwerk funktionieren. Auch hier kann einiges reduziert werden, insbesondere da das kleine Netzwerk aus der Unterrichtseinheit in der ersten Version gerade einmal fünf Geräte⁵² beinhaltet. Diese fünf Geräte müssen aber trotzdem adressiert werden. Das kann, je nach Idee der Schülerinnen und Schüler, durch ein Buchstabe oder eine Zahl oder auch etwas komplizierter erfolgen. Die Lernenden werden ihr eigenes System entwickeln und dabei auch gleich definieren, wie die einzelnen Geräte ihre eigene Nummer erhalten und speichern.

⁴⁹Siehe Anhang A und Anhang B

⁵⁰IP-Adressen der Version 4, zum Beispiel «192.168.1.42»

⁵¹IP-Adressen der Version 6, zum Beispiel «2001:0db8:85a3:08d3:1319:8a2e:0370:7344»

⁵²4x Endgerät «Scratch» und 1x Endgerät «Watcher»

Umsetzung in einer Programmierumgebung

Protokolle, IP-Adressen, Metadaten in einem Netzwerk werden normalerweise tief in der Infrastruktur von den einzelnen Systemen umgesetzt und verankert. Eine entsprechende Umsetzung ist für den Unterricht nicht realistisch, da die Informatikkenntnisse viel zu hoch sein müssten. Die Umsetzung soll deshalb mit einer deutlich einfacheren Lösung geschehen, in einer Programmierumgebung, welche die Lernenden entweder bereits kennen, oder die sie selber relativ einfach kennenlernen können. «Scratch» ist gemäss [Abschnitt 3.1.3](#) eine Programmierumgebung, die für die Zielstufe geeignet ist. Die Unterrichtseinheit ist darauf ausgelegt, dass die entsprechenden Schritte für die jeweiligen Bereiche in «Scratch» didaktisch reduziert umgesetzt werden.

4.1.2 Erworbene Kompetenzen

Aus der didaktischen Reduktion ([Unterabschnitt 4.1.1](#)) ergeben sich die folgenden Kompetenzen, die durch die Lernenden erworben werden sollen.

1. Die Lernenden verstehen, dass Geräte nur miteinander kommunizieren können, sofern sie über ein gemeinsames Netzwerk miteinander verbunden sind.
2. Die Lernenden sind in der Lage, eine eigene «Sprache» zu entwickeln, damit Geräte miteinander kommunizieren können. Dabei definieren sie einzelne spezifische Meta-Informationen, die nötig sind, damit sich die Geräte verstehen. Die Kommunikationsdaten werden mit den Meta-Informationen ergänzt.
3. Die Lernenden können ein System entwickeln, damit Geräte eindeutig in einem Netzwerk von mehreren Geräten voneinander unterschieden werden können.
4. Die Lernenden können ein eigenes, theoretisches Modell für die einfache Kommunikation zwischen zwei Geräten in der Praxis mit Hilfe der Programmierumgebung «Scratch» umsetzen.
5. Die Lernenden können Datenströme, transferiert mit einem bereits bekannten Protokoll, aus einem Netzwerk analysieren und herauslesen, welche Informationen von fremden Geräten versendet werden.
6. Die Lernenden haben eine Vorstellung davon, wie «echte» Geräte in einem «echten» Netzwerk über «echte» Protokolle miteinander kommunizieren. Sie können ihr eigenes «Modell» der Kommunikation auf andere Kommunikationsnetzwerke transferieren.

Überprüfung der Kompetenzen

Die ersten vier Kompetenzen aus dem [Unterabschnitt 4.1.2](#) sind bei den Schülerinnen und Schülern vorhanden, sobald sie die Unterrichtseinheit mit den vier Aufgaben gelöst haben. Inwiefern der Transfer der beiden letzten Kompetenzen («Datenströme mitlesen» und «eigenes Kommunikationsmodell auf die Realität anwenden») funktioniert, muss genauer überprüft werden. Ebenso sollte überlegt werden, wie die vier Kompetenzen bei den einzelnen Lernenden überprüft werden, da das Endresultat der Unterrichtseinheit ein Produkt einer ganzen Gruppe ist. Ob die einzelnen Schülerinnen und Schüler die Kompetenzen auch tatsächlich erworben haben, muss genauer untersucht werden. Mehr zur Überprüfung der Kompetenzen findet sich in [Unterabschnitt 5.1.1](#).

4.1.3 Kommentar für Lehrpersonen

Für die Lehrpersonen, die mit dieser Unterrichtseinheit arbeiten möchten und sie selber im Unterricht einsetzen, wird ein «Kommentar für Lehrpersonen» (siehe [Abschnitt C.2](#)) bereitgestellt. Neben der Idee und einer Anleitung für die Umsetzung der Unterrichtseinheit im eigenen Unterricht, fasst eine Sachanalyse die nötigen Hintergründe zusammen. Ebenfalls sind darin die beiden Interventionen mit der ganzen Klasse, der Einstieg zum Thema «Nachrichten versenden via Mobiltelefon» sowie der Input zum Thema «Protokolle» (siehe auch [Unterabschnitt 4.1.5](#)) skizziert.

4.1.4 Aufgabenkarten

Die Lernenden erhalten Anleitungskarten, die jeweils die Aufgabe sowie eine oder mehrere Hilfestellungen enthalten. So können sich die Lernenden an den Aufgaben orientieren, werden aber nicht gross eingeschränkt, so dass sie kreativ arbeiten können. Eine erste Hilfestellung für die Lösung der Aufgabe beziehungsweise ein erster Lösungsansatz ist auf den Karten vorbereitet, mehr aber nicht. Die Idee hinter den Karten ist eine Verknüpfung der «Skill Cards» und «Challenge Cards» von Schmalfeldt (siehe [Schmalfeldt 2019](#)). Mit den Karten können die Lernenden in ihrem eigenen Tempo arbeiten und Probleme selber erarbeiten, in erster Linie durch ausprobieren, in zweiter Linie durch Kombination mit den bereits erlernten Inhalten aus den vorangehenden Aufgaben. Mit den Hinweisen wird sichergestellt, dass die Hilfestellungen durch die Lehrperson auf ein Minimum reduziert werden.

In einem ersten Teil (siehe [Abbildung 4.1](#) und [Abbildung 4.2](#)) werden die Lernenden mit dem Senden und Empfangen von Nachrichten vertraut gemacht. Sie erlernen die Grundlagen für die nächsten Schritte.

1 Senden NETZWERKE

a) Schreibe ein Programm, das eine Nachricht ins Netz versendet.

b) Ergänze das Programm mit einer Texteingabe. Versende nun die Texteingabe ins Netz.

c) Ergänze das Programm so, dass in regelmässigen Abständen Nachrichten ins Netz versendet werden.

ABBILDUNG 4.1: Aufgabe 1: Anleitung
Quelle: Eigene Darstellung

1 Hilfe NETZWERKE

Nachricht senden

NETZ Nachricht -> Netzwerk nachricht

Texteingabe

frage Welche Nachricht möchtest du versenden? und warte

setze nachricht auf Antwort

Nachricht regelmässig versenden

wiederhole fortlaufend

NETZ Nachricht -> Netzwerk nachricht

warte 1 Sekunden

ABBILDUNG 4.2: Aufgabe 1: Hilfestellung
Quelle: Eigene Darstellung

Im zweiten Teil (siehe [Abbildung 4.5](#) und [Abbildung 4.6](#)) wenden die Lernenden die beiden Funktionen «Senden» und «Empfangen» an und erstellen einen einfachen Chat, damit sie Nachrichten zwischen zwei Geräten hin und her senden können.

2 Empfangen NETZWERKE

a) Schreibe ein Programm, das alle Nachrichten aus dem Netz anzeigt.

b) Schreibe ein Programm, das nur „HEY“-Nachrichten anzeigt.

c) Schreibe ein Programm, das alle Nachrichten anzeigt, die mit „HEY“ beginnen.


Die Katze kann die Nachrichten ausgeben.

Für die Aufgabe b) und c) musst du mit einem anderen Team zusammenarbeiten. Ein anderes Gerät muss Nachrichten mit dem Inhalt „HEY“ verschicken sowie Nachrichten, die mit „HEY“ beginnen.

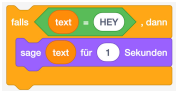
ABBILDUNG 4.3: Aufgabe 2: Anleitung Quelle: Eigene Darstellung

2 Hilfe NETZWERKE

Nachricht empfangen



Text überprüfen



Text vergleichen




ABBILDUNG 4.4: Aufgabe 2: Hilfestellung Quelle: Eigene Darstellung

3 Chat NETZWERKE

a) Entwickle ein System auf dem Papier, um mit einem anderen Gerät einen Chat zu starten. Besprich dein System mit dem anderen Team.

b) Programmiere nun einen Chat, mit dem du Nachrichten an ein anderes Gerät senden und von diesem Gerät empfangen kannst.

Ergänze dein Programm so, dass es nur auf die Chat-Nachrichten vom anderen Gerät reagiert.

ABBILDUNG 4.5: Aufgabe 3: Anleitung Quelle: Eigene Darstellung

3 Hilfe NETZWERKE

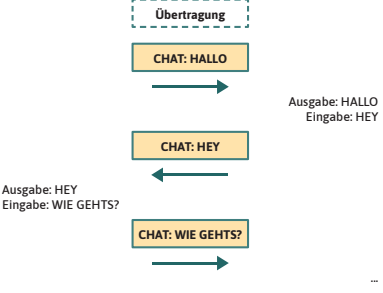


ABBILDUNG 4.6: Aufgabe 3: Hilfestellung Quelle: Eigene Darstellung

In diesem Teil entwickeln die Lernenden bereits ein einfaches Protokoll, so dass fremde Nachrichten im Netzwerk nicht berücksichtigt werden. Der eigens programmierte Chat würde sonst alle Nachrichten verarbeiten, die im Netzwerk versendet werden, und nicht nur die Direktnachrichten vom Partnerteam. Da die beiden Teams nur miteinander arbeiten, ist davon auszugehen, dass die Identifikation der Nachrichten über dieses einfache Protokoll noch eindeutig ist und nicht mit den anderen Gruppen kollidiert.

Für die letzte Aufgabe (siehe [Abbildung 4.7](#) und [Abbildung 4.8](#)) verknüpfen die Lernenden die bereits erarbeiteten Grundlagen. Nun ist eine gemeinsame Absprache nötig: die Schülerinnen und Schüler müssen gemeinsam ein einfaches Protokoll (siehe [Unterabschnitt 4.1.5](#))

entwickeln, um Chat-Nachrichten von einem beliebigen Gerät an ein anderes Gerät zu senden. In jeder dieser Chat-Nachricht muss neben der eigentlichen Nachricht eine Information über den Absender sowie über den Empfänger enthalten sein.

4
Netzwerk
NETZWERKE

a) Überlegt als ganze Gruppe, wie ihr Geräte identifizieren könnt.

b) Programmiere nun einen Chat, mit dem du Nachrichten an ein beliebiges Gerät senden kannst. Empfange Nachrichten nur von diesem Gerät.

c) Erweitere das Programm so, dass du einen Chat beenden kannst.

d) Nachdem ein Chat beendet wird, sollen neue Chats möglich sein.

ABBILDUNG 4.7: Aufgabe 4: Anleitung
Quelle: Eigene Darstellung

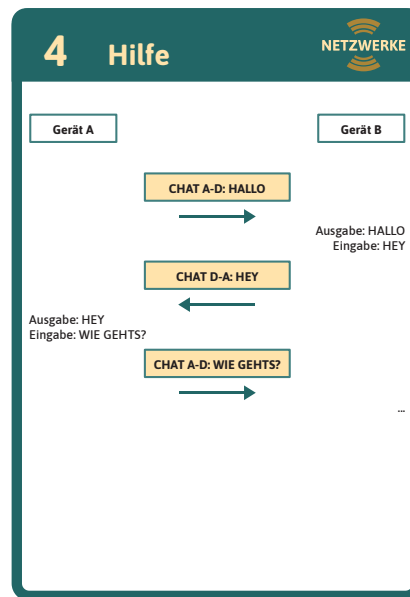


ABBILDUNG 4.8: Aufgabe 4: Hilfestellung
Quelle: Eigene Darstellung

4.1.5 Protokoll

Die vier Aufgaben können von den Lernenden grundsätzlich eigenständig erarbeitet werden. Vor der vierten Aufgabe soll für alle Lernenden ein Input durchgeführt werden und Protokolle besprochen werden. In diesem Input geht es darum, dass die Lernenden verstehen, dass Protokolle aus einem vordefinierten Ablauf bestehen. Die Lernenden kennen dazu bereits Beispiele:

- Telefongespräch:
 1. Person A: gibt die Nummer ein
 2. Person A: startet den Anruf
 3. Person B: nimmt den Anruf entgegen
 4. Person B: «VornameB, Hallo»
 5. Person A: «Hallo VornameB, da ist VornameA»
 6. Person A: startet Gespräch
 7. ...
- Begrüßung mit Handschlag
 1. Person A: hält Hand hin
 2. Person B: ergreift Hand
 3. Man begrüßt sich
 4. Man lässt die Hände wieder los

Die Lernenden sollen nun selber als Gruppe ein geeignetes Protokoll entwickeln, damit die einzelnen Geräte miteinander kommunizieren können.

Ein einfaches Protokoll ist in Abbildung 4.9 aufgezeichnet.

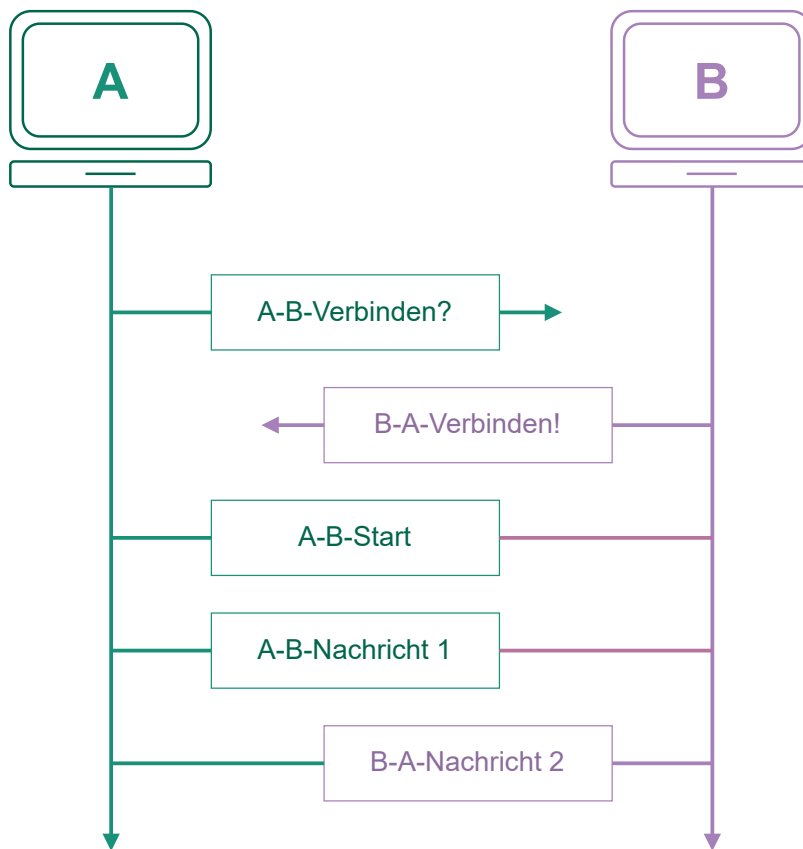


ABBILDUNG 4.9: Mögliches Protokoll zur Kommunikation im Netzwerk mit mehreren Geräten. Quelle: Eigene Darstellung

Im Protokoll werden alle Nachrichten mit dem Absender und dem Empfänger ergänzt, so dass diese nun aus drei Teilen bestehen:

- Empfänger
- Absender
- Nachricht

Die drei Teile der Nachricht sind dabei mit einem Bindestrich⁵³ voneinander getrennt und immer gleich aufgebaut: <Empfänger>-<Absender>-<Nachricht> oder eben A-B-Start. Im Protokoll können zudem drei spezielle Nachrichten definiert werden, die eine beliebige Kommunikation zwischen zwei Geräten starten:

- Verbinden?
- Verbinden!

