

Universität Würzburg
Lehrstuhl für Physik und ihre Didaktik
Fakultät für Physik und Astronomie

Schriftliche Hausarbeit
im Rahmen der ersten Staatsprüfung
für ein Lehramt an Gymnasien

Einführung in den Mikrocontrollereinsatz im Schul- und
speziell dem Physikunterricht

Eingereicht von
Dickmeis, Lars

im
März/2023

Betreuer: Dr. M. Elsholz

Gutachter: Dr. S. Lück

Inhaltsverzeichnis

0	Einleitung: Experimentieren in Zeiten von Corona	5
1	Die Hardware	6
1.1	<i>Das Mikrocontroller-Board als Herzstück eines Projektes</i>	<i>6</i>
1.1.1	Arduino oder Raspberry Pi?	7
1.1.2	Die Arduino-Serie: Das Original	8
1.1.3	Die ESP-Serie: Eine günstige Alternative	12
1.1.4	Fazit: Welches System eignet sich für den Einsatz im Physikunterricht?	14
1.2	<i>Sensoren: Die Sinnesorgane eines Projektes</i>	<i>15</i>
1.2.1	Grundlagen der elektrischen Messtechnik	15
1.2.2	Die Verwendung von Sensoren mit analogem Ausgangssignal	17
1.2.3	Die Verwendung von Sensoren mit digitalem Ausgangssignal	20
1.2.4	Shields: Erweiterungsmöglichkeiten eines Boards	21
1.2.5	Das Grove-System	23
1.2.6	Beispiele	24
1.3	<i>Weitere Bestandteile eines Projektes</i>	<i>25</i>
2	Die Software	26
2.1	<i>Die Arduino IDE: Wie programmiere ich einen Arduino?</i>	<i>26</i>
2.2	<i>Der Arduino-Programm-Code</i>	<i>28</i>
2.3	<i>Arduino-Bibliotheken</i>	<i>29</i>
2.4	<i>Arduino ganz ohne Programmieren - Geht das?</i>	<i>29</i>
2.5	<i>Arduino und das Internet of Things</i>	<i>31</i>
2.5.1	Was ist das Internet of Things?	31
2.5.2	Umsetzung von IoT-Projekten mit dem Arduino	32
3	Didaktische Perspektive: Was bringt der Einsatz im Schulkontext?	34
3.1	<i>Ein Beitrag zur Bildung in der digitalen Welt</i>	<i>34</i>
3.2	<i>Der Praxiseinsatz im Physikunterricht</i>	<i>36</i>
3.3	<i>Authentizität des Experimentierens im Unterricht</i>	<i>39</i>
3.4	<i>Das Schulbuch der Zukunft</i>	<i>40</i>
3.4.1	Mikrocontroller im Schulbuch der Zukunft	41
3.4.2	Die Einbindung von Mikrocontrollern in tet.folio	42

3.5	<i>Kritische Hinterfragung des Schuleinsatzes</i>	43
4	Wie lassen sich Mikrocontroller in der Schule konkret einsetzen?	46
4.1	<i>Klassische Physik</i>	46
4.2	<i>Moderne Physik</i>	49
4.3	<i>Fachübergreifende Projektideen</i>	51
5	Projekt: Das gläserne Ökosystem	52
5.1	<i>Die Lernziele</i>	53
5.2	<i>Ablauf des Schülerlabors</i>	54
5.2.1	Erste Phase: Vorüberlegungen mit biologischem Fokus	54
5.2.2	Zweite Phase: Physical Computing	57
5.2.3	Dritte Phase: Datenauswertungen und -präsentation	61
5.2.4	Vierte Phase: Naturwissenschaftliches Experimentieren	65
5.2.5	Fünfte Phase: Verallgemeinerung auf das globale Erdsystem	65
5.3	<i>Einbindung in tet.folio</i>	66
6	Fazit	68
7	Danksagung	69
8	Literaturverzeichnis	70
A	Anhang	73
B	Erklärung nach §29 LPO-I	77

0 Einleitung: Experimentieren in Zeiten von Corona

„Das Experiment spielt im naturwissenschaftlichen Unterricht eine zentrale Rolle“ (Duit et al., 2010, S.1). Das Experimentieren sollte daher in der Planung, Ausarbeitung und Durchführung von Unterricht in den Naturwissenschaften besonders bedacht werden. Diese These stützt eine Videostudie von Tesch und Duit, nach der mit ca. 70% ein beachtlicher Teil der Unterrichtszeit auf das Experimentieren verwendet wird (Duit et al., 2010). Während der Corona-Pandemie zeigte sich allerdings eine massive Verschiebung des Schwerpunkts weg vom Experimentieren. Eine Studie der Uni Konstanz zeigt mittels einer Online-Lehrerbefragung, dass bei etwa der Hälfte aller Lehrkräfte das Experimentieren eine untergeordnete Rolle im Distanzunterricht spielte. Die Studie zeigt auch, dass alternative Ansätze wie Simulations- beziehungsweise Modellbildungsprogramme oder das Experimentieren mithilfe von Smartphones nur bedingt genutzt wurden. Stattdessen zeigt sich ein Trend zu vermehrten Argumentations-, Text- und Rechenaufgaben (Henne et al., 2021). Aufgrund dieser Entwicklung stellt sich die Frage nach weiteren Möglichkeiten, mit denen Schüler und Schülerinnen (von hier an bezeichnet mit SchülerInnen) auch zu Hause experimentieren können. Ein möglicher Ansatz hierfür besteht in der Verwendung von Mikrocontrollern, beispielweise einem Arduino. Die Diskussion der Möglichkeiten und Grenzen dieses Ansatzes sowie die Suche nach kostengünstigen, individualisierbaren und anfängerfreundlichen Systemen, mit denen Lernende auch zu Hause wissenschaftliche Fragen experimentell bearbeiten können, stellt die Motivation für die vorliegende Arbeit dar.

Im Rahmen dieser Arbeit soll eine Übersicht über die Mikrocontroller-Anwendung im Schulkontext und dabei speziell dem Physikunterricht gegeben werden. In den ersten beiden Kapiteln wird der Blick hierzu zuerst auf die Hardware und anschließend auf die Software geworfen. Die beiden Kapitel geben einen Überblick über die verschiedenen Facetten eines Projektes und sind daher als Einstiegskapitel für „Mikrocontroller-Anfänger“ gedacht. Darüber hinaus soll eine Einschätzung über die Anwendbarkeit verschiedener Systeme im Schulkontext erfolgen. Anschließend werden in Kapitel drei didaktische Gesichtspunkte, die für den Einsatz von Mikrocontrollern im Unterricht sprechen, diskutiert. Im vierten Kapitel werden konkrete Anwendungsbeispiele vorgestellt. Zum Schluss der Arbeit wird als Praxisbeispiel ein Schülerlabor vorgestellt, das im Rahmen des Labs4Future an der Universität Würzburg entstanden ist. In diesem Projekt werden verschiedene Umweltparameter mithilfe von Arduino-Sensoren in einem geschlossenen Ökosystem aufgezeichnet.

1 Die Hardware

Für die Planung eines Mikrocontroller-Projektes müssen grundlegend zwei verschiedene Seiten bedacht werden: Auf der einen Seite die Hardware, das heißt der Mikrocontroller selbst, die verschiedenen Sensoren und weitere elektronische Bauteile. Auf der anderen Seite steht die Software, also der Programmcode, mit dem der Mikrocontroller angesteuert wird. Im ersten Kapitel dieser Arbeit soll ein Überblick über die verschiedenen Komponenten auf Hardware-Seite gegeben werden.

1.1 Das Mikrocontroller-Board als Herzstück eines Projektes

Ein Mikrocontroller ist ein „*Prozessorsystem, das eigenständig bestimmte Aufgaben, besonders Mess- und Steuerungsfunktionen, ausführen kann*“ (Duden.de, 2022). Er stellt daher die Grundlage jedes Projektes dar. Da am Mikrocontroller alle Daten zusammenlaufen und verarbeitet werden, kann man sich ihn als eine Art Gehirn eines Projektes vorstellen (Brühlmann, 2019). Für die Anwendung in der Praxis sind die Prozessoren zusammen mit verschiedenen Widerständen, Kondensatoren und Anschlüssen auf Leiterplatten gelötet. Die fertige Leiterplatte nennt man Mikrocontroller-Board, wobei umgangssprachlich oft nicht zwischen dem Board und dem eigentlichen Mikrocontroller unterschieden wird.

Ein Beispiel für ein solches Mikrocontroller-System ist die Arduino-Plattform. Da diese Plattform auf dem so genannten „Open Source“ Prinzip beruht, kann jeder die Soft- und Hardware-Komponenten weiterentwickeln, nachbauen und unter neuem Namen verkaufen. Hierzu stehen die Software und detaillierte Baupläne aller Hardwarekomponenten auf der Arduino-Website unter docs.arduino.cc frei zur Verfügung. Man findet daher heute eine nahezu unendliche Auswahl an Mikrocontroller-Boards. Um einen kurzen Überblick über zumindest einen kleinen Teil dieser Boards zu geben, werden im Folgenden drei große Hersteller-Serien vorgestellt.

1.1.1 Arduino oder Raspberry Pi?

Die beiden Systeme Arduino und Raspberry Pi werden oft als ähnliche, konkurrierende Plattformen wahrgenommen. Tatsächlich unterscheiden sie sich in Technik und Anwendung aber grundsätzlich (Strobel, 2019). Im folgenden Abschnitt sollen die beiden Systeme miteinander verglichen werden. Anschließend wird eine Einschätzung gegeben, welches System sich für den Schuleinsatz eignet.

Das Arduino-System besteht hardwareseitig aus verschiedenen Mikrocontroller-Boards und dazugehörigen Sensoren, Displays und vielem mehr. Softwareseitig gibt es eine Programmierumgebung, in der ein so genannter „Sketch“ geschrieben werden kann, der anschließend auf den Mikrocontroller aufgespielt wird und dort in einer Endlosschleife abläuft. Zur Programmierung eines Arduinos benötigt man also zwingend einen Computer oder sonstiges digitales Endgerät. Die Raspberry Pi Modelle fungieren hingegen wie kleine, eigenständige Computer, da sie, im Vergleich zu den Arduinos, über einen deutlich besseren Prozessor, einen eigenen Arbeitsspeicher und die Anschlussmöglichkeiten eines Bildschirms, einer Maus und einer Tastatur verfügen. Nach dem Aufspielen des Betriebssystems kann ein Raspberry Pi deshalb alle Funktionen eines „normalen Desktop-PCs“ erfüllen, weshalb man bei ihnen nicht von Mikrocontrollern, sondern von Mini oder Ein-Platinen-Computern spricht (Girwidz & Watzka, 2018). Ein Raspberry Pi kann daher deutlich komplexere und rechenintensivere Aufgaben übernehmen als ein Arduino, der im Prinzip nur dieselben Arbeitsschritte immer und immer wieder abarbeitet. Die Funktionsweisen der beiden Systeme sind also grundlegend verschieden, auch wenn sie in Projekten die gleichen Aufgaben übernehmen können.

Neben der deutlichen technischen Überlegenheit bergen die Raspberry Pis allerdings auch Nachteile: Die fortgeschrittenere Technik und die Notwendigkeit eines Bildschirms, einer Maus und einer Tastatur, machen die Nutzung eines Raspberry Pis teurer als einen Arduino. Des Weiteren gestaltet sich die Inbetriebnahme für Anfänger durch das Aufspielen eines Betriebssystems schwieriger (Strobel, 2019).

Zusammenfassend lässt sich festhalten, dass sich abhängig von der Anwendung mal ein Mikrocontroller, mal ein Ein-Platinen-Computer eignet. Das Arduino-System eignet sich eher für einfache, sich wiederholende Datenerfassungs- oder Steuerungsprozesse. Werden die Aufgaben komplexer und die Platine soll rechenintensive Prozesse übernehmen oder mehrere

Prozesse parallel behandeln, sollte das Raspberry-System verwendet werden. Auch für die Verwendung im Schulkontext kann man kein generell überlegenes System ausmachen. Für den Physikunterricht scheint die Verwendung eines Mikrocontrollers vorteilhaft, da SchülerInnen für deren Verwendung kaum Vorkenntnisse benötigen, sodass der Fokus schnell auf die eigentliche Messtechnik gelegt werden kann. Für den Informatik-Unterricht scheint die Verwendung des Raspberry Pis geeigneter, da hiermit ein tieferer Einstieg in die Programmierung möglich ist.

1.1.2 Die Arduino-Serie: Das Original

Das erste Arduino-Board wurde 2005 (Söderby, 2023) auf Studenten ausgelegt, die keine Programmier- und Elektronik-Kenntnisse mitbringen (Margolis et al., 2020). Diesen Fokus hat die Plattform bis heute behalten, sodass ihr Motto immer noch mit „*Basteln, Spaß und Lernen*“ beschrieben werden kann (Brühlmann, 2019). Um den Einstieg in das Arbeiten mit Mikrocontrollern zu vereinfachen, bündelt die Arduino-Plattform Soft- und Hardware. Dieses Kapitel beschränkt sich allerdings auf die Hardware-Seite, also die so genannten „Arduino-Boards“.

Auf der offiziellen Arduino-Website finden sich heute etwa 25 verschiedene Arduino-Boards. Ein vollständiger Vergleich würde den Rahmen dieses Kapitels deutlich sprengen, weshalb sich im Folgenden auf einen Überblick über die Arduino-Serie und den konkreten Vergleich von drei Boards beschränkt wird.

Der größte Vorteil der original Arduino-Serie ist, dass deren Boards für Anfänger und damit auch für den Schulgebrauch sehr gut geeignet sind: Sie sind online gut dokumentiert, das heißt man findet unzählige Anleitung für diverse Projekte auch auf Deutsch. Die Arduino-Website (arduino.cc) liefert zudem viele Tutorials, Projekte zum Nachbauen und ein Forum, in dem sich Lösungen zu nahezu allen denkbaren Fragen finden. Die Beiträge in diesem Forum sind allerdings meistens auf Englisch verfasst, was potenziell zu Problemen im Schulkontext führen kann. Ein weiterer großer Vorteil der Arduino-Serie für die Verwendung in der Schule ist „arduino education“. Dieses speziell auf die Lehre ausgelegt STEAM-Programm ($\triangleq$ Science, Technology, Engineering, Arts, Mathematics) soll den Arduino-Einsatz in Schulen vereinfachen (*Arduino Education*, o. D.). Ausgelegt auf die Altersgruppen „middle school“ (11-14 Jahre), „high school“ (14-17 Jahre) und „university“ (17+ Jahre) findet man deshalb im Arduino-Store aufeinander aufbauende „Hardware-Kits“, die als fertige Baukästen aus einem

Board und verschiedenen Sensoren verkauft werden. Diese ermöglichen direkt eine Vielzahl von Projekten, ohne weitere Komponenten bestellen zu müssen. Einen Überblick über die momentan in Deutschland verfügbaren Sets gibt Abbildung 1. Die einzelnen Sets im Detail zu behandeln, würde den Rahmen dieses Kapitels sprengen. Der interessierte Leser findet Infos zu den Education-Sets beispielsweise unter: <https://www.christiani.de/schule/arduino-education/>.



Abbildung 1: Überblick über die momentan in Deutschland verfügbaren Arduino Education Sets.

Bildquelle:
<https://www.christiani.de/cms/arduino-education/>
 (Abgerufen am 5.3.2023)

Mit den Hardware-Kits erwirbt man außerdem den Zugriff auf Online-Lerninhalte wie verschiedene so genannte „Homeschooling Guides“ oder Video-Tutorials, die auch auf Deutsch verfügbar sind (*Arduino Education*, o. D.). Ein weiterer Vorteil besteht in der Bereitstellung der App „Arduino Science Journal“, mit der einfache Experimente mit dem Handy überwacht werden können, sodass die Lernenden keinen zusätzlichen Computer benötigen. Die Daten der angeschlossenen Arduino-Sensoren können zudem mit Daten der im Handy integrierten Sensoren kombiniert werden (*Arduino Education*, o. D.).

Neben diesen klaren Vorteilen gibt es allerdings auch Nachteile der Arduino-Serie. Viele Arduino-Boards haben keine integrierte Möglichkeit einer Internetanbindung, was je nach Anwendungszweck ein Problem darstellen kann. Um dieses Problem zu lösen, kann man so genannte „Shields“ für die Boards kaufen. Diese aufsetzbaren Erweiterungs-Module gibt es für viele verschiedene Anwendungszwecke: Beispielsweise für das Einbinden eines Ethernet-Anschlusses, eines WiFi-Moduls, einer Einheit zur Verwendung mobilen Internets oder der Einbindung eines SD-Karten-Moduls. Die Verwendung passender Shields ermöglicht eine einfache Erweiterung der Arduino-Boards ohne Verkabelungs- oder Programmieraufwand (siehe auch Kapitel 1.2.4). Neben dem Nachteil der teils fehlenden Internetanbindung gibt es weitere Einschränkungen bei der Verwendung von Arduinos: Die Arduino-Boards sind mit einem vergleichsweise kleinen Speicher ausgerüstet und besitzen eine geringe Prozessorgeschwindigkeit. Die damit einhergehenden Einschränkungen sind für die meisten

Anwendungen im Schulkontext allerdings irrelevant, da auch die bei den Standard-Arduinos verbauten Speicher und Prozessoren für die meisten Schülerprojekte ausreichend sind. Der letzte zu nennende Nachteil der Arduino-Serie ist ihr Preis. Die originalen Boards sind im Vergleich zu ihren Klonen oder Alternativen in etwa doppelt so teuer¹. Eine günstigere Alternative wird im nächsten Kapitel vorgestellt.