⁵³Je nach Umsetzung kann der Bindestrich als Trennzeichen der drei Abschnitte Probleme bereiten, da er auch in normalen Nachrichten vorkommen kann. In diesem Fall kann der Bindestrich durch ein beliebiges anderes Trennzeichen, beispielsweise ein / ersetzt werden.

- **Start**

Sobald ein Gerät eine Kommunikation mit einem beliebigen Gerät starten möchte, sendet es die Nachricht `<Absender>-<Empfänger>-Verbinden?` los. Andere Geräte müssen einkommende Nachrichten nach dem eigenen Gerät als Empfänger und die Nachricht auf `Verbinden?` scannen. Sobald die Verbindung angefragt wird, kann mit `<Empfänger>-<Absender>-Verbinden!`⁵⁴ die Anfrage bestätigt werden. Kommt als weitere Nachricht `start` zurück, folgen anschliessend die Nachrichten, die vom Programm nicht mehr automatisch verarbeitet werden sollen, sondern diese Nachrichten sollen im Chat auftauchen.

4.1.6 Klassengrösse

Für die Unterrichtseinheit empfiehlt es sich, in halben Klassen zu arbeiten. Pro Endgerät können zwei bis drei Personen arbeiten. Für die Erprobung wurden 4 Endgeräte erbaut, somit können bis zu 12 Lernende am Unterricht teilnehmen. Alternativ könnten mehr Endgeräte eingerichtet werden.

4.1.7 Zeitaufwand

Für das Unterrichtsszenario sollten minimal drei Lektionen à 45 Minuten zur Verfügung stehen, um der Planung (siehe [Tabelle 4.1](#)) zu entsprechen. Die Erfahrungen haben gezeigt, dass die Umsetzung mehr Zeit in Anspruch nehmen kann.

Zeit	Form	Aktivität	Material
00:00	Input	Was braucht es, damit ich vom Handy Nachrichten verschicken kann?	
00:10		Verteilen Geräte, Gruppeneinteilung	4x Endgeräte, 1x Watcher
00:15		Inbetriebnahme Geräte, Erklärung Funktion	
00:25	ind. GA ⁵⁵	Aufgabe 1: Senden	Scratch-Anwendung
00:35	ind. GA	Aufgabe 2: Empfangen	
00:45	ind. GA	Aufgabe 3: Erarbeitung Protokoll & einfacher Chat	
01:30	Input	Wie funktionieren Protokolle?	
01:40	ind. GA	Aufgabe 4: Klassen-Chat	

TABELLE 4.1: Planung Unterrichtseinheit.

4.2 Erprobung Szenario

Das Unterrichtsszenario wurde mit verschiedenen Halbklassen erprobt. Die Halbklassen waren in Gruppengrößen zwischen 3 und 11 Schülerinnen und Schülern im Alter von 13 bis 19 Jahre eingeteilt. Die Versuche wurden iterativ mit fünf verschiedenen Gruppen durchgeführt und laufend evaluiert und überarbeitet. Insbesondere bei den ersten Durchläufen

⁵⁴Achtung: in diesem Fall ändern sich Absender und Empfänger, Gerät A startet die Anfrage mit «Verbinden?» an B (B-A-Verbinden?), B antwortet darauf mit A-B-Verbinden!. Der Absender ist immer das jeweilige Gerät, das die Mitteilung versendet.

⁵⁵individuelle Gruppenarbeit

lag der Fokus darauf, dass die Technik funktionierte und die Unterrichtsmaterialien den Lernenden bei ihrem Lernprozess eine Hilfe waren.

Die Bilder *Abbildung 4.10*, *Abbildung 4.11*, *Abbildung 4.12* und *Abbildung 4.13* zeigen einzelne Momentaufnahmen aus der Durchführung mit den Lernenden.

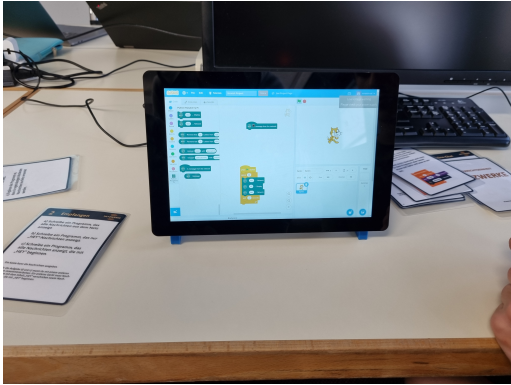


ABBILDUNG 4.10: Material einer einzelnen Gruppe bei der Durchführung
Quelle: Eigene Aufnahme

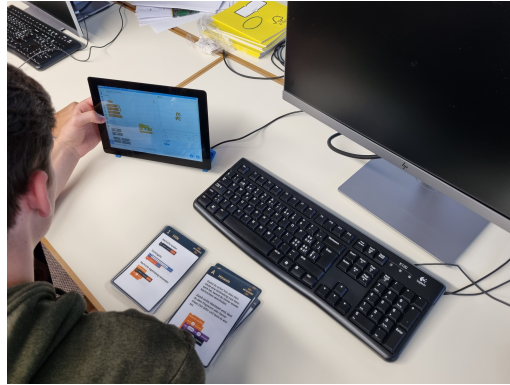


ABBILDUNG 4.11: Ein Schüler bei der Bedienung des Endgeräts «Scratch»
Quelle: Eigene Aufnahme



ABBILDUNG 4.12: Zwei Mädchen beim Entwickeln ihrer Lösungsstrategie
Quelle: Eigene Aufnahme

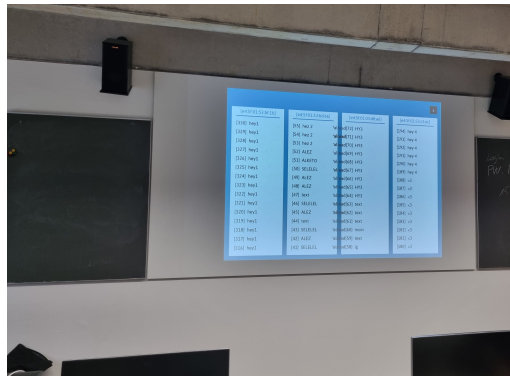


ABBILDUNG 4.13: Projektion des Endgeräts «Watcher» für die Darstellung aller versendeten Nachrichten
Quelle: Eigene Aufnahme

4.2.1 Technische Probleme

Bei der Durchführung sind insbesondere in den ersten Durchläufen technische Probleme aufgetaucht, die zwischen den Durchläufen jeweils behoben wurden:

Übertragung von Texten

Die Nachrichten werden als JSON-Objekte zwischen Scratch und Server übertragen, bevor diese via LoRaWAN an andere Geräte übermittelt werden. Die Tests wurden mit Zahlen durchgeführt, sobald in den Durchläufen Texte verschickt wurden, konnten die Server keine Nachrichten mehr verarbeiten, da die Umwandlung von Texten zu JSON-Objekten anders funktioniert als die Umwandlung von Zahlen. Das Problem konnte rasch behoben werden.

Anzeige auf dem «Watcher» (1)

Die Anzeige auf dem «Watcher» zeigt die letzten Nachrichten an, je nach Platz auf dem Bildschirm etwas mehr oder etwas weniger. Wenn die Lernenden nun unendlich wiederholend

neue Test-Nachrichten versenden, ist die Anzeige relativ schnell voll⁵⁶. Neue Nachrichten werden zwar nach wie vor angezeigt, die Anzeige verändert sich aber nicht mehr. Als Lösung werden nun alle Nachrichten durchnummeriert und hochgezählt. Die Anzeige verändert sich nun, auch wenn die Nachrichten die gleichen bleiben.

Anzeige auf dem «Watcher» (2)

Die Anzeige auf dem «Watcher» war nicht darauf ausgelegt, längere Texte anzuzeigen. Die Anzeige von den Nachrichten kam durcheinander und die neuen Nachrichten konnten nicht mehr gelesen werden. Als Lösung werden die Nachrichten nun gekürzt, falls sie zu lange für die Anzeige sind (siehe auch [Abbildung 3.7](#)). Die Lernenden merken trotz der Kürzung, um welche Nachrichten es sich handelt.

Variablen können nicht erstellt werden

In der Scratch-Umgebung können keine neuen Variablen erstellt werden. Das Problem liegt an der Bedienung über die virtuelle Tastatur. Bei dieser digitalen Eingabe werden keine Ereignisse ausgelöst, sobald eine Taste gedrückt wird. Scratch ist aber darauf programmiert, dass auf exakt diese Ereignisse reagiert wird. Die Lösung war hier eine Anpassung der Scratch-Umgebung⁵⁷, so dass die Variable abgespeichert wird, auch wenn keine Eingabe über eine richtige Tastatur getätigt wird.

Fehlende Antwort beim Frage-Antwort-Block

Ähnlich wie bei [Abschnitt 4.2.1](#) wurden hier mit der virtuellen Tastatur keine Eingaben vom User gespeichert. Die «Antwort» vom «Frage und warte»-Block blieb auf jeden Fall leer. Auch hier war die Lösung, dass die Scratch-Umgebung angepasst wird⁵⁸. In diesem Fall musste die eingegebene Antwort ausgelesen werden, direkt nachdem die Antwort bestätigt wurde.

4.2.2 Konzeptuelle Probleme

Bezug Lebenswelt

Damit der Bezug zur Lebenswelt der Jugendlichen noch klarer hervorgehoben wird, wurde dieser direkt in den Aufgabenkarten verankert. In einer ersten Version war die Verbindung zu WhatsApp- und Snapchat-Nachrichten nur Teil der Instruktion zu Beginn der Unterrichtseinheit. Da die Lernenden sich sehr stark an den Aufgabenkarten orientieren, wurde diese Grundidee dort eingebaut (siehe [Abschnitt C.1](#)).

Einfache Hilfestellungen

In den ersten Durchführungen hat sich gezeigt, dass bei allen Gruppen ähnliche Probleme auftauchten. Mit vier Hinweis-Karten, die ebenfalls in den Aufgabenkarten integriert wurden, konnten diese Probleme verringert werden.

- **Verbindungsaufbau überprüfen und Verbindungsabbruch verhindern**

⁵⁶Das Test-Programm, welches von vielen Lernenden in den ersten Schritten entwickelt wurde, verschickt jede Sekunde eine gleiche Nachricht.

⁵⁷siehe github.com/btemperli/scratch-gui, commit 0e56af6

⁵⁸siehe github.com/btemperli/scratch-gui, commit 4803b4c

Das Problem ist, dass die Verbindung bei Nichtgebrauch teilweise unterbrochen wird. Sobald regelmässig Nachrichten verschickt werden, verringert sich erstens das Problem der Unterbrechung und zweitens sieht man direkt, wenn die Nachrichten nicht mehr auf dem «Watcher» ankommen, dass etwas schief läuft.

- **Der eingehende Nachrichtenblock wechselt nach jedem Aufruf zur nächsten Nachricht**

Der Nachrichtenblock darf nur einmal verwendet werden. Wird er ein zweites Mal aktiviert, wird im Hintergrund gleich die nächste Nachricht geladen. Das Problem taucht auf, wenn die Nachricht zuerst überprüft wird, beispielsweise ob die Nachricht mit dem Buchstaben «A» beginnt, und danach die gleiche Nachricht ausgegeben wird. Im Hintergrund wechselt die Nachricht direkt nach der Überprüfung und die nächste Nachricht aus der Liste wird anschliessend ausgegeben. Verhindert wird dies, indem mit Variablen gearbeitet wird. Die Variable behält ihren Wert, bis dieser aktiv angepasst wird.

- Parallelität: Scratch-Programme vereinfachen
- Scratch-Block «frage und warte» erläutern

Die 4 Karten sind im Anhang (siehe [Abschnitt C.1](#)) aufgeführt.

4.3 Evaluation Szenario

Die Auswertung des durchgeführten Unterrichtsszenarios wurde mit Hilfe von einem digitalen Fragebogen durchgeführt. Der Fragebogen wurde in vier Teilen aufgebaut und dient einer ersten, groben, hauptsächlich quantitativen Rückmeldung zur Unterrichtseinheit. Der Fragebogen ist darauf ausgelegt, einfache Antworten auf die verschiedenen Bereiche zu erhalten, die in der Unterrichtseinheit entwickelt wurden, und erste Rückmeldungen zur Durchführung zu erhalten.

Die total 30 Lernenden wurden auf freiwilliger Basis gebeten, den Fragebogen auszufüllen. 12 Lernende haben die Umfrage zumindest teilweise ausgefüllt. Die Resultate sind unter [Anhang D](#) aufgeführt. In den folgenden Abschnitten werden die wichtigsten Resultate aufgeführt und es wird versucht, erste Schlüsse für die weitere Überarbeitung der Unterrichtseinheit zu ziehen.

4.3.1 Teil 1: Übersicht

In diesem Teil wurde evaluiert, wie zufrieden die Lernenden mit ihrem Programm waren, wie schwierig die Aufgaben empfunden wurden, was funktioniert hat und was nicht. Weiter wurde gefragt, in welcher Konstellation die Teilnehmenden gearbeitet haben und welche Aufgaben von den einzelnen Lernenden übernommen wurden. Zur Zuordnungsbarekeit der Resultate mit den durchgeführten Szenarien wurde Name und Alter⁵⁹ abgefragt.

1. Wie zufrieden bist du mit deinem Scratch-Programm?

Stern-Bewertung, Single-Choice mit 1 bis 6 Sternen.

12 Antworten, durchschnittliche Bewertung: 4.83 von 6 Sternen.

2. Wie schwierig waren die Aufgaben?

⁵⁹Name und Alter werden aus Gründen der Anonymität aus den Resultaten entfernt.

1: super einfach, 10: super schwierig. Single Choice, Werte von 1 bis 10.

12 Antworten, durchschnittliche Bewertung: 6.42 von 10.

3. Konntest du alle Aufgaben lösen?

Die Mehrheit der Lernenden gab an, dass sie «fast alle» Aufgaben lösen konnten, wie in Abbildung 4.14 zu sehen ist.

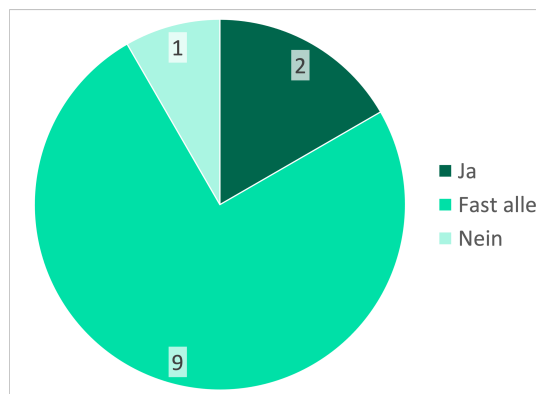


ABBILDUNG 4.14: Auswertung Frage 3 Quelle: Eigene Darstellung

4. Was hat funktioniert?

«Das sich die Nachrichten von alleine⁶⁰ senden»

«Das hin und her schicken von Nachrichten (das Netzwerk) [sic!]