Die aktuell verfügbaren Arduino-Boards sind in die drei Familien Nano, MKR und Classic eingeteilt (*Arduino Hardware*, o. D.). Die Nano-Familie beinhaltet hierbei, wie der Name schon vermuten lässt, die kleinsten Boards mit einer Größe von etwa 45x18mm (*Arduino-Store*, o. D.). Die Boards der Familie bieten verschiedene „on Board“ verbaute Sensoren und WiFi- oder Bluetooth-Module. Die MKR-Familie ist auf Anwendungen für das „Internet der Dinge“ ausgelegt und deshalb standardmäßig mit einem Modul ausgestattet, dass eine Möglichkeit der Datenübertragung bietet. Das Thema „Internet der Dinge“ wird in Kapitel 2.5 genauer behandelt. Des Weiteren besitzt die MKR-Familie einen „Crypto Chip“ zur sicheren Datenübertragung (*Arduino Hardware*, o. D.). Boards dieser Familie werden im folgenden Vergleich nicht weiter behandelt, da sie für den Einstieg im Schulkontext zu komplex sind. Die Classic-Familie beinhaltet die „Standard-Boards“ wie den Arduino UNO, Arduino Due oder Arduino Micro. Der Arduino UNO ist der Klassiker unter den Arduino-Boards und das meistgenutzte und am besten dokumentierte Board der Serie (*Arduino-Store*, o. D.). Der Arduino Micro ermöglicht neben dem Arduino Leonardo als einziges Board eine spezielle Art der USB-Kommunikation, die die Boards auch als Maus oder Tastatur an einem Computer funktionieren lässt. Dieses Feature wird für den Schulgebrauch allerdings nicht zwingend benötigt.

In der auf der nächsten Seite folgenden Tabelle werden die drei Boards Arduino UNO Rev3, Arduino Nano Every und Arduino Due miteinander verglichen. Für die Preisangaben wurden hierzu die Preise des Arduino-Stores übernommen. Es soll allerdings darauf hingewiesen werden, dass die Boards auch über andere Shops und Plattformen zu günstigeren Preisen erhältlich sind.

¹ Vergleich des Arduino UNO Rev3 (21,90€ auf [berrybase.de](https://www.berrybase.de), Stand: 6.2.2023) mit dem AZ-Delivery-Board ATmega328 (9,99€ auf [az-delivery.de](https://www.az-delivery.de), Stand: 6.2.2023)

	Arduino UNO Rev3	Arduino Due	Arduino Nano Every
Größe	68,6 x 53,4mm	102,52 x 53,3mm	45 x 18mm
Betriebsspannung	5V	3,3V	5V
Digitale Inputs	14	54	12
Analoge Inputs	6	12	8
Flash Memory	32KB	512KB	48KB
Clock-Speed	16Mhz	84Mhz	20Mhz
RAM	3KB	96KB	6KB
WiFi/ Bluetooth	Nein/Nein	Nein/Nein	Nein/Nein
Preis*	24,00€	42,00€	10,40€
Anwendung	Einstiegsprojekte	Große und rechenintensive Projekte	Projekte, die ein kleines Board benötigen

Tabelle 1: Gegenüberstellung der drei Arduino-Modelle Uno Rev3, Due und Nano Every (Quelle: arduino.cc, Stand: 26.1.2023) (Preise laut Arduino-Store unter store.arduino.cc, Stand 26.1.2023)*

Der Arduino UNO und Arduino Nano bieten neben der normalen auch eine WiFi-fähige Version des Boards ab 20,80€ (Nano) bzw. 46,70€ (Uno) (*Arduino-Store*, o. D.). Alternativ können die Boards mit WiFi-Shields nachgerüstet werden. Diese Lösung bietet der offizielle Arduino-Store allerdings nicht an, weshalb auf Drittanbieter zurückgegriffen werden muss (mehr dazu in Kapitel 1.2.4). Sollte man eine Internetverbindung benötigen, ist es deshalb einfacher, ein Board zu verwenden, das bereits eine Internetschnittstelle mitbringt. In Tabelle 1 fällt auf, dass der Arduino Due mit einer niedrigeren Betriebsspannung von 3,3V arbeitet. Um Schäden oder fehlerhafte Messungen zu vermeiden, ist bei der Anschaffung von elektronischen Komponenten deshalb darauf zu achten, dass alle Komponenten auf die korrekte Betriebsspannung des Mikrocontrollers ausgelegt sind. Die in Tabelle 1 aufgelisteten technischen Eigenschaften dienen als Vergleichsgrundlage in den folgenden Abschnitten, weshalb an dieser Stelle auf eine weitere Diskussion verzichtet wird.

Die folgende Tabelle soll die Vor- und Nachteile der Arduino-Serie zusammenfassen. Innerhalb der Serie lässt sich festhalten, dass sich der Arduino UNO am besten für Einsteiger und daher auch für den Schuleinsatz eignet (*Arduino-Store*, o. D.).

+	—
• Einsteigerfreundlich • Tutorials und Anleitungen auch auf Deutsch • Arduino education • Arduino-App: Science Journal • Fertige Hardwarekits für Schulen	• Teils fehlende Möglichkeit der Internetverbindung • Höherer Preis • Technisch unterlegen (für den Schulkontext aber ausreichend)

Tabelle 2: Zusammenfassung der Vor- und Nachteile der Arduino-Serie

1.1.3 Die ESP-Serie: Eine günstige Alternative

Wie bereits erwähnt gibt es mittlerweile unzählige Klone und Nachbauten der Arduino Boards, aber auch neu entwickelte Alternativen, die mit der Arduino-Plattform kompatibel sind. In diesem Kapitel soll die ESP-Serie der Firma Espressif mit ihren Vertretern ESP8266 und ESP32 als eine populäre Arduino-Alternative vorgestellt werden. Die beiden Begriffe ESP8266 und ESP32 bezeichnen hierbei die verbauten Mikrocontroller. Die erhältlichen Boards haben anderen Namen, beispielweise NodeMCU oder D1 mini.

Der erste konkrete Vorteil der ESP-Serie gegenüber der „originalen“ Arduino-Serie ist ihr geringer Preis. Boards mit einem ESP8266 sind bereits ab 5€, Boards mit einem ESP32 ab etwa 10€ erhältlich². Der zweite, im Hinblick auf den Preis beachtliche Vorteil, liegt in der technischen Überlegenheit, da die ESP-Boards bereits in der Grundausstattung eine WiFi-Schnittstelle und Boards mit einem ESP32 sogar zusätzlich eine Bluetooth-Schnittstelle besitzen. Neben der drahtlosen Schnittstelle sind auch Speicher und Rechenleistung deutlich verbessert. Die ESP32-Version des D1 mini liefert für unter 12€³ schon einen Flash-Speicher von 4MB, einen RAM von 512kB und eine Clock-Speed von bis zu 240MHz (*Datenblatt D1 mini ESP32*, o. D.). Vergleicht man diese Werte mit denen der Arduino-Boards aus Abschnitt 1.1.1, kann man einen deutlichen Unterschied feststellen. Da die beiden ESP-Mikrocontroller mit der Arduino IDE, also der Software zur Programmierung der Arduinos, kompatibel sind, kann man sie analog zu den oben genannten Arduino-Boards programmieren. Die Einbindung der Boards in die Arduino IDE ist allerdings etwas umständlich, da die nötige Schnittstelle nicht automatisch installiert wird.

² Onlineshop AZ-Delivery (az-delivery.de), Stand: 26.1.2023

³ Onlineshop AZ-Delivery (az-delivery.de), Stand: 26.1.2023

Ein weiterer Vorteil liegt in ihrer Größe, da die ESP basierten Boards deutlich kleiner sind als die meisten Arduino-Boards. Der Größenunterschied soll in Abbildung 2 veranschaulicht werden.

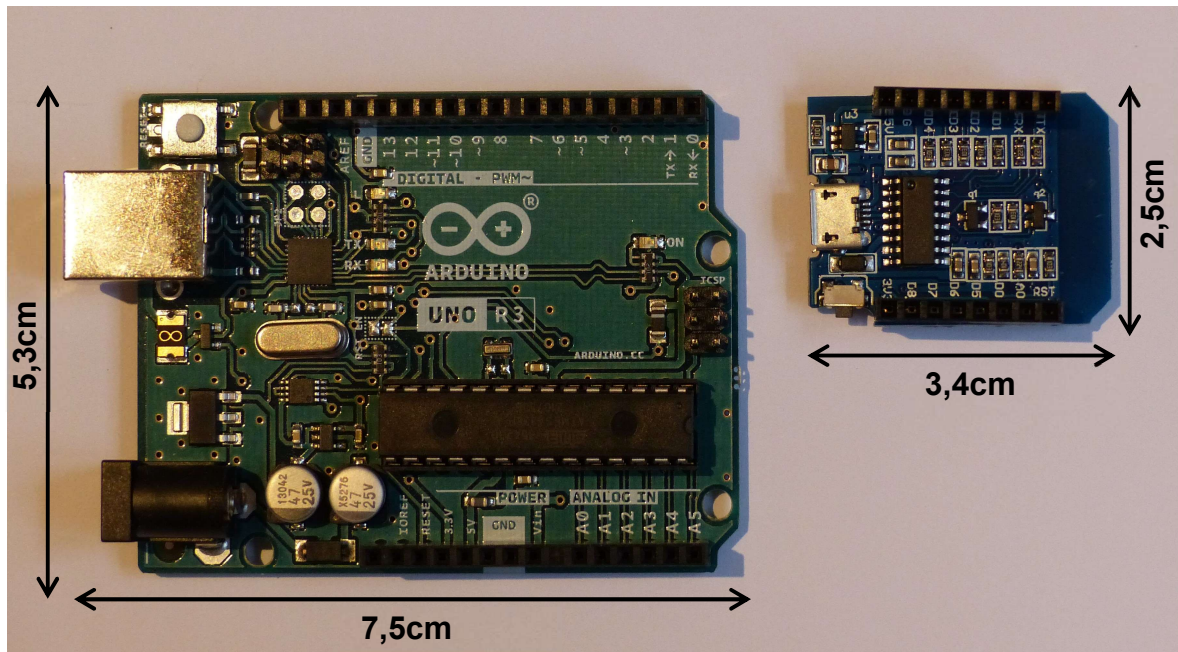


Abbildung 2: Größenvergleich des Arduino UNO Rev3 (links) und des ESP8266 basierten D1mini (rechts). Die Maße sind abgemessen und sollten daher nur als Größenorientierung dienen.
Bildquelle: Eigene Aufnahme