«Es hat funktioniert, dass eine Gruppe die Chats der anderen Gruppe empfangen konnte und man etwas schreiben konnte, welches dann im Netzwerk erschienen ist. [sic!]

Auszüge aus den einzelnen Rückmeldungen zur Frage 4.

5. Was hat nicht funktioniert?

«Das Gerät ist oft abgestürzt»

«Es ist immer wieder mal abgestürzt»

«Zuerst hatten wir Probleme damit, dass es nur bestimmte Nachrichten anzeigt, welche z.B. ein Z davor hatten.»

«Nur von einem anderen Gerät Nachrichten zu erhalten»

Auszüge aus den einzelnen Rückmeldungen zur Frage 5.

Insbesondere die Antworten der Fragen 4 und 5 geben Ausschluss darüber, wo die Probleme beim Bearbeiten der Aufgaben liegen. Das Hauptproblem ist tatsächlich, dass die Verbindung zwischen dem Python-Server im Hintergrund und der Scratch-Umgebung im Vordergrund nicht sehr stabil läuft. Die Geräte müssen in der aktuellen Ausführung neu gestartet werden, sobald diese Verbindung getrennt wird. Das ist zeitaufwändig und es geht der aktuelle Stand des Programms verloren. Dieses Problem wird im [Unterabschnitt 4.4.2](#) genauer erläutert.

⁶⁰Gemeint ist wohl, dass das Programm in regelmässigen Abständen automatisch eine Nachricht versendet.

4.3.2 Teil 2: Aufgaben & Hilfe

In diesem Abschnitt wurde evaluiert, wie und wo Hilfe in Anspruch genommen wurde, und wie hilfreich die Aufgaben(-karten) aufgefasst wurden.

13. Waren die Aufgabenkarten verständlich?

Single Choice, «Ja» oder «Nein». Mit 100% Ja-Stimmen ist das Fazit sehr klar, die Aufgabenkarten sind verständlich, was auch [Abbildung 4.15](#) nochmals aufzeigt.

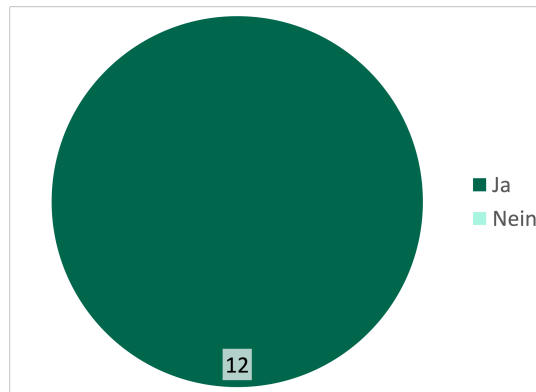


ABBILDUNG 4.15: Auswertung Frage 13 Quelle: Eigene Darstellung

14. Wo hast du Hilfe geholt während dem Programmieren?

Multiple Choice mit den folgenden Möglichkeiten zur Auswahl:

- «Hilfe-Seite auf den Aufgabenkarten»
- «Im eigenen Team»
- «Bei einem fremden Team»
- «Bei der Lehrperson»

Die Schülerinnen und Schüler wählten oft mehrere Antworten, am meisten Hilfe wurde bei der Lehrperson, gefolgt vom eigenen Team, geholt. Die Details sind in [Abbildung 4.16](#) zu sehen.

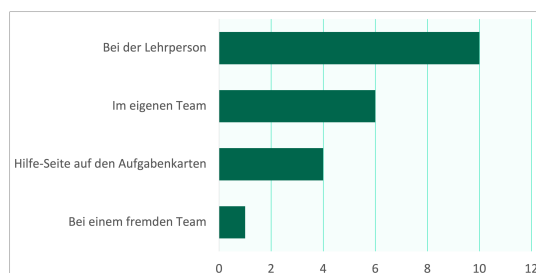


ABBILDUNG 4.16: Auswertung Frage 14 Quelle: Eigene Darstellung

17. Was war schwierig?

- «Die letzten Aufgaben [sic!]
- «Ab Aufgabe 3 kam ich nicht mehr ganz mit»
- «Mit der eher schlechten Hardware auszukommn. [sic!]

Auszüge aus den einzelnen Rückmeldungen zur Frage 17.

18. Was war einfach?

«Es war alles sehr anspruchsvoll»

«Ins netz zu chatten [sic!]

«Die ersten Aufgaben»

«Programmieren»

Auszüge aus den einzelnen Rückmeldungen zur Frage 18.

Die Fragen 13 und 14 geben Aufschluss darüber, dass die Aufgaben- und Hilfekarten funktionieren. Die Hilfekarten können gemäss Aufgabe 14 noch etwas detaillierter sein. Bei der Auswertung der Antwort 14 muss berücksichtigt werden, dass nicht alle Umfrageteilnehmenden bereits über die Hinweiskarten verfügen konnten, da diese erst ab der Hälfte der Durchführungen eingesetzt wurden. Ebenso muss man in Betracht ziehen, wie gerne die Lernenden Hilfestellungen lesen oder ob diese lieber Hilfe bei der Lehrperson erbeten, sofern diese direkt verfügbar ist. Die Fragen 17 und 18 zeigen auf, dass die Schwierigkeit nach den ersten beiden Aufgaben steil ansteigt. Allenfalls kann zwischen dem einfachen «Senden»⁶¹ und «Empfangen»⁶² und dem späteren «Kommunizieren»⁶³ eine weitere Aufgabe eingefügt werden, die die Kommunikation zwischen mehreren Geräten besser einführt.

4.3.3 Teil 3: Lerninhalte

22. Beurteile die folgenden Aussagen auf ihre Richtigkeit.

Verschiedene Aussagen zu Netzwerken sollten durch die Lernenden beurteilt werden, ob diese «richtig», «falsch» oder «nicht beurteilbar» sind. Die Aussagen wären alle richtig⁶⁴. Die Lernenden waren sich aber teilweise unsicher und gaben an, dass die Aussagen «nicht beurteilbar» seien oder stuften diese als «falsch» ein. Die Details finden sich in [Abbildung 4.17](#), die entsprechenden Aussagen waren die folgenden:

- [1] Um Daten in einem Netzwerk zu übertragen, braucht es immer einen Empfänger.
- [2] Um Daten in einem Netzwerk zu übertragen, braucht es immer einen Sender.
- [3] Ein Netzwerkprotokoll definiert, in welcher Form Daten übermittelt werden.
- [4] Bei einer SMS wird neben der Nachricht auch die Nummer von Empfänger und Sender mitgeschickt.
- [5] Ein Netzwerkprotokoll muss allen Geräten im Netzwerk bekannt sein.
- [6] Ein Netzwerk besteht immer aus mehreren Geräten.

⁶¹Aufgabe 1

⁶²Aufgabe 2

⁶³Aufgaben 3 und 4

⁶⁴Bei der Aussage 5 («Ein Netzwerkprotokoll muss allen Geräten im Netzwerk bekannt sein.») könnte man noch etwas besser formulieren: «Ein Netzwerkprotokoll muss den an der Kommunikation beteiligten Geräten im Netzwerk bekannt sein.»

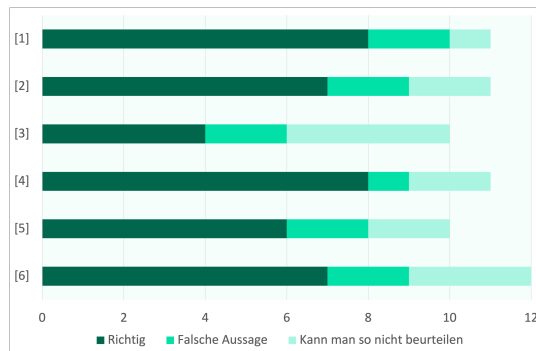


ABBILDUNG 4.17: Auswertung Frage 22 Quelle: Eigene Darstellung

Die gemachten Aussagen der Lernenden lassen darauf schliessen, dass diese die Materie grundsätzlich verstehen⁶⁵. Die Aussagen können aber nicht direkt der Unterrichtseinheit zugeordnet werden, da nicht überprüft wurde, wie der Wissensstand der Lernenden vor der Durchführung der Unterrichtseinheit war. Ebenfalls wurde keine der Aussagen direkt in der Unterrichtseinheit angesprochen, was zwar dafür spricht, dass die Materie verstanden wird, aber in diesem Kontext nicht mit dem Lerninhalt der Einheit übereinstimmt.

26. Welche Gefahren könnten in einem Netzwerk lauern?

- «Hacker, die Nachrichten abfangen. [sic!]»
- «Man kann gehackt werden [sic!]»
- «viren, hacker [sic!]»
- «Leute, die die Nachrichten mitlesen.»

Auszüge aus den einzelnen Rückmeldungen zur Frage 18.

Auch bei dieser Frage zeigt sich erneut, dass die genannten Punkte der Lernenden nicht auf die Unterrichtseinheit zurückzuführen sind. Abgesehen von der letzten Aussage, dem Mitlesen von Nachrichten, wurden die Inhalte in der Unterrichtseinheit nicht angesprochen.

Der dritte Teil kann darüber Aufschluss geben, wie der Wissensstand der Lernenden aussieht. Einen Zusammenhang zur Unterrichtseinheit kann jedoch nicht direkt hergestellt werden. Bei einer weiteren Bearbeitung des Szenarios sollte auch die Auswertung (siehe [Unterabschnitt 4.4.4](#)) detaillierter aufgeleitet werden.

4.4 Überarbeitung Szenario

Aufgrund der Beobachtungen während der Durchführung der Unterrichtseinheit und der anschließenden schriftlichen Evaluation durch die Teilnehmenden ergeben sich nun folgende Möglichkeiten für die Überarbeitung:

4.4.1 Überprüfung Hardware

Die Hardware hat nicht immer einwandfrei funktioniert. Insbesondere die Bedienung des Touch-Screens hat zwei beobachtbare Probleme verursacht:

⁶⁵Mehr als die Hälfte der abgegebenen Antworten sind korrekt

- **Doppelklick**

Der Doppelklick auf dem Startbildschirm funktioniert nicht so, wie sich das die Teilnehmenden gewohnt sind. Die Scratch-Anwendung zu starten hat deshalb Probleme verursacht. Das Problem kann behoben werden, indem die Scratch-Anwendung ohne Interaktion automatisch gestartet wird, sobald das Gerät angeschaltet wird.

- **Touch-Fehler bei einem der Touch-Screens**

Eines der Geräte erlitt einen punktuellen Ausfall der Touch-Funktion. So konnte der Buchstabe «E» auf der digitalen Tastatur nicht angewählt werden. Das Problem konnte nicht genauer eingegrenzt werden, kann aber umgangen werden, indem die Nachrichten und das Protokoll den Buchstaben «E» nicht nutzen. Das Problem könnte auch an der digitalen Tastatur liegen, die ein Problem mit dem Buchstaben «E» hat.

4.4.2 Verbindungsprobleme Scratch und Python

Die interne Verbindung zwischen der Scratch-Umgebung und des Python-Servers bricht von Zeit zu Zeit ab, meist aufgrund eines Fehlers in der Python-Umgebung. Bis jetzt konnte der Fehler nicht reproduziert und eingegrenzt werden, so dass eine Behebung nicht möglich ist. Zwei mögliche Weiterentwicklungen können das Problem aber beheben:

- Eingrenzung des Problems und Behebung des Fehlers, so dass der Python-Server nicht mehr aufgrund des Fehlers beendet wird.
- Mit einer weiteren Software überprüfen, ob die Verbindung noch läuft. Sobald die Verbindung getrennt wird, kann diese Drittsoftware den Server erneut starten, so dass die Verbindung im laufenden Betrieb wieder hergestellt werden kann.

Ein weiteres Problem entsteht bei der Bedienung durch die Schülerinnen und Schüler. Sobald die Verbindung abbricht, erscheint eine Fehlermeldung, die auf das Problem hinweist. Die Lernenden klicken die Fehlermeldung weg, ohne diese zu lesen. In der Fehlermeldung wird ebenfalls erwähnt, dass das Problem durch einen Neustart behoben werden kann.

4.4.3 Weiterentwicklung Scratch

Scratch wird stetig weiterentwickelt⁶⁶, insbesondere werden Fehler laufend behoben, sobald sie entdeckt werden. Diese Anpassungen sollten regelmässig in die eigene Scratch-Umgebung fließen und das Programm somit stetig aktuell gehalten werden.

Ähnliche Probleme wie unter [Abschnitt 4.2.1](#), die bis jetzt noch nicht entdeckt wurden, könnten so allenfalls früher beseitigt werden ohne spezifische Anpassungen an der Scratch-Umgebung.

In einem zweiten Schritt können weitere Teil-Applikationen von Scratch übernommen werden und eingebaut werden. Beispielsweise gibt es die Anwendung «scratch-www», mit der die Projektübersichtsseite⁶⁷ von einzelnen Scratch-Projekten eingebaut werden kann. Auf der Projektseite werden einzelne Projekte angezeigt und können mit Anmerkungen versehen und direkt abgespielt werden. Eine weitere Möglichkeit wäre, die eigene Scratch-Anwendung mit der User-Datenbank von Scratch zu verbinden. So können erstellte Netzwerk-Projekte mit dem eigenen Scratch-Account verknüpft werden.

⁶⁶Die «scratch-vm»-Applikation wird beinahe täglich, teilweise auch mehrmals täglich, aktualisiert, wie github.com/LLK/scratch-vm/commits/develop aufzeigt.

⁶⁷Zum Beispiel scratch.mit.edu/projects/104/

Hinweis

Weitere Scratch-Anwendungen benötigen allenfalls eine Verbindung zu den Scratch-Servern. Dies funktioniert nur, wenn der einzelne «Raspberry Pi» eine funktionierende Internetverbindung hat. Beim Projekt wurde darauf geschaut, dass diese Verbindung nicht existieren muss, und alle Programmteile ohne Internetverbindung funktionieren. Eine benötigte Internetverbindung kann durchaus weitere Probleme mit sich bringen, das beginnt bei der Eingabe von WLAN-Passwörtern^a. Die Geräte könnten auch nicht mehr orts- und netzunabhängig eingesetzt werden.

^aDie Desktop-Umgebung auf dem «Raspberry Pi» ist beispielsweise nicht darauf vorbereitet, dass eine digitale Tastatur verwendet werden kann.

4.4.4 Detaillierte Auswertung der Unterrichtseinheit

Damit aufgezeigt werden kann, wie erfolgreich die Unterrichtseinheit in Bezug auf deren Lerninhalte ist, sollte eine wissenschaftliche Auswertung der Inhalte durchgeführt werden. Die ersten Durchführungen haben gezeigt, dass die Aufgaben lösbar sind und die Lernenden dabei ein eigenes Netzwerkprotokoll entwickeln können. Ob die gewählte Form die richtige Herangehensweise ans Thema ist, ob die Lernenden die erlernten Inhalte auch in anderen Bereichen anwenden und in andere Gebiete transferieren können, müsste genauer untersucht werden. Sobald Netzwerke verbreiteter im Unterricht behandelt werden, sobald die Empfehlungen von Modrow und Strecker (siehe [Abschnitt 1.1](#) und Modrow und Strecker 2016) umgesetzt sind und das Thema «Internet» in der Schule vermehrt auch aus technologischer Sicht besprochen wird, kann die vorliegende Unterrichtseinheit mit anderen Unterrichtseinheiten verglichen werden.

4.5 Veröffentlichung Unterrichtsszenario

Die Unterlagen (Aufgabenkarten, Hinweiskarten, Kommentar für Lehrpersonen, vorbereitete Abbilder der Installationen für die Raspberry Pis) wurden alle so erstellt, dass sie frei zugänglich gemacht werden können. Sämtliche verwendeten Materialien, insbesondere Grafiken und Bilder⁶⁸, wurden entweder selber hergestellt oder stammen aus Quellen, die eine freie Weiterverbreitung ermöglichen.

Die angepasste Versionen von Scratch sind zur freien Verfügung unter Github veröffentlicht:

- Die Scratch-Anwendung für den Raspberry Pi: github.com/btemperli/scratch-raspberry
- Der Python-Server für die Kommunikation zwischen «Scratch» und «LoRa»: github.com/btemperli/raspberry-scratch-server

Alle weiteren Materialien sind unter temperli.io/master veröffentlicht und frei zugänglich.

⁶⁸Entsprechende Quellen sind dennoch unter [Anhang C](#) erwähnt.

Kapitel 5

Ergebnisse & Diskussion

5.1 Zusammenfassung und Ausblick

Auf theoretischer und später auch auf praktischer Ebene wurde erforscht, wie man im Zyklus 3 mit Lernenden eine Unterrichtseinheit zum Thema Netzwerk erstellen kann. Verknüpft mit moderner Technologie sind die entwickelten Geräte in der Lage, ein kleines Netzwerk über ihren jeweiligen LoRaWAN-Adapter, verknüpft mit je einem Raspberry Pi zu erstellen. Das ganze wird gesteuert via einer «Scratch»-Umgebung, die direkt auf den Raspberry Pis läuft und individuell programmiert werden kann.

Es wurde aufgezeigt, dass es möglich ist, den Bereich «Physical Computing» mit modernen und offenen Systemen zu erweitern. Mit dem «Raspberry Pi» konnte ein Gerät gefunden werden, das im Bereich «Prototyping» in der Industrie eingesetzt wird, genauso wie im Freizeitbereich bei der Entwicklung von kleinen intelligenten Geräten für zu Hause. In der Schule können die kleinen Computer eingesetzt werden und direkt via Scratch, einer bereits im schulischen Umfeld etablierten Programmierumgebung, programmiert und gesteuert werden. Die «Raspberry Pi»-Computer sind offene Systeme, die mit anderen offenen Systemen kompatibel sind und entsprechend ergänzt werden können. Neue technologische Entwicklungen können damit ausprobiert und entdeckt werden. Am Beispiel der Verknüpfung eines «Raspberry Pi» konnte aufgezeigt werden, dass diese Verbindung mit dem LoRa-Netzwerk erstellt werden kann, und die Geräte via «LoRaWAN» miteinander kommunizieren können und nun in der Lage sind, Daten auf Scratch-Befehle hin miteinander auszutauschen. Die Dreiecksbeziehung zwischen «Scratch», «Raspberry Pi» und «LoRa»-Netzwerk basiert auf einer «Websocket»-Verbindung in einer Python-Umgebung, die beliebig erweitert werden kann.

5.1.1 Weiterentwicklung der Ergebnisse

Im Laufe dieser Arbeit hat sich gezeigt, dass weitere Entwicklungsschritte möglich sind und durchaus spannende, weitere Effekte der «Netzwerktechnologie im Unterricht» hervorgerufen werden können.

Technologische Weiterentwicklung

In der nachfolgenden, nicht abschliessenden Liste werden einzelne technische Punkte kurz erläutert, die in naher oder ferner Zukunft umgesetzt werden können. Ebenfalls wird erwähnt, welche Effekte und Auswirkungen die Weiterentwicklung jeweils haben könnte.

- Asynchrone Verschlüsselung

Mit Hilfe einer asynchronen Verschlüsselung können unabhängige verschlüsselte Kommunikationen erfolgen. Die teilnehmenden Geräte können als erstes einen beliebigen

öffentlichen Schlüssel übermitteln und die Nachrichten an das Gerät können mit einem Public-/Private-Key-Verfahren verschlüsselt werden. Andere Geräte, auch der «Absender», können die verschlüsselten Nachrichten nicht entschlüsseln, dies kann nur das Gerät, welches den privaten Schlüssel, zugehörig zum öffentlichen Schlüssel, besitzt.

- Anwendungsgebiete «Scratch» in Verbindung mit «Raspberry Pi» öffnen

In der vorliegenden Arbeit wurde eine Verbindung zwischen «Physical Computing», blockbasierter Programmierung und neuester Technologie gesucht, gefunden und erprobt. Dies mit einer beispielhaften Unterrichtseinheit im Bereich Netzwerktechnologie. Die offenen Systeme rund um den Microcontroller Raspberry Pi lassen bestimmt viele weitere Anwendungen zu. Verknüpft mit Scratch könnte das neue Wege eröffnen, um neue Technologie bereits auf Stufe Zyklus 3 im Unterricht einzubauen. Das könnten beispielsweise die Ansteuerung von Motoren für ein eigenes kleines Rennauto sein, der Einsatz von GPS-Lokalisation zur genauen Erfassung von Standorten oder eine Verknüpfung mit Wetter-Sensoren, um eine eigene Wetter-Station zu entwickeln.

Konzeptuelle Weiterentwicklung

Der Fokus bei der vorliegenden Arbeit liegt beim Konzept einer technischen Umsetzung von Scratch, in Anwendung mit einem Raspberry Pi und verknüpft mit dessen technischen Schnittstellen. Die wissenschaftliche Evaluation der Unterrichtseinheit ist entsprechend klein ausgefallen. Diese und weitere Punkte finden sich in der folgenden, nicht abschliessenden Liste mit Punkten, die konzeptuell zu weiterer Forschungsarbeit führen könnten.

- Kompetenzüberprüfung

Ob und in welchem Umfang die Kompetenzen aus [Unterabschnitt 4.1.2](#) von den Lernenden erreicht wurden, sollte für eine Weiterentwicklung vertieft untersucht werden. Allenfalls sollte daraufhin die Unterrichtseinheit angepasst oder die Kompetenzen ergänzt werden. Falls die Kompetenzen erfolgreich erworben werden, sollten diese in den einzelnen Lehrplänen verankert werden.

- Wissenschaftliche Evaluation Unterrichtseinheit

Qualitativ oder quantitativ sollte untersucht werden, wie erfolgreich die Unterrichtseinheit zum Thema «Netzwerk» auf der Zielstufe Sek 1 eingesetzt werden kann.

- Weitere Themen erschliessen

Mit der Verbindung von «Scratch» mit dem «Raspberry Pi» erschliessen sich viele weitere Themen, die sich durch die Ergänzung mit neuer Hardware ergeben. Mit einem «Raspberry Pi» können beispielsweise Motoren verknüpft werden und über Scratch-Blöcke könnten entsprechende Befehle zur Ansteuerung der Motoren weitergeleitet werden. Werden zusätzlich neben Aktoren noch Sensoren eingesetzt, kann ein funktionierender Roboter zusammengesetzt und selbstständig programmiert werden.

5.2 Fazit

Das Fazit gibt einen Rückblick auf die eingangs erwähnten Forschungsfragen ([Abschnitt 1.2](#)), die im Laufe dieser Arbeit beantwortet werden konnten, sowie einen abschliessenden Ausblick.

5.2.1 Neue Technologie

Welche neuen Ansätze für die Erschaffung von Netzwerken als Unterrichtsgegenstand gibt es aufgrund des stetigen Wandels im Bereich der Technologie?

Der stetige Wandel der Technologie ermöglicht es heute, neue Technologie im Unterricht einzusetzen. Einerseits ist die neue Technologie erschwinglich, andererseits gibt es Vorteile bei ihrem Einsatz, da beispielsweise deren Einrichtung im schulischen Umfeld nicht mit übermäßigem Aufwand verbunden ist. Der Einsatz von «LoRa» in Kombination mit einem «Raspberry Pi» ist neu, sowie ohne grosse Installationsarbeit im Schulhaus möglich. Ein «LoRaWAN»-Netzwerk ist kostenlos und benötigt, abgesehen von den Adaptern, den Microcontrollern und der entsprechenden Software, sowie ein wenig Strom, keine weitere Verkabelung oder Installation von WLAN-Routern und -Verstärkern. Der Einsatz ist sogar ausserhalb des Schulhauses möglich. Solche modernen Technologien sollen nicht nur eingesetzt werden, da sie einfach zu installieren sind. Sie sollen auch eingesetzt werden, damit die Schülerinnen und Schüler möglichst früh mit den neuen Möglichkeiten, die sich durch die neuen Technologien ergeben, in Kontakt treten können. Die hier präsentierten Entwicklungen sind nicht nur dank einer einfachen Infrastruktur fortschrittlich, sie verbinden auch die bekannte Programmierumgebung «Scratch» mit den technisch fortgeschrittenen Geräten und ermöglichen so einen tiefen Einstieg in dieses neue Umfeld.

Ein möglicher Ansatz für die Erschaffung von Netzwerken als Unterrichtsgegenstand ist sicher der hier präsentierte: die «LoRa»-Technologie mit «Physical Computing» verbinden und im Unterricht ohne weitere Infrastruktur einsetzen.

Ähnliche Ansätze gibt es mit den verschiedenen Netzwerktechnologien aus [Tabelle 2.1](#). Dabei ist sicher zu erwähnen, dass der Bereich «Internet of Things» ideal ist, um, unabhängig von bestehender Infrastruktur, neuste Technologie ins schulische Umfeld zu bringen. Eine bekannte Standardanwendung aus dem Bereich «IoT», das Sammeln von Daten in der Natur oder der allgemeinen Umgebung rund ums Schulhaus, kann hier auch als Beispiel für fächerübergreifenden Unterricht erwähnt werden. Im Informatikteil werden Daten gesammelt, die dann im Geografie- oder Biologieteil ausgewertet werden, ist eine mögliche Anwendung für den viel propagierten «Fächerübergreifenden Unterricht» (siehe Caviola 2011).

Allgemeine Ansätze zur Erschaffung von neuen Bereichen aus der Informatik, verknüpft mit neuester Technologie, gehen in die gleiche Richtung. Die Technologie sollte einfach installiert werden können, eine günstige Beschaffung macht den Einsatz für Schulen interessanter und wenn die Bedienung durch die Lernenden so einfach wie möglich gestaltet wird, kann die moderne Technologie den Weg von der Industrie ins Schulhaus schaffen. Ein Problem dabei ist aktuell, dass die neue Technologie je länger je komplexer ist und deshalb kaum von Lehrpersonen selbstständig für den Einsatz im Unterricht vorbereitet wird. Hier sollte die Forschung einspringen und den Einsatz der neuen Technologie für den Unterricht vereinfachen und zugänglich machen. Beispielsweise ermöglicht die Verknüpfung mit der Programmierumgebung «Scratch» den Zugang von neuer Technologie auf jeden Fall für die Lernenden.

5.2.2 Netzwerktechnologie im Unterricht

Welche Bereiche aus der Netzwerktechnologie sind sinnvoll für den Unterricht?

Die Netzwerktechnologie ist eine komplexe Materie aus der Informatik. Das trifft auch auf weitere Technologien oder Sachverhalte aus der realen Welt zu. Solche komplexen Themen müssen für den Unterricht vereinfacht und heruntergebrochen werden. Meist ist es sinnvoll,

dass einzelne Unterthemen ausgeklammert sowie andere nochmals vereinfacht werden, um sie mit den Lernenden zu behandeln. Aus dem Gebiet der Netzwerke sind die Inhalte der Netzwerkarchitektur, der Protokolle und der Adressierung von Geräten in der Schule behandelbar und können gezielt behandelt werden. Welche Geräte müssen vorhanden sein, damit ein Netzwerk erstellt werden kann, wie können die Geräte einander erreichen sowie miteinander kommunizieren und in welcher Form werden einzelne Nachrichten übermittelt, damit sie vom gewählten Gegenüber auch gelesen werden können? Mit der vorgeschlagenen Unterrichtseinheit (siehe Kapitel 4) werden genau diese Themen behandelt. Nebenher erfahren die Schülerinnen an einem vereinfachten Beispiel, wie die Kommunikation ihrer täglich genutzten Technologie funktioniert.

5.2.3 Netzwerke im Zyklus 3

In welcher Form sollen Schülerinnen und Schüler des Zyklus 3 die Netzwerktechnologie kennenlernen?

Die Lerninhalte der Schule sind spiralförmig angeordnet und bauen jeweils auf zuvor Gelerntem auf. Netzwerke erscheinen im Lehrplan 21, der die Kompetenzen des Zyklus 1 bis Zyklus 3 definiert, nur an wenigen Orten. Die erwähnten Kompetenzen handeln dann auch nur entfernt von der Technologie hinter den Netzwerken, sondern befassen sich beispielsweise mit dem Speichern von Dateien an verschiedenen Orten und setzen den Fokus darauf, dass Dateien entweder lokal auf dem Computer oder ortsunabhängig in einem Netzwerk abgespeichert werden können. Nach dem Zyklus 3 werden die Inhalte an fortführenden Schulen beispielsweise vom Rahmenlehrplan fürs Gymnasium definiert. Mit dem Spiralcurriculum sollten sich neue Inhalte, in diesem Fall die technischen Grundlagen von Netzwerken, an den später auftauchenden Inhalten orientieren. Die Netzwerkgrundlagen für den Zyklus 3 wurden deshalb auf einfache Protokolle, dem Übermitteln von Nachrichten und dem Adressieren von Geräten im Netzwerk eingeschränkt. Diese Inhalte decken sich auch mit Lehrplänen aus den Vereinigten Staaten. Andere Länder behandeln das Thema «Netzwerke» ähnlich dürftig wie der Lehrplan aus der Schweiz.

5.2.4 Netzwerktechnologie erleben

Wie kann die Netzwerktechnologie vereinfacht und eingebettet werden, so dass der Unterrichtsgegenstand enaktiv erlebt werden kann?

Sobald Lernende den Lerngegenstand aktiv erleben, erlernen sie die Inhalte besser und behalten diese länger. Sie kennen nicht nur die Netzwerktechnologie, sondern können die Technik auch selber anwenden. Dieses enaktive Erlebnis wird mit der vorgeschlagenen Unterrichtseinheit ermöglicht. Die Schülerinnen und Schüler sehen direkt, wie ihre Nachrichten im Netzwerk auftauchen. Sie können direkt versuchen, andere Nachrichten abzufangen und zu interpretieren. Ihre eigenen Programme können direkt auf Nachrichten reagieren, Antworten senden und eine bilaterale Kommunikation starten. Durch das Programmieren der einzelnen Teilschritte in Scratch können die Schülerinnen und Schüler in bekannter Umgebung mit neuen Inhalten arbeiten. Wenn normalerweise komplexe Programme erstellt werden müssen, damit einzelne Komponenten über «LoRa» miteinander kommunizieren können, wurde die Technik so erweitert, dass via einzelne »Scratch«-Blöcke die «LoRa»-Komponenten am «Raspberry Pi» gesteuert werden können. Die Lernenden sehen durch die Erweiterung mit dem «Watcher»-Gerät nicht nur, dass ihre Nachrichten direkt versendet wurden, sondern erleben auch gleich, dass eine Kommunikation in einem offenen Netzwerk grundsätzlich offen ist und von allen teilnehmenden Geräten mitgelesen werden kann.