Der Nachteil der ESP-Serie liegt darin, dass man die gut dokumentierte Arduino-Plattform verlässt. Es gibt unzählige verschiedene ESP-Board Varianten, über die je nach Hersteller nur dürftige Infos erhältlich sind. Des Weiteren sind die meisten Einsteiger-Tutorials und Anleitungen auf Arduino-Boards ausgelegt. Alle Projekte funktionieren zwar auch auf einem ESP-Board, benötigen dann allerdings Anpassungen im Programmcode und dem Schaltplan. Neben dieser generellen Einschränkung haben auf dem ESP8266 basierende Boards eine weitere Einschränkung: Sie bieten lediglich einen einzigen analogen Input-Pin (*Datenblatt D1mini ESP8266*, o. D.), was für viele Projekte nicht ausreichend ist. Dieses Problem kann zwar über die Einbindung eines externen Analog-Digital-Wandlers behoben werden (mehr hierzu in Kapitel 1.2.1), es stellt jedoch für Einsteiger im Schulkontext eine weitere Hürde dar.

Das Fazit zur ESP-Serie fällt daher gemischt aus. Prinzipiell bietet die Serie dem erfahrenen Nutzer viele Vorteile. Im Schulkontext ist allerdings davon auszugehen, dass die SchülerInnen keine Erfahrung mit Mikrocontrollern und deren Verwendung haben, weshalb die Arduino-Serie im Schulkontext geeigneter scheint. Sollte ein Projekt aber auf möglichst geringe Kosten ausgelegt sein oder es zwingend erforderlich sein, dass die verwendeten Boards eine WiFi- oder

Bluetooth-Schnittstelle haben, kann die ESP-Serie eine passende Lösungsmöglichkeit darstellen. Wenn man sich für ein ESP-basiertes Board entschieden hat, sollte bei deren Beschaffung für die Schule darauf geachtet werden, dass die so genannten „Pin-Header“ ($\triangleq$ Anschlusspins) schon an das Board gelötet sind. Oftmals werden die Boards ohne bereits angelötete Kontakte versendet. Die nachfolgende Tabelle fasst die Vor- und Nachteile der ESP-Serie auf einen Blick zusammen.

+	—
• Gute Leistung zu günstigem Preis • WiFi-Schnittstelle • Bluetooth-Schnittstelle (nur ESP32)	• Wesentlich schlechter dokumentiert als die Arduino-Boards • Unübersichtliche Vielzahl an ESP-basierten Mikrocontroller-Boards • Nur ein analoger Input-Pin (ESP8266) • Etwas umständliche Initialisierung der Boards in der Arduino IDE

Tabelle 3: Fazit der Vor- und Nachteile der ESP basierten Boards

1.1.4 Fazit: Welches System eignet sich für den Einsatz im Physikunterricht?

Diese Frage lässt sich nicht eindeutig beantworten, da jede der oben genannten Serien eigene Vor- und Nachteile mitbringt. Welches System im Schulkontext genutzt wird, muss daher davon abhängig gemacht werden welche Vorkenntnisse die Nutzer mitbringen, in welchem Fach das System eingesetzt wird und was das eigentliche Ziel der Nutzung ist. Für den Einsatz im Physikunterricht empfiehlt sich die Arduino-Serie, da sie ein breites Nutzungsspektrum für den Unterricht bietet, ohne dass hierfür eine lange Einführung oder tiefgreifende Vorkenntnisse nötig sind. Viele Arduino-Hardwareboxen, Online-Inhalte und Anleitungen sind zudem didaktisch zumindest vorstrukturiert, was die Nutzung im Unterricht einfacher gestaltet. Um die Kosten zu senken, kann allerdings auch auf identische Klone der Arduino Boards von anderen Herstellern zurückgegriffen werden. Diese haben oft einen günstigeren Preis, aber denselben Aufbau und dieselbe Funktionsweise wie die Arduino Boards.

1.2 Sensoren: Die Sinnesorgane eines Projektes

Neben den Mikrocontrollern als Herzstück braucht es weitere Bestandteile, um ein Projekt an die jeweiligen Anforderungen und für die jeweilige Nutzung anzupassen. Hierzu gibt es die verschiedensten Optionen an Sensoren, Displays oder Aktoren. Im folgenden Kapitel soll auf die Sensoren als Sinnesorgane des Mikrocontrollers eingegangen werden. Diese sind insbesondere für die Verwendung im naturwissenschaftlichen Unterricht wichtig und bieten Lernenden die Möglichkeit, verschiedene Messwerterfassungssysteme selbstständig zu realisieren. Das vorliegende Kapitel soll einen Überblick über die Funktionsweise und Anwendung der Arduino-Sensoren geben. Hierzu werden zuerst einige messtheoretische Grundlagen und die Kommunikation mit den Sensoren behandelt. Anschließend wird auf die Verwendung der verschiedenen Sensorarten näher eingegangen.

1.2.1 Grundlagen der elektrischen Messtechnik

In der heutigen Welt dominieren digitale Geräte unseren Alltag. Einen ähnlichen Trend sieht man in der Messtechnik, in der zunehmend rein elektrische Messwertverarbeitung verwendet wird (Lerch, 2016, S. 3). Um eine nicht elektrische Messgröße elektrisch detektieren zu können, muss innerhalb des Messvorgangs die physikalische Größe des Informationsparameters gewandelt werden. Ein einfaches Beispiel stellt hierbei die Wandlung der physikalischen Größe Temperatur in einen Spannungsabfall an einem temperaturabhängigen Widerstand dar (Parthier, 2019). Dieses Wandlungsprinzip liegt der Funktionsweise der meisten Arduino-Sensoren zu Grunde.

Viele der für Arduino-Anwendungen verwendeten Sensoren liefern ein analoges Messsignal. Um dieses digital weiterverarbeiten zu können, ist zusätzlich die Wandlung in ein digitales Signal nötig. Bei einem solchen Bauteil spricht man von einem Analog-Digital-Wandler (A/D-Wandler). In dieser Ausarbeitung soll auf die prinzipielle Funktionsweise der Wandlung, nicht aber auf die genaue technische Umsetzung eingegangen werden. Der interessierte Leser findet genaue Beschreibungen hierzu beispielweise in den Büchern „Elektrische Messtechnik“ von Reinhard Lerch oder „Messtechnik“ von Rainer Parthier.

„Der unendliche Wertevorrat der analogen Größe wird auf einen endlichen Wertevorrat von Teilbereichen abgebildet.“ (Parthier, 2019, S.17). Diese Definition gibt ein erstes Gefühl, was bei der A/D-Wandlung geschieht: Ein kontinuierliches, aus unendlich vielen Messpunkten

bestehendes Signal soll durch endlich viele diskrete Messpunkte abgebildet werden. Die nachfolgende Abbildung soll diesen Vorgang schematisch erläutern:

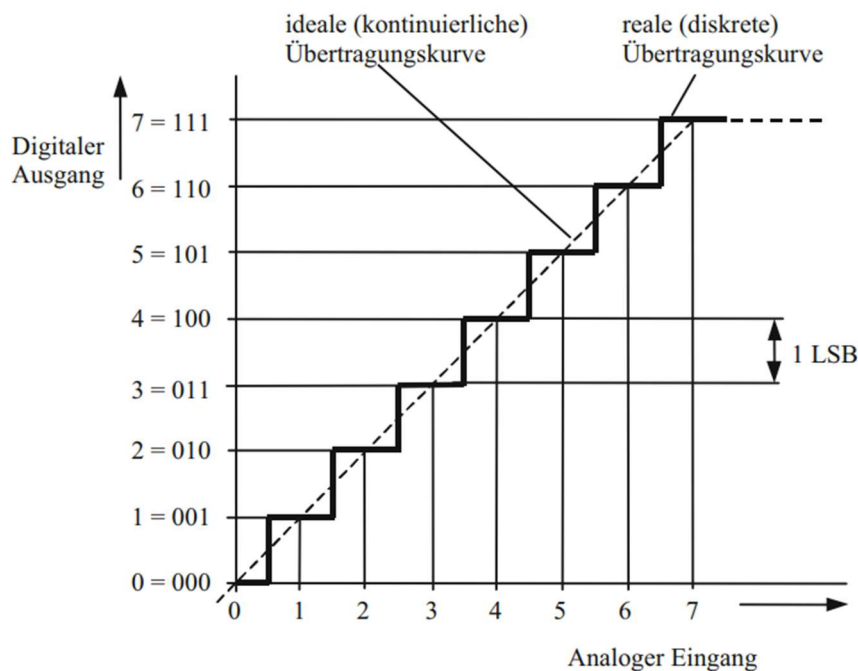


Abbildung 3: Schematische Abbildung der Wandlung eines analogen in ein digitales Signal. Das kontinuierliche Signal wird über die LSBs (Least-Significant-Bits) in diskrete Stücke zerlegt.