5.2.5 Ausblick und offene Fragen

Mit einfachen und verständlichen «Scratch»-Befehlen können Nachrichten unter Einhaltung eines bestimmten Protokolls an Geräte aus dem Netzwerk versendet werden und von diesen Geräten ausgelesen werden. Die Lösung im Hintergrund besteht aus der bereits angesprochenen Dreiecksbeziehung zwischen «Scratch», «Python» und «LoRa», verknüpft durch eine «WebSocket»-Verbindung. Diese drei Komponenten sind, einzeln betrachtet, jeweils relativ trivial. In ihrem Zusammenspiel, das wohl komplexer ist, als es sich hier liest, ermöglicht die Lösung ein enaktives Erlebnis für die Zielstufe, das bereits mit einfachen Vorkenntnissen erlebt werden kann. Das textbasierte Programmieren von Python fällt weg, das Aufsetzen eines Netzwerks mit Routern, Computern im Netzwerk und deren Verbindung über IP-Adressen muss nicht durchgeführt werden und, im Gegensatz zu vielen Computeranwendungen, ist das finale Produkt überall einsetzbar, wo ein Stromanschluss vorhanden ist.

Die direkte Verknüpfung von «Scratch» mit einem «Raspberry Pi» ist im Bereich vom «Physical Computing» neu und definitiv nicht nur auf die hier ausgeführte «LoRa»-Anwendung beschränkt. So können grundsätzlich alle Aktoren und Sensoren, die mit einem «Raspberry Pi» verbunden werden können, systemunabhängig auch mit «Scratch» verbunden werden. Ein weiterer Schritt in der Welt von «Physical Computing», das im schulischen Umfeld erfolgreich eingesetzt wird, ist gemacht, darauf aufbauend folgen hoffentlich weitere.

Anhang A

Auflistung Hardware

A.1 Endgerät «Scratch»

Mit diesem Endgerät können die Schülerinnen und Schüler ihr eigenes Scratch-Programm schreiben und damit Nachrichten im Netzwerk versenden und empfangen. Für die Unterrichtsversuche wurden vier Geräte vorbereitet. Jede einzelne Komponente wird nur einmal benötigt pro Gerät.

Komponente	Preis	Bezugsquelle
Raspberry Pi Raspberry Pi 4 / 4GB / 1.5 GHz	CHF 72.80	distrelec.ch/de/p/30152781
Speicherkarte Sandisk ultra / MicroSD / 64GB	CHF 9.42	reichelt.com: SDSQUA4064GGN6FA
LoRa Hat Adafruit Radio Bonnet 4074 / 915MHz	CHF 30.15	adafruit.com/product/4074
LoRaWAN Antenne Adafruit 3340 / 900MHz	CHF 15.38	adafruit.com/product/3340
Touchscreen 10.1" Sunfounder 10.1 / HDMI / 1280x800px	CHF 113.10	amazon.de/dp/B08VNNLHVN

TABELLE A.1: Materialliste Endgerät «Scratch».

A.2 Endgerät «Watcher»

Auf diesem Endgerät werden die versendeten Inhalte aus dem Netzwerk dargestellt. Für die Unterrichtsversuche wurde ein einzelnes Gerät vorbereitet. Jede einzelne Komponente wird einmal benötigt.

Komponente	Preis	Bezugsquelle
Raspberry Pi Raspberry Pi 4 / 4GB / 1.5 GHz	CHF 72.80	distrelec.ch/de/p/30152781
Speicherkarte Sandisk ultra / MicroSD / 64GB	CHF 9.42	reichelt.com: SDSQUA4064GGN6FA
LoRa Hat Adafruit Radio Bonnet 4074 / 915MHz	CHF 30.15	adafruit.com/product/4074
LoRaWAN Antenne Adafruit 3340 / 900MHz	CHF 15.38	adafruit.com/product/3340
HDMI Adapter Micro HDMI zu HDMI	CHF 2.72	seeedstudio.com: 320210001
HDMI Kabel	-	
Bildschirm oder Beamer	-	

TABELLE A.2: Materialliste Endgerät «Watcher».

Anhang B

Auflistung Software

B.1 Endgerät «Scratch»

Auf den Endgeräten der Schülerinnen und Schülern muss die eigens erweiterte Scratch-Anwendung installiert sein und gestartet werden. Zudem benötigt es den eigens erstellten Python-Server, der für die Kommunikation zwischen Browser-Fenster (Scratch-Anwendung) und der LoRa-Erweiterung für den Raspberry Pi sorgt.

Programm	Bezugsquelle
Raspberry Pi OS (64-bit) Betriebssystem	raspberrypi.com/software/operating-systems
Scratch Web-Anwendung für Raspberry Pi	github.com/btemperli/scratch-raspberry
Scratch Server Python-Server für die Kommunikation	github.com/btemperli/raspberry-scratch-server
Raspberry Image «Scratch» Vorbereitetes Abbild für SD-Karten	temperli.io/master/images/scratch

TABELLE B.1: Software Endgerät «Scratch».

Mit dem Raspberry Image «Scratch» ist die hier aufgelistete Software vorbereitet und bereits fertig konfiguriert. Das Image kann auf eine handelsüblichen SD-Karte geladen werden und direkt mit einem «Raspberry Pi» verwendet werden. Zusätzlich wird nur die weitere Hardware aus [Abschnitt A.1](#) benötigt.

B.2 Endgerät «Watcher»

Auf dem Endgerät «Watcher» muss nur der Server für die Kommunikation zwischen LoRa-Erweiterung und Raspberry Pi installiert werden. Der Server wird ergänzt mit einer einfachen Ausgabe für die einfache Übersicht der verschiedenen verschickten Netzwerkdaten.

Programm	Bezugsquelle
Raspberry Pi OS (64-bit) Betriebssystem	raspberrypi.com/software/operating-systems
Scratch Server Python-Server für die Kommunikation	github.com/btemperli/raspberry-scratch-server
Raspberry Image «Watcher» Vorbereitetes Abbild für SD-Karten	temperli.io/master/images/watcher

TABELLE B.2: Software Endgerät «Scratch-Device».

Mit dem Raspberry Image «Watcher» ist die hier aufgelistete Software vorbereitet und bereits fertig konfiguriert. Das Image kann auf eine handelsüblichen SD-Karte geladen werden und direkt mit einem «Raspberry Pi» verwendet werden. Zusätzlich wird nur die weitere Hardware aus [Abschnitt A.2](#) benötigt.

Anhang C

Unterrichtsmaterialien

C.1 Aufgabenkarten & Hilfestellungen

Version 3 vom 19. Juli 2022.

Bildquelle, falls nicht anders angegeben: Eigene Darstellung.



ABBILDUNG C.1: WhatsApp-Icon Quelle: Wikipedia, Autor «WhatsApp», commons.wikimedia.org/wiki/File:WhatsApp.svg


ABBILDUNG C.2: SnapChat-Icon Quelle: Wikipedia, Autor «Saucy», en.wikipedia.org/wiki/File:Snapchat_logo.svg

ABBILDUNG C.3: Mobiltelefon mit einigen Apps Quelle: Christian Wiediger, «Unsplash», unsplash.com/photos/ZYLmudR28SA

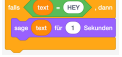


2 Hilfe


Nachricht empfangen



Text überprüfen



Text vergleichen



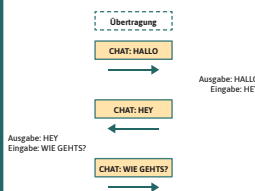
3 Chat

a) Entwickle ein System auf dem Papier, um mit einem anderen Gerät einen Chat zu starten. Besprich dein System mit dem anderen Team.

b) Programmiere nun einen Chat, mit dem du Nachrichten an ein anderes Gerät senden und von diesem Gerät empfangen kannst.

Ergänze dein Programm so, dass es nur auf die Chat-Nachrichten vom anderen Gerät reagiert.

3 Hilfe



4 Netzwerk

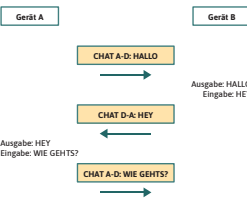
a) Überlege gemeinsam mit allen Anderen, wie ihr Geräte identifizieren könnt.

b) Programmiere nun einen Chat, mit dem du Nachrichten an ein beliebiges Gerät senden kannst. Empfange Nachrichten nur von diesem Gerät.

c) Erweitere das Programm so, dass du einen Chat beenden kannst.

d) Nachdem ein Chat beendet wird, sollen neue Chats möglich sein.


4 Hilfe



A Hinweis


Damit du sicher bist, dass dein Gerät eine Verbindung zum Netzwerk hat, kannst du Nachrichten immer wieder verschicken.

Damit dabei aber nichts überlastet wird, lässt du dem Programm aber besser etwas Zeit dafür. Du kannst das Programm jeweils eine Sekunde warten lassen.




B Hinweis

Eingehende Nachrichten erhältst du über den Block «Nachricht».



Dabei wird jeweils die letzte Nachricht ausgegeben. Damit du die Nachricht weiter verwenden kannst, musst du diese als Variable zwischenspeichern.

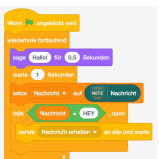
Beim erneuten Auslesen wird nämlich bereits die nächste Nachricht ausgegeben.



C Hinweis

Du kannst bestimmte Programmteile separat ausführen, wenn ein bestimmtes Ereignis auftritt. Das kannst du selber steuern mit dem Block «sende ____ an alle».

Dein Hauptprogramm kann in dieser Zeit blockiert werden.



D Hinweis

Damit du auf Nachrichten reagieren kannst, brauchst du den «frage und warte»-Block.



C.2 Kommentar für Lehrpersonen

Version 2 vom 20. Juli 2022.

UNTERRICHTSEINHEIT

NETZWERKE

KOMMENTAR FÜR LEHRPERSONEN

Masterarbeit Beat Temperli

Einleitung



«Der mediale Alltag von Jugendlichen ist schon seit Längerem durch das Handy und das Internet geprägt. In der Schweiz sind kaum noch Jugendliche zu finden, die nicht täglich ein Handy und das Internet nutzen.»

JAMES-Studie 2020

«In einer Zeit, wo das Internet so alltäglich, präsent und vor allem gesellschaftsverändernd ist, muss dieses Themengebiet in der Schule aufgegriffen werden.»

Modrow & Strecker, 2016

Internet und Netzwerke

Die Schülerinnen und Schüler bewegen sich täglich im Internet. In der Schule ist das sicher nicht täglich ein Thema.

Im Lehrplan sind in den Kompetenzen bei der Medienbildung und in der Informatik die Themen «Internet» und «Netzwerk» an wenigen Stellen abgebildet. Dabei geht es aber hauptsächlich um die Anwendungen aus dem Internet, die in der Schule genutzt werden, beispielsweise das Speichern von Dateien auf der Cloud.

Die technologischen Hintergründe, beispielsweise wie eine WhatsApp-Nachricht den Weg vom Mobiltelefon auf das Mobiltelefon eines Freundes findet, ist weder im Lehrplan noch in den Lehrmitteln abgebildet.

Physical Computing

Computer in die Hand nehmen, ausprobieren, eigene Projekte damit umsetzen, aber auch bereits Computer anfassen und selber zusammenbauen oder programmieren wird unter dem Begriff «Physical Computing» zusammengefasst. Die Lernenden erleben direkt, ob ihre Programme funktionieren und sehen, was der Computer macht. Sie «be-greifen», wie der Computer funktioniert. Der Bereich Physical Computing ist je länger je mehr ein Thema im Informatik- oder Projektunterricht.

Die Programmiersprache «Scratch» kennen die Schülerinnen und Schüler allenfalls bereits aus der Primarstufe, oder auch aus anderen Informatiklektionen. Sie ist optimal für Einsteigerinnen und Einsteiger und hilft beim Erlernen der Grundfertigkeiten im Programmieren. Die Blockprogrammierung verhindert allgemeine Programmierfehler bereits zu Beginn und ist für den Einsatz an der Schule ausgelegt.

Diese vorliegende Unterrichtseinheit verknüpft die Bereiche «Physical Computing» mit «Netzwerken». Schülerinnen und Schüler können ihre eigene Kommunikation programmieren, direkt in der Programmierumgebung «Scratch».

Sachanalyse



Netzwerktheorie

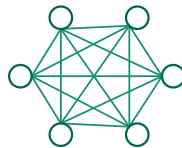
Bei der Netzwerktheorie eignen sich 3 Bereiche für den Unterricht auf der Stufe Zyklus 3:

- Netzwerkarchitektur
- Protokolle
- Adressierung einzelner Geräte

Netzwerkarchitektur

Bei der Netzwerkarchitektur ist die Grundvoraussetzung, dass einzelne Geräte miteinander verbunden sind, nur so können sie miteinander kommunizieren. In unserer Unterrichtseinheit sind die Geräte über Funk miteinander verbunden, das heisst die Geräte dürfen nicht zu weit auseinander entfernt sein, damit sie miteinander Daten austauschen können.

Bei der Topologie von Netzwerken gibt es verschiedene Strukturen. Im Internet ist die gängige Struktur eine hierarchisch aufgebaute Sternstruktur, wobei im Zentrum des Sterns meist der «Router» liegt, die anderen Geräte sind mit dem Router verbunden, welcher das ganze Netzwerk zusammenhält.



Damit die Geräte alle auch als Vermittler fungieren könnten, ist unsere Struktur nach dem Prinzip «Fully Connected» angeordnet, sprich es sind alle Geräte mit allen anderen Geräten verbunden und es gibt keine Hierarchie. So kann jedes Gerät mit jedem anderen Gerät kommunizieren.

Adressierung

Die Adressierung von Geräten in einem Netzwerk funktioniert mit den IP-Adressen, dies basiert auf dem «Internet Protocol». Darin wird definiert, wie die Verteilung der IP-Adresse funktioniert und welche Unterscheidungen damit gemacht werden können. Die IP-Adressen werden aktuell in zwei Versionen verwendet, IPv4 ermöglicht den Einsatz von etwa 4 Milliarden Geräten, IPv6 erhöht diese Zahl enorm. Aktuell findet der Wechsel von IPv4 zu IPv6 statt.

In unserem Netzwerk dürfen die Lernenden ihr eigenes System für eine Adressierung verwenden, welches nach einem ähnlichen Prinzip wie bei der Verteilung von IP-Adressen funktionieren wird:

- Geräte erhalten eine bestimmte Nummer (z.B. 0-9 oder A-Z)
- Geräte behalten bei einer neuen Einwahl ins Netzwerk ihre Nummer
- Nummern werden, sofern möglich, nur einmal vergeben
- Wenn alle Nummern besetzt sind, kann sich kein neues Gerät im Netzwerk einwählen

Protokolle

Protokolle definieren, wie in einem Netzwerk Daten, insbesondere Meta-Daten, versendet werden. So kann eine übermittelte Botschaft nicht nur die Nachricht enthalten, sondern auch mit Informationen zum Sender und Empfänger versehen werden. Aus der Nachricht «Hallo Freund» wird mit einem Protokoll die Botschaft «A-B-Hallo Freund». Das entwickelte Protokoll definiert in diesem Fall, dass das Gerät «A» vom Gerät «B» die Nachricht «Hallo Freund» erhält. Dieses sehr einfache Protokoll wird definiert, indem vor jede Nachricht zuerst der Empfänger und dann der Absender hinzugefügt wird.

Sachanalyse



LoRa

LoRa ist eine Funktechnologie für die Übertragung von Daten. Diese Datenübertragung wurde entwickelt für den Einsatz in Geräten vom «Internet of Things». Kleine Datenmengen können mit wenig Energie über weite Distanzen übertragen werden, so dass aus der Umwelt gesammelte Daten über mehrere Kilometer hinweg an andere Geräte übertragen werden können. Diese kleinen «Internet of Things»-Geräte können mit einer Batterie betrieben werden, senden von Zeit zu Zeit Daten und überleben ohne weitere Stromversorgung über mehrere Monate oder Jahre. Die LoRa-Technologie funktioniert ähnlich wie die Übertragung in einer WLAN-Umgebung.

Die Geräte der Unterrichtseinheit sind mit LoRa-Adaptoren ausgestattet und übertragen die Daten über ein eigenes LoRa-Netzwerk.

Scratch

Scratch ist eine Programmierumgebung, die das Erlernen des Programmierens vereinfacht. Im Gegensatz zur textbasierten Programmierung ist die grafische Programmierung greifbarer für Programmieranfänger. Scratch ist eine Lernumgebung, entwickelt vom MIT, die gängige Anfängerfehler, wie beispielsweise falsch gesetzte Kommata, von Beginn an verhindert. Mit den Scratch-Blöcken können einzelne Befehle auf einer grafischen Ebene zusammengesetzt und zu einem funktionierenden Programm verknüpft werden. Die Umgebung ist kinderfreundlich gestaltet und ermutigt die Lernenden zum Ausprobieren, Fehlschlagen und erneut Starten sowie dem erfolgreichen Umsetzen von eigenen Ideen.

Raspberry Pi

Die Microcontroller sind kleine Computer, die mit einem Linux-Betriebssystem ausgestattet wurden. Abgesehen von wenig Prozessor-Leistung, weniger Speicherplatz und der fehlenden Peripherie unterscheiden sie sich nicht von «normalen» Computern.

Die «Raspberry Pi» eignen sich gut für den Einsatz im schulischen Umfeld und im Bereich von «Physical Computing», sie können beliebig mit Sensoren, Aktoren oder weiteren Geräten ausgestattet werden und individuell programmiert werden.

Die Geräte der Unterrichtseinheit sind mit einem Touchscreen und den für die Kommunikation benötigten LoRa-Adaptoren ausgestattet. Auf den Geräten ist die bekannte «Scratch»-Umgebung installiert, welche in jedem Browser gestartet werden könnte und den Kindern als Programmierumgebung bereits bekannt sein könnte.

Im Hintergrund ist die Scratch-Umgebung mit den LoRa-Adaptoren verbunden, so dass via «Scratch»-Blöcke direkt kommuniziert werden kann.

Kompetenzen



Kompetenzerwerb

Die folgenden Kompetenzen werden von den Lernenden erworben.

1. Die Lernenden verstehen, dass Geräte nur miteinander kommunizieren können, sofern sie über ein gemeinsames Netzwerk miteinander verbunden sind.
2. Die Lernenden sind in der Lage, eine eigene «Sprache» zu entwickeln, damit Geräte miteinander kommunizieren können. Dabei definieren sie einzelne spezifische Meta-Informationen, die nötig sind, damit sich die Geräte verstehen. Die Kommunikationsdaten werden mit den Meta-Informationen ergänzt.
3. Die Lernenden können ein System entwickeln, damit Geräte eindeutig in einem Netzwerk von mehreren Geräten voneinander unterschieden werden können.
4. Die Lernenden können ein eigenes, theoretisches Modell für die einfache Kommunikation zwischen zwei Geräten in der Praxis mit Hilfe der Programmierumgebung «Scratch» umsetzen.
5. Die Lernenden können Datenströme, transferiert mit einem bereits bekannten Protokoll, aus einem Netzwerk analysieren und herauslesen, welche Informationen von fremden Geräten versendet werden.
6. Die Lernenden haben eine Vorstellung davon, wie «echte» Geräte in einem «echten» Netzwerk über «echte» Protokolle miteinander kommunizieren. Sie können ihr eigenes «Modell» der Kommunikation auf andere Kommunikationsnetzwerke transferieren.

Einbettung im Lehrplan 21

Die oben aufgeführten Kompetenzen sind so im Lehrplan 21 nicht zu finden. Verwandte Kompetenzen aus dem Bereich Internet & Netzwerk sind die folgenden:

MI 2.3.j: SuS können lokale Geräte, lokales Netzwerk und das Internet als Speicherorte für private und öffentliche Daten unterscheiden.

MI 2.3.m: SuS können das Internet als Infrastruktur von seinen Diensten unterscheiden (z.B. WWW, E-Mail, Internettelefonie, Soziale Netzwerke).

MI 2.3.b: SuS können sich mit eigenem Login in einem lokalen Netzwerk oder einer Lernumgebung anmelden.

MI 2.3.n: SuS können die Risiken unverschlüsselter Datenübermittlung und -speicherung abschätzen.

Ablauf



Grober Ablauf

Der grobe Ablauf der Unterrichtseinheit bildet sich aus den folgenden 5 Punkten. Je nach Vorwissen der Schülerinnen und Schüler kann der erste Punkt weggelassen werden.

1. Kennenlernen von Scratch.
2. Kennenlernen der neuen Scratch-Blöcke zum Senden von Nachrichten.
3. Kennenlernen der neuen Scratch-Blöcke zum Empfangen von Nachrichten.
4. Nachrichten spezifisch an ein anderes Gerät senden, bilaterale Kommunikation aufbauen. Dazu muss ein einfaches Protokoll entwickelt werden.
5. Das Netzwerk wird so erweitert, dass mit verschiedenen Geräten kommuniziert werden kann. Dazu muss das Protokoll aus den vorangehenden Aufgaben allenfalls erweitert werden.

Für die Punkte 2 bis 5 ist der folgende Ablauf empfohlen und getestet. Die einzelnen Punkte können mehr Zeit in Anspruch nehmen, je nach Level der Klasse.

Für die vier Aufgabenteile sollten mindestens drei Lektionen eingeplant werden.

Zeit	Form	Aktivität	Material
00:00	Input	Was braucht es, damit ich vom Handy Nachrichten verschicken kann?	
00:10		Verteilen Geräte, Gruppeneinteilung	
00:15		Inbetriebnahme Geräte, Erklärung Funktion	4x Scratch-Geräte, 1x Watcher
00:25	ind. GA	Aufgabe 1: Senden	Scratch-Anwendung
00:35	ind. GA	Aufgabe 2: Empfangen	
00:45	ind. GA	Aufgabe 3: Erarbeitung Protokoll & einfacher Chat	
01:30	Input	Wie funktionieren Protokolle?	
01:40	ind. GA	Aufgabe 4: Klassen-Chat	

Der Input «Wie funktionieren Protokolle?» kann bereits relativ früh nach dem Start der Aufgabe 3 eingebracht werden.

Aufgaben



Gruppeneinteilung

Die Aufgaben können von den Lernenden selbstständig gelöst werden. Pro «Scratch»-Gerät benötigt man ein Aufgabenset mit den Aufgaben-, Hinweis- und Hilfekarten.

Die Schülerinnen und Schüler arbeiten optimalerweise in Paaren, Gruppen von drei oder vier Personen sind ebenfalls möglich.

Das System vom «Pair Programming» funktioniert so, dass eine Person am Computer arbeitet und diesen bedient, die andere Person denkt laut mit und gibt Anweisungen, bedient aber selber nicht den Computer. Von Zeit zu Zeit werden die beiden Rollen «Bedienerin» und «Beraterin» getauscht.

Es empfiehlt sich, das System «Pair Programming» bereits im früheren Unterricht einzuführen.

Aufgabenkarten

Für die vier Aufgaben gibt es je eine Aufgabenkarte, die von den Lernenden eigenständig, aber in der richtigen Reihenfolge, bearbeitet werden sollen.

Hilfestellungen

Zu jeder Aufgabe hat es eine kleine Hilfestellung. Mit der jeweiligen Hilfestellung sollten bereits viele Probleme eigenständig gelöst werden können.

Zusätzlich zu den Hilfestellungen gibt es vier Hinweiskarten. Diese Hinweise sind relevant für die Programmierung mit «Scratch» und sollten von den Lernenden berücksichtigt werden.

Die Aufgaben sind bewusst offen formuliert und lassen verschiedene Lösungswege zu und ermöglichen den Lernenden, eigene, kreative Lösungen zu entwickeln. Dazu gehört Ausprobieren, Scheitern und erneut Ausprobieren, bis das gewünschte Ziel erreicht wurde. Unterstützung der Lernenden in diesem Prozess ist viel hilfreicher als Unterstützung bei der Lösungsfindung.

Für grössere Probleme sollte die Lehrperson trotzdem zur Verfügung stehen.

Intervention



Nachrichten versenden vom Mobiltelefon?

Die Schülerinnen und Schüler senden täglich viele Nachrichten. Dabei entscheiden sie, welche Nachricht an Freunde, welche an die Lehrperson und welche in den Klassenchat gesendet wird.

Wie weiss aber die Nachricht, wohin sie gesendet wird?

Dieser Frage werden wir in der Unterrichtseinheit nachgehen und genau das herausfinden.

Lösung

Mit der Nachricht werden diverse weitere Informationen versendet. Von wem kommt die Nachricht, an wen wird die Nachricht versendet. Weiter wird die Nachricht wahrscheinlich verschlüsselt, so dass niemand diese abfangen und die Kommunikation zwischen mehreren Personen mitlesen kann.

Wie funktionieren Protokolle?

Protokolle definieren bekannte Abläufe, zum Beispiel wie Daten im Internet versendet werden. Protokolle kennen die Schülerinnen und Schüler aber bereits aus ihrem Alltag.

Beispiel 1: Telefonieren

1. Person A: gibt die Nummer ein
2. Person A: startet den Anruf
3. Person B: nimmt den Anruf entgegen
4. Person B: «VornameB, Hallo»
5. Person A: «Hallo VornameB, da ist VornameA»
6. Person A: startet Gespräch

Beispiel 2: Begrüssung mit Handschlag

1. Person A: hält Hand hin
2. Person B: ergreift Hand
3. Man begrüsst sich
4. Man lässt die Hände wieder los

Als Klasse soll nun definiert werden, wie eine Kommunikation zwischen zwei Geräte ablaufen soll und wie diese *genau* definiert wird.

Lösung

Nachrichten können mit dem Empfänger und dem Absender versehen werden, immer nach dem gleichen Schema: *Empfänger - Absender - Nachricht*.

Dabei werden die Geräte mit Buchstaben versehen, so dass eine Nachricht beispielsweise wie folgt aussieht:

A-B-Hallo ihr vier!

Diese Nachricht kommt also von Gerät B und wird an Gerät A gesendet.

Lösungsvorschläge











Lösungsvorschläge
Dieser Lösungsvorschlag basiert auf dem Protokoll «Empfänger»-«Absender»-«Nachricht».



VIEL ERFOLG BEIM UMSETZEN!

Kontakt

Beat Temperli

beat@temper.li

<https://temperli.io/master>

(c) Beat Temperli 2022

Anhang D

Auswertung

D.1 Rückmeldungen Unterrichtseinheit «Netzwerk»

ID	Wie zufrieden bist du mit deinem Scratch-Programm?	Wie schwierig waren die Aufgaben? (1 = super einfach, 10 = super schwierig)	Konntest du alle Aufgaben lösen?
1		4	10 Nein
2		5	7 Fast alle
3		5	7 Fast alle
4		6	6 Fast alle
5		4	5 Fast alle
6		5	6 Fast alle
7		6	6 Fast alle
8		6	7 Ja
9		3	6 Fast alle
10		5	3 Ja
11		5	7 Fast alle
12		4	7 Fast alle

ID	Was hat funktioniert?	Was hat nicht funktioniert?	Hast du dein Programm alleine oder in Gruppen bearbeitet?
1	Das sich die Nachrichten von alleine senden	Es ist immer wieder mal abgestürzt	zu zweit
2		Das Scratch hat eigentlich gut funktioniert.	zu zweit
3			zu dritt
4			zu dritt
5			zu zweit
6	1-2	3	zu viert
7	Die Gruppenarbeit	Das Gerät ist oft abgestürzt	zu zweit
8	Das hin und her schicken von Nachrichten (das Netzwerk)	Manchmal ist unser Gerät wegen Überhitzung abgestürzt sonst nichts	zu zweit
9	ins netz zu gehen	der kleine speicherplatz	zu dritt
10	Eigentlich alles	Zuerst hatten wir Probleme damit, dass es nur bestimmte Nachrichten anzeigt, welche z.B. ein Z davor hatten.	zu dritt
11	Es hat funktioniert, dass eine Gruppe die Chats der anderen Gruppe empfangen konnte und man etwas schreiben konnte, welches dann im Netzwerk erschienen ist.	Am Anfang hatten wir Probleme, dass die eine Gruppe die Chats der anderen empfangen konnte, später hat es dann funktioniert.	zu zweit
12	Worte auf die Leinwand schreiben, von anderen Geräten Nachrichten erhalten, auf dem kleinen Bildschirm hinten Worte erscheinen lassen	Nur von einem anderen Gerät Nachrichten zu erhalten	zu viert

ID	Welche Aufgaben hast du übernommen?	Dein Geschlecht	Die Geschlechter in der Gruppe
1	Beraten;	weiblich	weiblich
2	Programmieren am Bildschirm; Mitdenken & Überlegen;	männlich	männlich
3	Mitdenken & Überlegen; Programmieren am Bildschirm; Beraten;	weiblich	weiblich
4	Programmieren am Bildschirm; Mitdenken & Überlegen; Beraten;	weiblich	weiblich
5	Programmieren am Bildschirm;	männlich	männlich
6	Programmieren am Bildschirm;	männlich	männlich
7	Mitdenken & Überlegen;	weiblich	weiblich
8	Mitdenken & Überlegen; Beraten; Programmieren am Bildschirm;	männlich	männlich
9	Mitdenken & Überlegen; Beraten;	männlich	männlich
10	Mitdenken & Überlegen;	männlich	männlich
11	Programmieren am Bildschirm;	weiblich	weiblich
12	Mitdenken & Überlegen;	männlich	männlich

ID	Der Name von deinem Programm	Waren die Aufgabenkarten verständlich?	Wo hast du Hilfe geholt während dem Programmieren?
1		Ja	Bei der Lehrperson;Bei einem fremden Team;Im eigenen Team;Hilfe-Seite auf den Aufgabenkarten;
2		Ja	Bei der Lehrperson;
3		Ja	Bei der Lehrperson;
4	Wer bin ich?	Ja	Im eigenen Team;Bei der Lehrperson;
5		Ja	Bei der Lehrperson;
6	Nicht veröffentlicht	Ja	Im eigenen Team;Bei der Lehrperson;
7		Ja	Hilfe-Seite auf den Aufgabenkarten;Bei Herrn Temperli;
8	Wir haben es nicht veröffentlicht wir haben es nur lokal gespeichert	Ja	Im eigenen Team;Bei der Lehrperson;
9		Ja	Im eigenen Team;
10	Haben wir nur lokal gespeichert und nicht veröffentlicht.	Ja	Bei der Lehrperson;
11		Ja	Bei der Lehrperson;Hilfe-Seite auf den Aufgabenkarten;
12		Ja	Hilfe-Seite auf den Aufgabenkarten;Im eigenen Team;Bei der Lehrperson;