Bildquelle: (Parthier, 2019, S. 18)

Man sieht in der Grafik ein kontinuierliches Signal (gestrichelte Linie), das durch eine endliche Auflösung dargestellt wird (durchgezogene Linie). Ein „LSB“ ($\triangleq$ Least-Significant-Bit) bezeichnet hierbei die kleinste aufzulösende Änderung. Bei technischen Geräten wird diese Auflösung durch die so genannte „Bit-Kodierung“ festgelegt. Folgendes Beispiel soll dies erläutern (*analogRead*, o. D.):

Ein analoges Signal von 0-5V soll mit einem 10-Bit A/D-Wandler ($\triangleq$ Auflösung der meisten Arduinos) aufgelöst werden. Die 10-Bit Auflösung entspricht einer Anzahl von $2^{10} = 1024$ Werten, die einen Messbereich von 5V abzudecken haben. Man erhält damit eine theoretische Auflösung von etwa $\frac{5V}{2^{10}} \approx 4,9 \text{ mV}$.

Neben der Auflösung spielt auch die Abtastrate eine wesentliche Rolle bei der A/D-Wandlung eines Signales. Das Shannonsche Abtasttheorem (Parthier, 2019) gibt eine Bedingung an, damit eine Sinusschwingung aus dem digitalisierten Signal wieder regeneriert werden kann (Parthier, 2019, S. 19):

$$f_{\text{Abtastung}} > 2 * f_{\text{Signal}}$$

Bei der Arbeit mit Mikrocontrollern muss auf die oben genannten Bedingungen in vielen Fällen nicht geachtet werden. Bei der Aufzeichnung von Signalen mit hohen Frequenzen oder der Fehleranalyse sollte man sich dieser Grundlagen allerdings bewusst sein.

Je nach Art und Bauweise der Mikrocontroller-Boards besitzen diese verschieden viele analoge und digitale Eingänge. Bei der Verwendung eines analogen Eingangs wird die Wandlung des Signals direkt auf dem Board übernommen. Viele Sensoren verfügen des Weiteren über Versionen, die einen integrierten A/D-Wandler besitzen und so ein digitales Ausgangssignal liefern. Bei der Beschaffung der Sensoren sollte daher unbedingt darauf geachtet werden welches Ausgangssignal ein Sensor liefert. Soll ein Sensor mit analogem Ausgangssignal verwendet werden, aber kein analoger Input-Pin verfügbar sein, kann auf einen externen A/D-Wandler zurückgegriffen werden. Aufgrund der meist nur begrenzt verfügbaren analogen Inputs empfiehlt es sich in der Praxis für größere Projekte, Sensoren mit digitalem Ausgangssignal zu verwenden. Die prinzipielle Entstehung des digitalen Messsignals sollte im Laufe der Nutzung dennoch thematisiert werden, um dieses Thema den SchülerInnen bewusst zu machen. So kann der „Black-Box-Charakter“ einer Messung ein Stück weit aufgebrochen werden.

1.2.2 Die Verwendung von Sensoren mit analogem Ausgangssignal

Bei der Verwendung von Sensoren sollten in der Praxis zwei Dinge beachtet werden: Auf der einen Seite der Anschluss der Sensoren, auf der anderen Seite die Kommunikation mit diesen. Für Sensoren mit analogem Ausgangssignal gestaltet sich das wesentlich einfacher als für digitale Sensoren. In diesem Abschnitt soll die Verkabelung auf einem so genannten „Breadboard“ vorgestellt werden. In Abschnitt 1.2.5 findet sich ein neueres System, das eine einfachere Anwendung, beispielsweise für Schulzwecke, ermöglicht.

Für die Verkabelung der Sensoren in den ersten Projekten werden meistens so genannte Dupont- oder Jumper-Kabel verwendet. Hierbei handelt es sich um ein Steckersystem, das ohne Löten auskommt. Sollen mehrere Komponenten in einem Projekt verwendet werden, bietet es sich zusätzlich an, ein so genanntes „Breadboard“ zu verwenden. Das ist eine Steckplatine, die das Aufbauen komplexerer Schaltungen mit Steckverbindungen ohne Löten ermöglicht. Viele Sensoren lassen sich aber auch direkt auf das Breadboard stecken. Die Steckplatinen sind online in verschiedensten Größen verfügbar, sodass sich Projekte jeglichen Umfangs umsetzen lassen

(beispielsweise auf reichelt.de). Der Schaltplan einer solchen Steckplatine ist nachfolgend in Abbildung 4 dargestellt. An den beiden verbundenen „Bus-Leitungen“ links und rechts im Bild wird die Stromzufuhr der Sensoren mit der Stromversorgung des Mikrocontrollers verbunden. Alle weiteren Datenpins werden in der Mitte des Breadboards angeschlossen. Dort können die Sensoren auch direkt aufgesteckt werden.

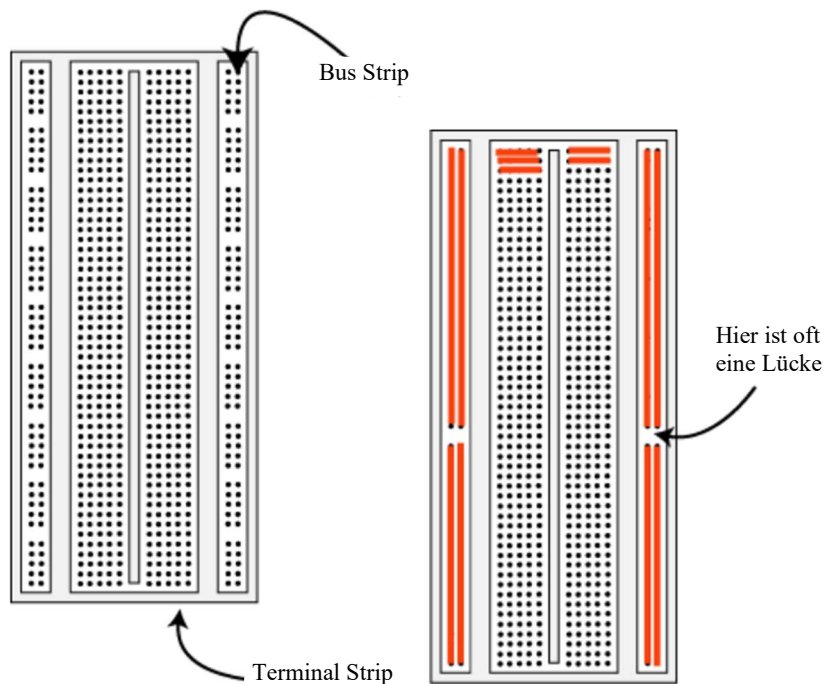


Abbildung 4: Aufbau eines Arduino-Breadboards mit zur Stromversorgung verwendeten Bus-Strips rechts und links sowie zur freien Verfügung stehenden Terminal-Strips. Die orangenen Linien visualisieren verbundene Steckplätze.

Bildquelle:
<https://www.seeedstudio.com/blog/2020/01/06/how-to-use-a-breadboard-wiring-circuit-and-arduino-interfacing/>
 (Abgerufen am 1.3.2023)

Um einen analogen Sensor an den Arduino anzuschließen, benötigt dieser zuerst eine Stromversorgung. Hierzu verbindet man die entsprechenden Pins des Sensors wahlweise direkt oder über das Breadboard mit den beiden Pins GND und 3,3V beziehungsweise 5V des Mikrocontrollers (abhängig von der Art des Sensors). Für die Datenübertragung wird ein weiteres Kabel verwendet, das den DATA-Pin des Sensors mit einem beliebigen analogen Input-Pin des Mikrocontrollers verbindet. Die analogen Inputs besitzen einen integrierten A/D-Wandler, bei den meisten Arduinos hat dieser eine Auflösung von 10Bit. Das analoge Messsignal wird dann intern digitalisiert und ausgelesen. Hierzu wird der Befehl „analogRead(A?)“ verwendet, wobei statt dem „?“ die Nummer des verwendeten analogen Input-Pins eingetragen wird. Nachfolgendes Bild soll einen Eindruck dafür geben, wie fertige Projekte mit einem Breadboard aussehen können. In diesem Projekt wird die Umgebungstemperatur und die Luftfeuchtigkeit mit dem Kombisensor DHT22 gemessen. Die Messwerte werden auf einem LC-Display ausgegeben, dessen Kontrast über ein Potentiometer geregelt werden kann.

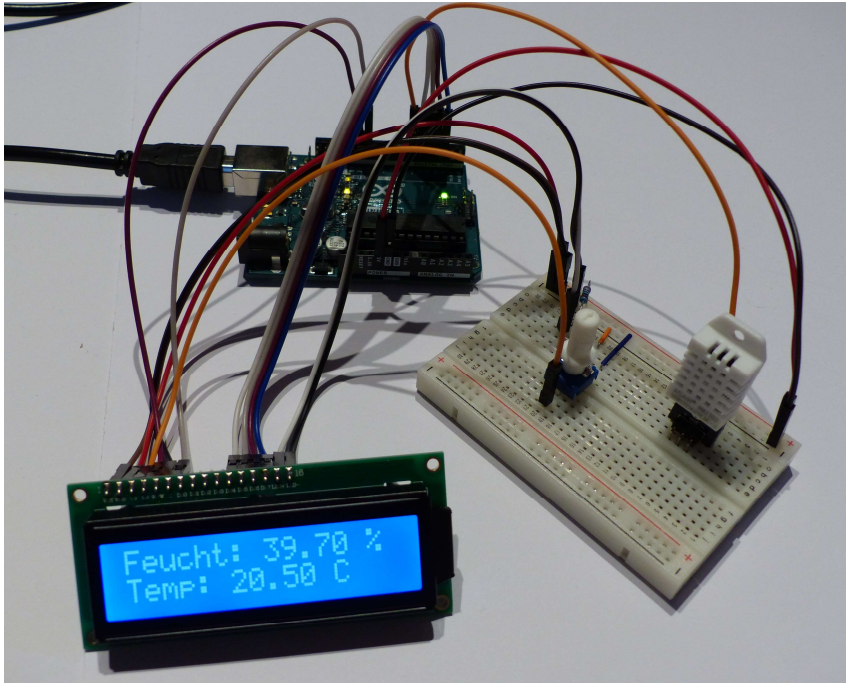


Abbildung 5: Beispielhafter Aufbau eines Arduino-Projektes mit einem Breadboard und Jumper-Kabeln. In diesem Versuchsaufbau werden die Temperatur und Luftfeuchtigkeit mit dem DHT22 gemessen. Die Werte werden auf einem LC-Display ausgegeben.

Bildquelle: Eigene Aufnahme

Schon an diesem Projekt mit nur zwei angeschlossenen Komponenten sieht man, dass die Verkabelung eines Projektes schnell kompliziert und unübersichtlich werden kann. Im Laufe der Arbeit werden deshalb mehrere Möglichkeiten vorgestellt, wie man diesen Arbeitsschritt für Lernende vereinfachen kann.

Ein Nachteil bei der Verwendung analoger Sensoren besteht darin, dass diese nicht direkt den gewünschten Messwert ausgeben. Der Mikrocontroller misst lediglich eine Spannung, aus der der Messwert berechnet werden muss. Um den Sensor einsatzbereit zu machen, muss deshalb eine Berechnung des Messwertes aus dem am Pin anliegenden Spannungswert implementiert werden. Der zu schreibende Code beschränkt sich hier meistens auf eine Zeile, die aus dem Datenblatt des Sensors übernommen werden kann. Dies kann Lernenden einerseits ein tieferes Verständnis der Messtechnik hinter einem Experiment ermöglichen, auf der anderen Seite aber auch ein weiteres Hindernis bei der Anwendung darstellen. Eine weitere Einschränkung bei der Verwendung analoger Sensoren besteht in der Anzahl der verfügbaren analogen Input-Pins, da viele günstigere Boards über maximal einen analogen Input-Pin verfügen.

1.2.3 Die Verwendung von Sensoren mit digitalem Ausgangssignal

Bei der Verwendung digitaler Sensoren muss man nach der Entstehung des Sensorsignales grundlegend zwei Arten unterscheiden: Auf der einen Seite Sensoren, die ein digitalisiertes Messsignal einer analogen Messgröße übertragen, auf der anderen Seite Sensoren, die prinzipiell nur zwei Zustände messen und ausgeben können. Die Sensoren der zweiten Art können analog zu Kapitel 1.2.2 an das Mikrocontroller-Board angeschlossen und über die Funktion „digitalRead(D?)“ ausgelesen werden. Sie übertragen ihre zwei Zustände über die Ausgabe von Spannungen. Abhängig von der Übertragungslogik gibt es hierfür fest definierte Spannungsbereiche für ein HIGH- und LOW-Signal (Chung & Bagur, 2023). Ein Beispiel für einen Sensor mit einem „echt digitalen“ Ausgangssignal ist ein Bewegungssensor, der nur die Zustände Bewegung JA oder NEIN erkennen kann.

Sensoren, die ein digitalisiertes Ausgangssignal einer analogen Messgröße liefern, haben einen integrierten A/D-Wandler. Deren praktische Verwendung ist allerdings etwas komplexer, da es mehrere Kommunikationsprotokolle gibt, mit denen auch die Art der Verkabelung variiert. Die Grundlagen dieser digitalen Kommunikationsprotokolle sind für Lernende nur schwer zu verstehen, weshalb man sich unter Umständen mit deren Funktion zufriedengeben und sie gerade bei jüngeren SchülerInnen nicht tiefer thematisieren sollte. Im Schulkontext ist es empfehlenswert, sich für den Einstieg auf ein gängiges Protokoll festzulegen.

Im folgenden Abschnitt wird das I²C-Protokoll („I-Quadrat-C-Protokoll“) als solches Beispiel vorgestellt, da die meisten Sensoren in einer I²C-kompatiblen Version verfügbar sind. Das universelle System eignet sich für Anfänger, da es nur zwei Kabel zur Datenübertragung benötigt und so den Verkabelungsaufwand verglichen zu anderen Protokollen, die teils vier Datenkabel benötigen, gering hält. Im I²C-Protokoll läuft die Datenübertragung über einen so genannten Datenbus, über den der Mikrocontroller mit mehreren Sensoren auf einer „Leitung“ kommunizieren kann. Der I²C-Datenbus benötigt zwei Kabel für die Kommunikation: Eine Datenleitung und eine Taktleitung. Diese bezeichnet man in der Praxis meist als SDA ($\triangleq$ Serial Data) und SCL ($\triangleq$ Serial Clock) (Margolis et al., 2020). Auf dem Datenbus regelt ein Master (meist der Mikrocontroller) die Kommunikation mit den verschiedenen Slaves (Sensoren oder andere Komponenten). Der Master sendet Kommandos über den Bus und die Slaves reagieren darauf, sodass mehrere Komponenten auf einer Leitung kommunizieren können (Brühlmann, 2019). Jeder Sensor besitzt eine I²C-Adresse. Solange sich die Adressen in einem Projekt nicht

doppeln, können theoretisch bis zu 128 Teilnehmer an einem Bus verwendet werden (Brühlmann, 2019). In der Praxis treten bei dieser Anzahl an Slaves allerdings weitere Einschränkungen auf. Die theoretische Teilnehmeranzahl soll lediglich zeigen, dass ein einziger I²C-Bus ausreicht, um auch größere Projekte zu realisieren. Um die Kommunikation auf Softwareseite zu implementieren, gibt es die „Wire.h“-Bibliothek, die eine fertige I²C-Kommunikationslösung liefert. Weitere Infos zu den Bibliotheken finden sich in Kapitel 2.3.

Bei der Beschaffung von I²C-kompatiblen Sensoren sind einige Dinge zu beachten: Zum einen die so genannten Pull-Up-Widerstände, die für eine zuverlässige Datenübertragung nötig sind. Für den Schulkontext sollten diese im Idealfall schon verbaut sein, da es sonst zu einer unzuverlässigen Datenübertragung kommen kann. Ein weiteres Kriterium ist die zur Übertragung verwendete Spannung. Viele Sensoren und beispielsweise auch die ESP-basierten Boards arbeiten mit einer Logik von 3,3V. Die meisten Arduino-Boards arbeiten allerdings mit einer internen Übertragungslogik von 5V. Die verschiedenen Spannungen können einen Sensor beschädigen oder zu Übertragungsfehlern führen. Es sollte also darauf geachtet werden, dass alle verwendeten Komponenten mit der Spannungs-Logik des Mikrocontrollers kompatibel sind, da man sich hierdurch das Einbinden eines zusätzlichen Logic-Level-Converters erspart. Weitere Informationen zu den Übertragungslogiken 3,3V und 5V findet man beispielsweise bei Chung und Bagur (2023) unter: <https://docs.arduino.cc/learn/microcontrollers/5v-3v3>.

1.2.4 Shields: Erweiterungsmöglichkeiten eines Boards

Hardware-Erweiterungen und aufsteckbare Zusatzplatinen nennt man im Arduino-Kosmos Erweiterungs-Shields oder einfach nur „Shields“. Diese Erweiterungen werden mit ihrer Pin-Leiste direkt auf das Mikrocontroller-Board aufgesteckt und benötigen somit keine weitere Verkabelung. Man spricht deshalb auch von einer „Sandwich-Montage“ (Brühlmann, 2019), die die Shields gerade für Anfänger und damit insbesondere für den Einsatz in Schulen geeignet macht. Diese Technik ermöglicht es, ein Projekt sehr einfach mit weiteren Funktionen auszustatten. Heute gibt es gerade für die gängigen Arduino-Boards wie den Arduino UNO eine Vielzahl an verschiedenen Shields. Im Folgenden soll ein kurzer Überblick gegeben werden, welche Möglichkeiten die Verwendung von Shields eröffnen (Liste nach Brühlmann, 2019):

1. Nachrüsten einer Internetverbindung

Über verschiedene Shields können Boards ohne native Internetanbindung mit dieser nachgerüstet werden. Hierzu gibt es Shields, die die Verwendung von Ethernet, WiFi oder mobilem Internet ermöglichen.