ID	Welche Hilfen waren nützlich?	Welche Hilfe-Seiten auf den Aufgabenkarten haben dir geholfen?	Was war schwierig?
1	Alle	Aufgabe 1: Senden;Aufgabe 2: Empfangen;	Ab Aufgabe 3 kam ich nicht mehr ganz mit
2	Die Lehrperson hat uns sehr gut geholfen.	Aufgabe 3: 1 zu 1 Chat;	Die letzten Aufgaben
3		Alle;	
4			
5		Aufgabe 4: Chat mit Protokoll;	
6	Von der Lehrperson und vom Team.	Aufgabe 2: Empfangen;	Die Aufgabe 3 zu verstehen.
7	Die mit den Karten fand ich sehr hilfreich	Aufgabe 2: Empfangen;Aufgabe 3: 1 zu 1 Chat;	
8	Die Lehrperson und mein Team	Aufgabe 2: Empfangen;Aufgabe 3: 1 zu 1 Chat;	Das hin und her schicken von Nachrichten
9	team, lehrperson	Aufgabe 2: Empfangen;Aufgabe 3: 1 zu 1 Chat;	
10	Die Tips der Lehrperson wenn wir etwas nicht gefunden haben, da wir noch nicht lange mit Scratch arbeiten.	Aufgabe 3: 1 zu 1 Chat;Aufgabe 4: Chat mit Protokoll;	Mit der eher schlechten Hardware auszukommn.
11	Die Hilfe-Karten waren sehr hilfreich und wenn wir nicht 100% sicher waren, haben wir die Lehrperson gefragt, dann sind wir sicher draus gekommen.	Aufgabe 2: Empfangen;	
12	kleine Tipps und auch kleine Hilfen beim Programm	Alle;	Den Programmblock richtig zu erstellen

Was war einfach?		Wie gut kennst du dich mit Netzwerken aus? (1: überhaupt nicht / 6: super gut)	Wie gut kennst du dich mit Netzwerkprotokollen aus? (1: überhaupt nicht / 6: super gut)
ID			
1	Aufgabe 1&2 hab sogar ich einigermaßen hinbekommen (;	1	1
2	Die ersten Aufgaben	4	4
3		4	4
4		5	2
5	programmieren	2	1
6	Die erste Aufgabe	3	2
7		2	2
8	Es war alles sehr anspruchsvoll	5	4
9	ins netz zu chatten	2	1
10	Das Programmieren in der Gruppe.	4	1
11		1	1
12	Aufgabe 1	2	1

Wie gut beherrschst du Scratch? (1: überhaupt nicht / 6: super gut)		Hast du Anmerkungen zu den Beurteilungen aus der letzten Frage?	Ein Netzwerk besteht immer aus mehreren Geräten.
ID			
1	2		Kann man so nicht beurteilen
2	4		Falsch
3	3		Richtig
4	4		Falsch
5	4		Kann man so nicht beurteilen
6	4	Nein	Richtig
7	4		Richtig
8	6		Kann man so nicht beurteilen
9	4		Richtig
10	2		Richtig
11	2		Richtig
12	3	Ich kenne mich zu wenig gut mit Netzwerken aus um alle Fragen beantworten zu können.	Richtig

ID	Ein Netzwerkprotokoll muss allen Geräten im Netzwerk bekannt sein.	Wieso braucht es Protokolle in Netzwerken?	Welche Elemente braucht es in einem Netzwerkprotokoll, damit Nachrichten von einem Gerät zu einem anderen Gerät übermittelt werden können?
1	Richtig		
2			
3	Richtig		
4	Richtig		
5	Richtig		
6	Falsch	Um das genaue format zu definieren.	Sender, Übermittler, Empfänger
7	Richtig		
8	Richtig		
9	Falsch	um es zu kontrollieren das machen menschen überall	
10	Kann man so nicht beurteilen		
11	Kann man so nicht beurteilen		
12			

ID	Bei einer SMS wird neben der Nachricht auch die Nummer von Empfänger und Sender mitgeschickt.	Was für Gefahren könnten in einem Netzwerk lauern?	Wie kannst du dich vor den oben genannten Gefahren schützen?
1	Kann man so nicht beurteilen		
2			
3	Richtig		
4	Falsch		
5	Richtig	Hacker die Nachrichten abfangen.	
6	Richtig	Hacking, Fehlfunktion	Nicht ins Netzwerk gehen
7	Kann man so nicht beurteilen	Man kann gehakt werden	Mit sicheren Passwörtern
8	Richtig	abfangen von Nachrichten	verschlüsselte Nachrichten
9	Richtig	viren, hacker	sich irgendwie absichern
10	Richtig	Leute, die die Nachrichten mitlesen.	Ja
11	Richtig	Man könnte gehackt werden, wenn man auf bestimmte Links drückt.	Man sollte nie auf irgendwelche Links drücken, wenn man den Absender nicht kennt. Man muss aber auch aufpassen bei solchen die man kennt, da die Hacker links über ein anderes Telefon schicken können.
12	Richtig		

ID	Ein Netzwerkprotokoll definiert, in welcher Form Daten übermittelt werden.	Möchtest du nochmals mit Scratch (Programmierumgebung) arbeiten?	Möchtest du nochmals mit diesen Touch-Computern (Display & Raspberry Pi) arbeiten?
1	Richtig	Ja	
2		Nein	Ja
3	Kann man so nicht beurteilen	Ja	Ja
4	Richtig	Nein	Ja
5	Kann man so nicht beurteilen	Ja	Ja
6	Richtig	Ja	Nein
7	Kann man so nicht beurteilen	Ja	Nein
8	Richtig	Ja	Ja
9	Kann man so nicht beurteilen	Ja	Nein
10	Falsch	Ja	Ja
11	Falsch	Ja	Ja
12		Ja	Ja

ID	Um Daten in einem Netzwerk zu übertragen, braucht es immer einen Sender.	Um Daten in einem Netzwerk zu übertragen, braucht es immer einen Empfänger.
1	Richtig	Richtig
2		
3	Kann man so nicht beurteilen	Kann man so nicht beurteilen
4	Falsch	Falsch
5	Richtig	Richtig
6	Richtig	Richtig
7	Falsch	Richtig
8	Richtig	Richtig
9	Richtig	Richtig
10	Richtig	Richtig
11	Kann man so nicht beurteilen	Falsch
12	Richtig	Richtig

Literaturverzeichnis

- Adafruit Industries, LLC (25. Juni 2022). *About Us: Adafruit Industries, Unique & Fun DIY Electronics and Kits*. URL: <https://www.adafruit.com/about> (besucht am 25.06.2022).
- Afari, Ernest und Myint Khine (1. Jan. 2017). „Robotics as an Educational Tool: Impact of Lego Mindstorms“. In: *International Journal of Information and Education Technology* 7, S. 437–442. DOI: [10.18178/ijiet.2017.7.6.908](https://doi.org/10.18178/ijiet.2017.7.6.908).
- Aref, Mohamed und Axel Sikora (Sep. 2014). „Free Space Range Measurements with Semtech Lora™ Technology“. In: *2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems*. 2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems, S. 19–23. DOI: [10.1109/IDAACS-SWS.2014.6954616](https://doi.org/10.1109/IDAACS-SWS.2014.6954616).
- Ariza, Jonathan Álvarez und Sergio González Gil (Mai 2022). „RaspyLab: A Low-Cost Remote Laboratory to Learn Programming and Physical Computing Through Python and Raspberry Pi“. In: *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje* 17.2, S. 140–149. ISSN: 1932-8540. DOI: [10.1109/RITA.2022.3166877](https://doi.org/10.1109/RITA.2022.3166877).
- Australian Curriculum, Assessment and Reporting Authority [ACARA] (2021). *The Australian Curriculum, Year 10, Design and Technologies & Digital Technologies*. URL: https://v9.australiancurriculum.edu.au/f-10-curriculum/learning-areas/design-and-technologies_digital-technologies/year-10 (besucht am 20.06.2022).
- Bandura, Albert und Hainer Kober (1976). *Lernen am Modell: Ansätze zu einer sozial-kognitiven Lerntheorie*. 1. Aufl. Stuttgart: Klett. 229 S. ISBN: 978-3-12-920590-7.
- Barras, Jean-Luc und Dominik Petko (2007). *Computer Und Internet in Schweizer Schulen*. URL: https://www.phsz.ch/fileadmin/autoren/fe_dateien/petko_barras_2007_computer_und_internet_in_schweizer_schulen.pdf.
- bastelgarage.ch (1. Juli 2022). *Dragino LoRa Shield - 868MHz v1.4 - Arduino*. URL: <https://www.bastelgarage.ch/dragino-lora-shield-868mhz-v1-4-arduino> (besucht am 15.07.2022).
- Bisdikian, Chatschik (2001). „An Overview of the Bluetooth Wireless Technology“. In: *IEEE Communications Magazine* 39.12, S. 86–94. ISSN: 1558-1896. DOI: [10.1109/35.968817](https://doi.org/10.1109/35.968817).
- Bleichroth, Wolfgang (1991). „Elementarisierung, Das Kernstück Der Unterrichtsvorbereitung“. In: *NiU Physik* 2.6, S. 4–11.
- Borowski, Christian u. a. (1. Sep. 2011). „Kinder Auf Dem Weg Zur Informatik: Wie Funktioniert Das Internet?“. In: S. 244–253. ISBN: 978-3-86877-009-4.

- Brinda, Torsten und Friedericke Braun (2017). *Schülervorstellungen im Zusammenhang mit Smartphones*. Gesellschaft für Informatik, Bonn. ISBN: 978-3-88579-668-8. URL: <http://dl.gi.de/handle/20.500.12116/4305>.
- Bruner, Jerome Seymour (1966). *Toward a Theory of Instruction*. Harvard University Press. 196 S. ISBN: 978-0-674-89701-4.
- (1970). *Der Prozess Der Erziehung*. Sprache Und Lernen. Berlin und Düsseldorf: Berlin Verlag.
- Cabrera, Lautaro, John H. Maloney und David Weintrop (12. Juni 2019). „Programs in the Palm of Your Hand: How Live Programming Shapes Children’s Interactions with Physical Computing Devices“. In: *Proceedings of the 18th ACM International Conference on Interaction Design and Children*. IDC ’19. New York, NY, USA: Association for Computing Machinery, S. 227–236. ISBN: 978-1-4503-6690-8. DOI: [10.1145/3311927.3323138](https://doi.org/10.1145/3311927.3323138).
- Caviola, Hugo (2011). „Wege zum guten fächerübergreifenden Unterricht: ein Handbuch für Lehrpersonen“. In: *Wege zum guten fächerübergreifenden Unterricht ein Handbuch für Lehrpersonen*. Bern: h.e.p. Verlag. ISBN: 978-3-03905-633-0.
- Chakaroy, Alexandra Gendreau u. a. (24. Juni 2021). „Challenges and Unexpected Affordances of Physical Computing Going Remote“. In: *Interaction Design and Children*. IDC ’21: Interaction Design and Children. Athens Greece: ACM, S. 276–282. ISBN: 978-1-4503-8452-0. DOI: [10.1145/3459990.3460711](https://doi.org/10.1145/3459990.3460711).
- D-EDK (2016). *Lehrplan 21. Gesamtausgabe. Bereinigte Fassung*. Luzern. URL: https://v-fe.lehrplan.ch/container/V_FE_DE_Gesamtausgabe.pdf.
- Deering, Steve E. und Alex Conta (Dez. 1998). *Generic Packet Tunneling in IPv6 Specification*. Request for Comments RFC 2473. Internet Engineering Task Force. 36 S. DOI: [10.17487/RFC2473](https://doi.org/10.17487/RFC2473).
- Department for Education (2014). *National Curriculum in England: Design and Technology Programmes of Study*. GOV.UK. URL: <https://www.gov.uk/government/publications/national-curriculum-in-england-design-and-technology-programmes-of-study/national-curriculum-in-england-design-and-technology-programmes-of-study> (besucht am 21.06.2022).
- Devalal, Shilpa und A. Karthikeyan (März 2018). „LoRa Technology - An Overview“. In: *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), S. 284–290. DOI: [10.1109/ICECA.2018.8474715](https://doi.org/10.1109/ICECA.2018.8474715).
- Devine, James u. a. (19. Juni 2018). „MakeCode and CODAL: Intuitive and Efficient Embedded Systems Programming for Education“. In: *ACM SIGPLAN Notices* 53.6, S. 19–30. ISSN: 0362-1340. DOI: [10.1145/3299710.3211335](https://doi.org/10.1145/3299710.3211335).
- Diethelm, Ira und Stefan Zumbrägel (2010). *Wie funktioniert eigentlich das Internet? - Empirische Untersuchung von Schülervorstellungen*. Gesellschaft für Informatik e.V. ISBN: 978-3-88579-262-8. URL: <http://dl.gi.de/handle/20.500.12116/19638>.
- Döbeli Honegger, Beat (16. März 2021). *Lehrmittel «Einfach Informatik»*. mia.phsz.ch. URL: <https://mia.phsz.ch/Sekundarstufe/EinfachInformatik> (besucht am 01.07.2021).

- Eberspächer, Jörg, Hans-Jörg Vögel und Christian Bettstetter (19. Dez. 2013). *GSM Global System for Mobile Communication: Vermittlung, Dienste und Protokolle in digitalen Mobilfunknetzen*. Springer-Verlag. 428 S. ISBN: 978-3-322-94063-6.
- Federli, Vanessa (20. Juni 2022). „Zug: 15-jähriger Schüler hackt Software für Maturaprüfungen“. In: *20 Minuten*. URL: <https://www.20min.ch/story/15-jaehriger-schueler-hackt-software-fuer-maturapruefungen-210517570074> (besucht am 15. 07. 2022).
- Fette, Ian und Alexey Melnikov (Dez. 2011). *Rfc6455: The WebSocket Protocol*. URL: <https://datatracker.ietf.org/doc/html/rfc6455> (besucht am 23. 11. 2021).
- Fischer, Claudia u. a. (2021). *Connected 4 Arbeitsbuch*. 1. Aufl. Zürich: Lehrmittelverlag Zürich. ISBN: 978-3-03713-779-6.
- Fraser, Neil (22. Okt. 2015). „Ten Things We’ve Learned from Blockly“. In: *Proceedings of the 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*. BLOCKS AND BEYOND '15. USA: IEEE Computer Society, S. 49–50. ISBN: 978-1-4673-8367-7. DOI: [10.1109/BLOCKS.2015.7369000](https://doi.org/10.1109/BLOCKS.2015.7369000).
- Freischlad, Stefan (1. Jan. 2022). *Filius*. URL: <https://www.lernsoftware-filius.de/> (besucht am 15. 07. 2022).
- Gramm, Andreas, Malte Hornung und Helmut Witten (2011). *Email (nur?) für Dich*. Informatik im Kontext. URL: <https://medienwissenschaft.uni-bayreuth.de/inik/entwuerfe/email-nur-fuer-dich/> (besucht am 15. 07. 2022).
- Hartmann, Werner, Michael Näf und Raimond Reichert (2006). *Informatikunterricht planen und durchführen*. eXamen.press. Berlin Heidelberg: Springer-Verlag. ISBN: 978-3-540-34484-1. DOI: [10.1007/978-3-540-34485-8](https://doi.org/10.1007/978-3-540-34485-8).
- Harvey, Brian und Jens Mönig (2010). „Bringing “No Ceiling” to Scratch: Can One Language Serve Kids and Computer Scientists?“ In: S. 10.
- Hiertz, Guido R. u. a. (Jan. 2010). „The IEEE 802.11 Universe“. In: *IEEE Communications Magazine* 48.1, S. 62–70. ISSN: 1558-1896. DOI: [10.1109/MCOM.2010.5394032](https://doi.org/10.1109/MCOM.2010.5394032).
- Hodges, Steve u. a. (Apr. 2020). „Physical Computing: A Key Element of Modern Computer Science Education“. In: *Computer* 53.4, S. 20–30. ISSN: 1558-0814. DOI: [10.1109/MC.2019.2935058](https://doi.org/10.1109/MC.2019.2935058).
- Hromkovic, Juraj und Tobias Kohn (17. Aug. 2018). *Einfach Informatik 7-9. Programmieren*. Klett Schulbuchverlag. 127 S. ISBN: 978-3-264-84463-4.
- IANA (26. Feb. 2021). *Protocol Numbers*. Protocol Numbers. URL: <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml> (besucht am 30. 11. 2021).
- International Telecommunication Union (1994). *Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*. Patent ITU-T Recommendation X.200. International Telecommunication Union.
- Janet Ellen Abbate (1994). „From ARPANET to Internet: A history of ARPA-sponsored computer networks, 1966-1988“. Pennsylvania: University of Pennsylvania. URL: <https://www.proquest.com/openview/ecbd3abfe969c0b0929041c91d63e756/1?cbl=18750&diss=y&pq-origsite=gscholar&parentSessionId=18GK1vmEFkY4fjcpHEAJnexZd4E2ss6bfYNXMBY9B64%3D> (besucht am 24. 05. 2022).