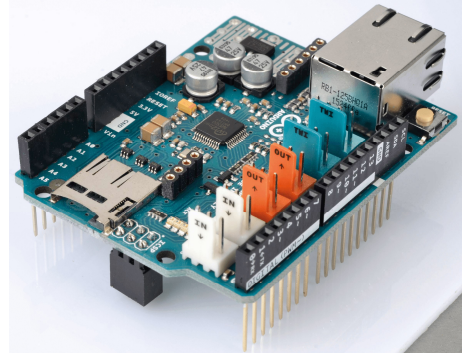


Abbildung 6: Beispiel eines Arduino UNO Erweiterungs-Shields. Dieses Shield beinhaltet unter anderem einen Ethernet- und MicroSD-Anschluss.

Bildquelle: <https://www.reichelt.de/arduino-shield-ethernet-shield-2-ohne-poe-w5500-arduino-shd-eth2-p159410.html> (Abgerufen am 8.3.2023)

2. Realisieren eines „Daten-Loggers“

Um Messdaten dauerhaft speichern zu können, muss der Mikrocontroller mit einem Speichermedium erweitert werden. Hierzu gibt es beispielsweise USB-, SD- oder MicroSD-Shields.

3. Verwenden einer externen Stromquelle

Die meisten Mikrocontroller-Boards werden über ein USB-Kabel mit Strom versorgt, weshalb man zwingend an einen Computer oder eine Steckdose gebunden ist. Möchte man allerdings, dass ein Projekt ortsunabhängig ist, benötigt man ein Shield, das die Versorgung mit Batterien oder einem Akku ermöglicht. So können beispielsweise Beschleunigungsdaten während einer Fahrt mit dem Auto, dem Fahrrad oder einer Achterbahn aufgezeichnet werden.

4. Das Vereinfachen der Verkabelungen

Über ein so genanntes Prototypen-Shield kann das in 1.2.2 beschriebene Breadboard direkt auf das Mikrocontroller-Board aufgesetzt werden. So können einfache Prototypen ohne weitere Kabel aufgebaut werden.

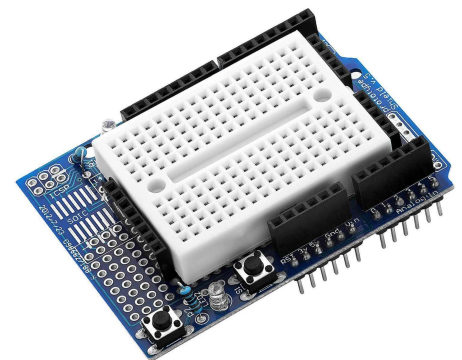


Abbildung 7: Arduino UNO Prototypen-Shield, zum vereinfachten Aufbau von Testschaltungen.

Bildquelle: www.az-delivery.de/products/prototyping-shield (Abgerufen am 8.3.2023)

5. Anbindung von nicht auf Mikrocontroller ausgelegten Komponenten

Neben den oben genannten Erweiterungen bieten Shields die Möglichkeit, verschiedenste weitere Komponenten in ein Projekt einzubinden. Hierzu gehören beispielsweise die Verwendung verschiedener Motoren aus dem Modellbau oder des „Wii-Nunchuks“. Die Erweiterung der für die Spielekonsole Wii entworfenen

Fernbedienung enthält einen dreiachsigen Beschleunigungssensor, der über ein passendes Shield in ein Projekt eingebunden werden kann.

Viele Shields stellen auf ihrer Oberseite die nicht benötigten Anschlusspins des Mikrocontroller-Boards als verfügbare Inputs bereit, sodass dessen Anschlussmöglichkeiten durch das aufgesetzte Board nicht eingeschränkt werden.

Für die originalen Arduino-Boards gibt es eine kleine Auswahl von Shields im Arduino-Webshop unter store.arduino.cc. Die Mehrzahl an verfügbaren Shields findet man bei anderen Anbietern. Arbeitet man mit einem Arduino-Klon, ist die Auswahl an verfügbaren Shields gegebenenfalls stark eingeschränkt. Dies ist ein weiteres Argument für den Einsatz der Arduino-Serie im Schulkontext.

1.2.5 Das Grove-System

Gerade Anfängern bereiten die oben vorgestellten Anschlussmöglichkeiten oft Probleme. Um in der Schule den Schwerpunkt nicht auf die Verkabelung der Sensoren oder die Kommunikation mit diesen legen zu müssen, sollte deren Verwendung möglichst intuitiv sein. Ein Ansatz hierfür sind fertige Sensorkomplexe mit „Plug and Play“-Steckersystemen, wie sie mittlerweile mehrere Firmen anbieten. Im Rahmen dieser Arbeit soll das Grove-System der Firma Seedstudio vorgestellt werden.

Die Firma Seedstudio bietet den so genannten „Seeduino“ an, ein Mikrocontroller-Board mit integrierten Grove-Anschlüssen. Alternativ gibt es für die gängigen Boards Grove-Erweiterungs-Shields (siehe Abbildung 8). Die Grove-Shields bieten vierpolige Anschlussbuchsen für alle Kommunikationsprotokolle. Die jeweilige Verkabelung wurde intern bereits vorgenommen, weshalb ein Sensor einfach an einer seinem Kommunikationsprotokoll entsprechenden Grove-Buchse angesteckt werden kann. Darüber hinaus muss man sich keine Gedanken über die

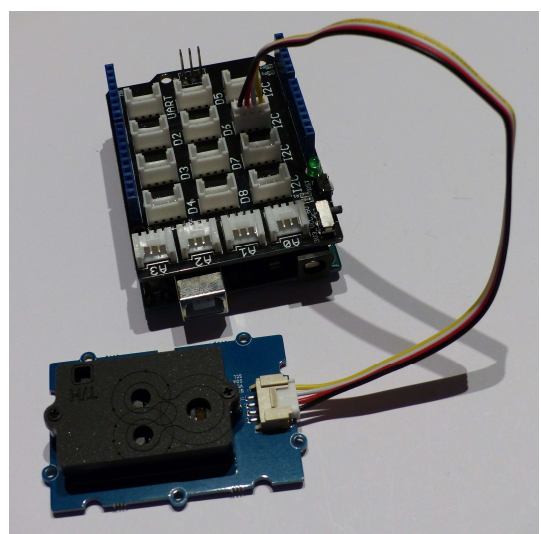


Abbildung 8: Auf dem Arduino UNO montiertes Grove Shield mit angeschlossenem CO₂-Sensor. Das Grove-System ermöglicht das wesentlich einfachere Verkabeln elektrischer Komponenten.

Bildquelle: Eigene Aufnahme

Stromversorgung oder die Verbindung der Datenleitung mit dem Arduino machen. Um die Verwendung der Grove-Sensoren weiter zu erleichtern, bietet Seedstudio für die meisten ihrer Grove-Komponenten Onlineanleitungen, Beispielprogramme sowie eine vorgefertigte Sensorbibliothek. So können auch ungeübte Anwender schnell erste Projekte realisieren, ohne dass tiefere Programmier- oder Elektronikkenntnisse erforderlich sind.

1.2.6 Beispiele

Wie bereits gezeigt, gibt es eine Vielzahl verschiedener Einsatzmöglichkeiten für Arduino-Sensoren. In der folgenden Übersicht sind beispielhafte Einsatzgebiete aufgeführt, für die gängige Sensoren im Handel sind. Als Quelle wurden hierzu im Wesentlichen drei verschiedene Onlineshops genutzt: Der offizielle Arduino-Store (store.arduino.cc), der Makershop (makershop.de) und der Onlineshop von ExpTech (exp-tech.de).

Sensorart	Messgröße/ Einsatzbereich der Sensoren	
Umweltsensoren	• Temperatur • Luftfeuchtigkeit • Bodenfeuchtigkeit	• Helligkeit • Windstärke • Luftdruck
Bewegungssensoren	• Abstand • Beschleunigung • Geschwindigkeit	• Neigung • GPS-Sensoren • Lichtschranken
Gassensoren	• CO₂-Gehalt • Sauerstoffgehalt	• Ozon-Gehalt* • Formaldehyd-Gehalt*
Elektro-Sensoren	• Elektrische Leitfähigkeit • Magnetfeldstärke	• Spannung • Stromstärke
Sonstige Sensoren	• Sensoren zum Schreiben eines EKGs	• Akustische Sensoren • RGB-Farbsensoren

*Tabelle 4: Übersicht über verschiedene mit Arduino-Sensoren messbare Größen mit konkreten Sensorbeispielen
: Die beiden Sensoren sind im Arduino-Store gelistet, laut Online-Berichten allerdings schon länger nicht lieferbar (Stand: 10.2.2023)

Den Sensoren zur Stromstärke- und Spannungsmessung kommt in dieser Liste eine spezielle Rolle zu, weshalb diese kurz thematisiert werden sollen. Prinzipiell kann jeder analoge Input-Pin des Arduinos zur Spannungsmessung zwischen dem Erdungslevel des Arduinos und dem angeschlossenen Spannungslevel verwendet werden. Für Spannungsmessungen benötigt man also eigentlich keinen externen Sensor. Da die Eingänge des Arduinos allerdings nur auf

Spannungen von 0-5V ausgelegt sind, benötigt man für das Messen höherer Spannungen einen Spannungsteiler. Dieser ist als externer „Sensor“ erhältlich, kann aber natürlich auch selbst aufgebaut werden. Ähnlich verhält es sich bei der Messung der Stromstärke über den Spannungsabfall an einem Widerstand. Auch diese Messung kann selbst aufgebaut werden oder als fertige Sensorlösung erworben werden. Die beiden Module stellen also keine Sensoren im eigentlichen Sinn dar, sondern eher eine Art Erweiterung der schon bestehenden Möglichkeiten.