- K-12 Computer Science Framework Steering Committee (2016). *K-12 Computer Science Framework*. Technical Report. New York, NY, USA: Association for Computing Machinery. 307 S.
- Kaczmarczyk, Lisa C. u. a. (10. März 2010). „Identifying Student Misconceptions of Programming“. In: *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. SIGCSE '10. New York, NY, USA: Association for Computing Machinery, S. 107–111. ISBN: 978-1-4503-0006-3. DOI: [10.1145/1734263.1734299](https://doi.org/10.1145/1734263.1734299).
- Katterfeldt, Eva-Sophie, Nadine Dittert und Heidi Schelhowe (1. Sep. 2015). „Designing Digital Fabrication Learning Environments for Bildung: Implications from Ten Years of Physical Computing Workshops“. In: *International Journal of Child-Computer Interaction*. Digital Fabrication in Education 5, S. 3–10. ISSN: 2212-8689. DOI: [10.1016/j.ijcci.2015.08.001](https://doi.org/10.1016/j.ijcci.2015.08.001).
- Keller, Lorenz (28. Apr. 2020). „E-Schooling: Microsoft-Apps haben die Schweizer Schulen erobert“. In: *Blick. Digital*. URL: <https://www.blick.ch/digital/apps/us-multi-microsoft-wird-von-schweizer-schulen-bevorzugt-dieses-teams-hat-ein-s-und-erfolg-id15866133.html> (besucht am 15. 07. 2022).
- Kohn, Tobias (2017). „Teaching Python Programming to Novices: Addressing Misconceptions and Creating a Development Environment“. Zürich: ETH Zürich. 166 S. URL: <https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/129666/eth-50720-02.pdf>.
- Kolb, David (1976). *The Learning Styles Inventory: Technical Manual*. McBer. 45 S.
- Krey, Olaf und Julia Schwanewedel (2018). „Lernen mit externen Repräsentationen“. In: *Theorien in der naturwissenschaftsdidaktischen Forschung*. Hrsg. von Dirk Krüger, Ilka Parchmann und Horst Schecker. Berlin, Heidelberg: Springer, S. 159–175. ISBN: 978-3-662-56320-5. DOI: [10.1007/978-3-662-56320-5_10](https://doi.org/10.1007/978-3-662-56320-5_10).
- Lehner, Martin (30. März 2020). *Didaktische Reduktion*. 2. überarbeitete Auflage. Haupt Verlag. 216 S. ISBN: 978-3-8252-5383-7.
- Liu, Jing, Xiaoyan Li und Jun Han (Jan. 2021). „A Comparative Study of Information Technology Curriculum Standards in Primary and Secondary Schools in China and the United States, Japan and Britain“. In: *2021 10th International Conference on Educational and Information Technology (ICEIT)*, S. 96–100. DOI: [10.1109/ICEIT51700.2021.9375549](https://doi.org/10.1109/ICEIT51700.2021.9375549).
- Mandl, Heinz (Jan. 2006). „Wissensaufbau Aktiv Gestalten“. In: *Schüler Wissen für Lehrer*, S. 28–30.
- McNally, Myles u. a. (3. März 2006). „Do Lego Mindstorms Robots Have a Future in CS Education?“ In: *ACM SIGCSE Bulletin* 38.1, S. 61–62. ISSN: 0097-8418. DOI: [10.1145/1124706.1121362](https://doi.org/10.1145/1124706.1121362).
- Meinel, Christoph und Harald Sack (18. Mai 2009). *Digitale Kommunikation: Vernetzen, Multimedia, Sicherheit*. Springer Science & Business Media. 425 S. ISBN: 978-3-540-92922-2.
- Mittermeir, Roland und Ernestine Bischof (1. Jan. 2022). *Informatik erLeben*. Einheit «Netze». URL: <http://informatik-erleben.uni-klu.ac.at/einheiten/n/> (besucht am 15. 07. 2022).
- Modrow, Eckart und Kerstin Strecker (20. Juni 2016). *Didaktik der Informatik*. De Gruyter Oldenbourg. ISBN: 978-3-486-72011-2. DOI: [10.1515/9783486720112](https://doi.org/10.1515/9783486720112).

- Perkins, Charles E. (Okt. 1996). *IP Encapsulation within IP*. Request for Comments RFC 2003. Internet Engineering Task Force. 14 S. DOI: [10.17487/RFC2003](https://doi.org/10.17487/RFC2003).
- Petäjällä, Juha u. a. (Juni 2017). „Evaluation of LoRa LPWAN Technology for Indoor Remote Health and Wellbeing Monitoring“. In: *International Journal of Wireless Information Networks* 24.2, S. 153–165. ISSN: 1068-9605, 1572-8129. DOI: [10.1007/s10776-017-0341-8](https://doi.org/10.1007/s10776-017-0341-8).
- Piaget, Jean (1947). *Psychologie Der Intelligenz*. Zürich: Rascher. 198 S.
- Postel, Jon (Aug. 1980). *User Datagram Protocol*. Request for Comments RFC 768. Internet Engineering Task Force. 3 S. DOI: [10.17487/RFC0768](https://doi.org/10.17487/RFC0768).
- (Sep. 1981). *Transmission Control Protocol*. Request for Comments RFC 793. Internet Engineering Task Force. 91 S. DOI: [10.17487/RFC0793](https://doi.org/10.17487/RFC0793).
- Przybylla, Mareen (17. Sep. 2018). „From embedded systems to physical computing : challenges of the «Digital World» in secondary computer science education“. Potsdam: Universität Potsdam. 296 S.
- Przybylla, Mareen und Ralf Romeike (20. Okt. 2015). „Key Competences with Physical Computing.“ In: *KEYCIT 2014: Key Competences in Informatics and ICT*. Potsdam: Universitätsverlag Potsdam.
- Raspberry Pi Foundation (25. Juni 2022). *Raspberry Pi Foundation – About Us*. Raspberry Pi - About us. URL: <https://www.raspberrypi.org/about/> (besucht am 25.06.2022).
- Regierungsrat Kanton Glarus (25. Juni 2019). *Lehrplan für das Gymnasium*. URL: https://www.kanti-glarus.ch/public/upload/assets/26358/Lehrplan_Gymnasium_Oberstufe_SN_FIN_19_02_2020.pdf (besucht am 13.06.2022).
- Resnick, Mitchel u. a. (Nov. 2009). „Scratch: Programming for All“. In: *Communications of the ACM* 52.11, S. 60–67. ISSN: 0001-0782, 1557-7317. DOI: [10.1145/1592761.1592779](https://doi.org/10.1145/1592761.1592779).
- Romeike, Ralf und Mareen Przybylla (2014). „Physical Computing and Its Scope – Towards a Constructionist Computer Science Curriculum with Physical Computing“. In: *Informatics in Education - An International Journal* 13.2, S. 241–254. ISSN: 1648-5831. URL: <https://www.cceol.com/search/article-detail?id=171588> (besucht am 31.05.2022).
- Rösler, Horst F. (1996). „Die Didaktische Reduktion - Eine Bestandsaufnahme“. In: *Naturwissenschaften im Unterricht. Chemie*.
- Sarı, Uğur u. a. (24. Jan. 2022). „Algorithmic Thinking Development through Physical Computing Activities with Arduino in STEM Education“. In: *Education and Information Technologies*. ISSN: 1573-7608. DOI: [10.1007/s10639-022-10893-0](https://doi.org/10.1007/s10639-022-10893-0).
- Schmalfeldt, Thomas (2019). „Einsatz von Skill Cards Und Story Cards Für Einen Kreativitätsfördernden Informatikunterricht Auf Der Sekundarstufe I“. In: *Informatik Für Alle*. Hrsg. von Arno Pasternak. Bonn: Gesellschaft für Informatik, S. 305–314. DOI: [10.18420/infos2019-c16](https://doi.org/10.18420/infos2019-c16).
- Schweizerische Konferenz der kantonalen Erziehungsdirektoren (27. Okt. 2017). *Rahmenlehrplan Für Die Maturitätsschulen: Informatik*. URL: https://edudoc.ch/record/131917/files/rlp_inf_2017_d.pdf (besucht am 03.09.2021).

- Scratch Wiki Community (4. Sep. 2018). *Deleting Part of a String - Scratch Wiki*. Scratch Wiki. URL: https://en.scratch-wiki.info/wiki/Deleting_Part_of_a_String (besucht am 17.05.2022).
- Senatsverwaltung Abteilung Bildung Berlin (2015). *Rahmenlehrplan Teil C Informatik, Jahrgangsstufen 7-10*. URL: https://bildungsserver.berlin-brandenburg.de/fileadmin/bbb/unterricht/rahmenlehrplaene/Rahmenlehrplanprojekt/amtliche_Fassung/Teil_C_Informatik_2015_11_10_WEB.pdf (besucht am 21.06.2022).
- Siebert, Horst (10. Juni 2005). *Pädagogischer Konstruktivismus*. 3. überarbeitete und erweiterte Auflage. Beltz. 150 S. ISBN: 978-3-407-29148-6.
- (2014). „Lehren Und Lernen Aus Konstruktivistischer Sicht“. In: *Hochschuldidaktische Weiterbildung an Fachhochschulen: Durchführung - Ergebnisse - Perspektiven*. Hrsg. von Rudolf Egger, Doris Kiendl-Wendner und Martin Pöllinger. Wiesbaden: Springer Fachmedien Wiesbaden, S. 49–68. ISBN: 978-3-658-01497-1. URL: https://doi.org/10.1007/978-3-658-01497-1_4.
- Sirkkiä, Teemu und Juha Sorva (15. Nov. 2012). „Exploring Programming Misconceptions: An Analysis of Student Mistakes in Visual Program Simulation Exercises“. In: *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*. Koli Calling '12. New York, NY, USA: Association for Computing Machinery, S. 19–28. ISBN: 978-1-4503-1795-5. DOI: [10.1145/2401796.2401799](https://doi.org/10.1145/2401796.2401799).
- Steil, Tamina (3. Feb. 2022). *WhatsApp-Nutzerzahlen im Überblick 2022*. Userlike. URL: <https://www.userlike.com/de/blog/whatsapp-nutzerzahlen#whatsapp-nutzer-altersgruppen> (besucht am 15.07.2022).
- Steinbach, Nico und Eckart Zitzler (2017). *Ein gamebasierter Ansatz zum Programmierunterricht in der Lehrerinnen- und Lehrerbildung*. Gesellschaft für Informatik, Bonn. ISBN: 978-3-88579-668-8. URL: <http://dl.gi.de/handle/20.500.12116/4309>.
- Streller, Sabine u. a. (2019). *Chemiedidaktik an Fallbeispielen: Anregungen für die Unterrichtspraxis*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-58645-7. DOI: [10.1007/978-3-662-58645-7](https://doi.org/10.1007/978-3-662-58645-7).
- Süss, Daniel, Gregor Waller, Jael Bernath u. a. (2020). *JAMES - Jugend, Aktivitäten, Medien - Erhebung Schweiz. Ergebnisbericht zur JAMES-Studie 2020*. Zürich: Zürcher Hochschule für Angewandte Wissenschaften.
- Süss, Daniel, Gregor Waller, Anna-Lena Huber u. a. (2014). *JAMES - Jugend, Aktivitäten, Medien - Erhebung Schweiz. Ergebnisbericht zur JAMES-Studie 2014*. Zürich: Zürcher Hochschule für Angewandte Wissenschaften.
- Süss, Daniel, Gregor Waller, Céline Külling u. a. (2018). *JAMES - Jugend, Aktivitäten, Medien - Erhebung Schweiz. Ergebnisbericht zur JAMES-Studie 2018*. Zürich: Zürcher Hochschule für Angewandte Wissenschaften.
- Süss, Daniel, Gregor Waller, Lilian Suter u. a. (2016). *JAMES - Jugend, Aktivitäten, Medien - Erhebung Schweiz. Ergebnisbericht zur JAMES-Studie 2016*. Zürich: Zürcher Hochschule für Angewandte Wissenschaften.
- Swidan, Alaaeddin, Felienne Hermans und Marileen Smit (8. Aug. 2018). „Programming Misconceptions for School Students“. In: *Proceedings of the 2018 ACM Conference on International Computing Education Research*. ICER '18. New York, NY, USA: Association for Computing Machinery, S. 151–159. ISBN: 978-1-4503-5628-2. DOI: [10.1145/3230977.3230995](https://doi.org/10.1145/3230977.3230995).

- Swisscom (Schweiz) AG (5. Juli 2021). *Factsheet LPN - Enterprise IoT - Low Power Network (LoRaWAN)*. URL: https://documents.swisscom.com/product/filestore/lib/84eb9c67-84fa-4e05-b5b0-c54c4baeeca7/fs_lpn-de.pdf (besucht am 31.05.2022).
- Tan, Z A u. a. (2. Dez. 2019). „Analysis on LoRa RSSI in Urban, Suburban, and Rural Area for Handover Signal Strength-Based Algorithm.“ In: *IOP Conference Series: Materials Science and Engineering* 705, S. 012012. ISSN: 1757-899X. DOI: [10.1088/1757-899X/705/1/012012](https://doi.org/10.1088/1757-899X/705/1/012012).
- The LEGO Group (1. Jan. 2022). *Über | MINDSTORMS | Offizieller LEGO® Shop CH*. URL: <https://www.lego.com/de-ch/themes/mindstorms/ev3> (besucht am 13.06.2022).
- Wang, Wenbo u. a. (2019). „Comparative Analysis of Channel Models for Industrial IoT Wireless Communication“. In: *IEEE Access* 7, S. 91627–91640. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2019.2927217](https://doi.org/10.1109/ACCESS.2019.2927217).
- Watzlawick, Paul, Janet Beavin Bavelas und Don D. Jackson (2011). *Menschliche Kommunikation: Formen, Störungen, Paradoxien*. Huber. 322 S. ISBN: 978-3-456-84970-6.
- Zendler, Andreas und Christian Spannagel (Mai 2008). „Empirical Foundation of Central Concepts for Computer Science Education“. In: *Journal on Educational Resources in Computing* 8.2, S. 1–15. ISSN: 1531-4278, 1531-4278. DOI: [10.1145/1362787.1362790](https://doi.org/10.1145/1362787.1362790).
- Zuckarelli, Joachim L. (2021). *Programmieren Lernen Mit Python Und JavaScript*. Wiesbaden: Springer. ISBN: 978-3-658-29850-0.