Obige Liste soll einen Überblick über mögliche Messungen geben, jedoch keinesfalls als vollständig angesehen werden. Dennoch können aus der Liste erste Ideen zum Einsatz im Physikunterricht entwickelt werden. Weitere Anwendungsmöglichkeiten werden in Kapitel 4 thematisiert.

1.3 Weitere Bestandteile eines Projektes

Neben Sensoren kann ein Mikrocontroller-Projekt noch verschiedene weitere Komponenten beinhalten. Da viele dieser Komponenten keine direkte Einbindung in den Unterricht oder zumindest keine direkte Anbindung an den Lehrplan zulassen, soll dieses Kapitel lediglich einen kurzen Überblick über die weiteren Möglichkeiten geben.

Ein zentraler Bestandteil in der Planung eines Projektes besteht darin, wie die mittels Sensoren gewonnen Daten ausgegeben werden sollen. Die einfachste Möglichkeit besteht in der kabelgebundenen Ausgabe der Daten am seriellen Monitor des angeschlossenen Computers. Neben dieser auf einen Computer oder Laptop angewiesenen Option gibt es noch viele weitere Möglichkeiten, die sich in Projekten realisieren lassen. Beispiele hierfür sind das Integrieren einer 7-Segment-Anzeige, die einzelne Zahlen von Null bis Neun darstellen kann, oder das Verwenden moderner LCD- oder OLED-Displays (Beispiele aus Brühlmann, 2019). Um sich Programmier- und Verkabelungsaufwand zu sparen, empfiehlt es sich, die Displays über das I²C-Protokoll zu betreiben. Neben der direkten Ausgabe an einem Display gibt es auch die Möglichkeit der Datenweitergabe über kabelgebundene oder kabellose Schnittstellen, sowie die Möglichkeit der lokalen Datenspeicherung auf einer SD- oder MicroSD-Karte.

Alternativ oder zusätzlich zur Datenausgabe können auch so genannte Aktoren die Mikrocontroller-Projekte zum Leben erwecken: Aktive Komponenten führen bestimmte

Aktionen auf Befehl des Mikrocontrollers aus. Als Beispiele können LEDs, Schaltrelais, Summer beziehungsweise Tongeneratoren oder Motoren genannt werden (Beispiele aus Brühlmann, 2019). Gerade hier sind der Kreativität der Nutzer bei der individuellen Umsetzung keine Grenzen gesetzt.

2 Die Software

Nach der genaueren Betrachtung verschiedener Hardware-Komponenten soll nun die Software-Seite genauer beleuchtet werden. Hierzu werden in den ersten drei Teilkapiteln die Grundlagen und Besonderheiten der Programmierung eines Arduinos behandelt, bevor in Abschnitt 2.4 verschiedene Möglichkeiten zur Minimierung des Programmieraufwands vorgestellt werden. Diese ermöglichen es, die Technik im Unterricht auch ohne Programmierkenntnisse zu verwenden. Im letzten Teil des Kapitels wird das in einer modernen Welt immer wichtiger werdende Thema „Internet of Things“ und dessen Umsetzung mit einem Arduino thematisiert. Da der Umfang dieser Arbeit keinen detaillierten Einstieg in die Programmierung zulässt, wird hierzu auf zahlreiche Bücher verwiesen. Einen detaillierten Einstieg in die Arduino-Programmierung findet man beispielsweise im Buch „Arduino Cookbook“ von Margolis, Jepson und Weldin.

Eine feine Unterscheidung zwischen dem Mikrocontroller und dem darauf basierenden Board ist für den Rest der Arbeit nicht mehr nötig, weshalb von hier an mit dem Begriff „Mikrocontroller“ stets das fertige Board gemeint ist.

2.1 Die Arduino IDE: Wie programmiere ich einen Arduino?

Wie in Kapitel 1.1.2 bereits beschrieben, bündelt die Arduino-Plattform Hard- und Software, wobei gerade die Softwareseite für Nutzer ohne Vorkenntnisse in der Programmierung problematisch sein kann. Ein für Anfänger geeignetes Programm sollte die benötigten Vorkenntnisse daher auf ein Minimum beschränken, sodass Mikrocontroller auch im Schulkontext gewinnbringend eingesetzt werden können, bevor Lernende tiefere Programmierkenntnisse erwerben.

„*Arduino is designed to be used by nontechnical people [..]*“ (Margolis et al., 2020, Kap.1.0). Getreu diesem Motto hat Arduino eine Lösung entwickelt, die auch den Einstieg auf Software-Seite möglichst einfach gestaltet. Die so genannte „*Arduino-IDE*“ ($\triangleq$ **I**ntegrated **D**evelopment **E**nvironment) bietet eine Entwicklungsumgebung, mit der jeder Arduino und insbesondere auch alle damit kompatiblen Boards programmiert werden können. Das Programm steht für alle Betriebssysteme kostenlos auf der Arduino-Website zum Download bereit. Hierbei besitzen alle Versionen dieselbe Benutzeroberfläche und können daher gleich bedient werden (*Arduino Software*, o. D.), was gerade für den Schuleinsatz mit Lernenden, die verschiedene Betriebssysteme verwenden, einen Vorteil darstellt. Neben dem heruntergeladenen Programm existiert auch eine Web-Version der Programmierumgebung. Diese ist nach einer kostenfreien Anmeldung auf der Arduino-Website verfügbar, bietet allerdings nur begrenzten Platz zum Speichern der Projekte (*Arduino Software*, o. D.).

In der IDE kann geschriebener Code gespeichert, überprüft ($\triangleq$ „compiled“) und bei einer bestehenden USB-Kabelverbindung auf den Arduino geladen werden. Die Bedienung des Programms ist dabei so gestaltet, dass sie keine lange Einarbeitungszeit benötigt, weshalb auch die Funktionenleiste nur zwingend notwendige Funktionen enthält (vgl. Abbildung 9). Sobald in den Einstellungen der IDE das richtige Board und der USB-Port, an dem das Board angeschlossen ist, ausgewählt wurden, können fertige Testprogramme verwendet werden. Diese fertigen Projekte lassen beispielsweise die auf dem Arduino verbaute LED blinken und können als erste Funktionstests sowie als Einstieg in den Arduino-Programmcodes dienen. Dieser soll im nächsten Abschnitt genauer vorgestellt werden.

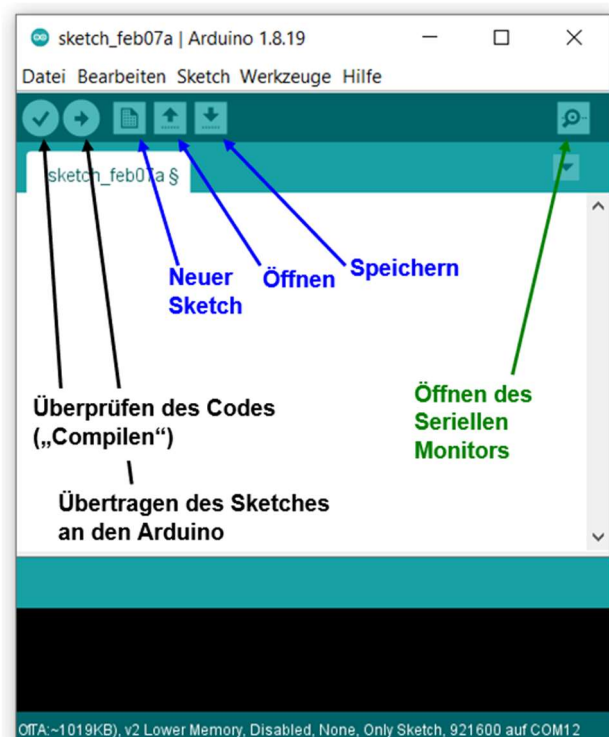


Abbildung 9: Benutzeroberfläche der Arduino IDE mit Kennzeichnung der wichtigsten Funktionen. Die Oberfläche ist sehr einfach gehalten, weshalb sie auch für Lernende geeignet ist.

Bildquelle: Eigene Aufnahme und Bearbeitung

Soll die Arduino-IDE mit einem Arduino kompatiblen Mikrocontroller wie dem ESP8266 oder dem ESP32 verwendet werden, müssen erst die „Support Files“ des Boards heruntergeladen werden. Dies kann über den Board-Manager des Programmes erledigt werden. Da es viele gute, kleinschrittige Anleitung hierzu gibt, wird an dieser Stelle hierauf verzichtet. Eine Anleitung findet man beispielsweise in Kapitel 10 des Buches „Arduino – Praxiseinstieg“ von Thomas Brühlmann.

2.2 Der Arduino-Programm-Code

Ein Arduino-Programm wird als „Sketch“ bezeichnet. In diesem Abschnitt soll das Erstellen eines solchen Sketches erläutert werden, wobei überwiegend auf die Struktur eines Sketches eingegangen wird. Die benötigten Programmierkenntnisse werden in dieser Arbeit nicht thematisiert. Eine Einführung hierzu findet sich unter anderem in Kapitel 2 des „Arduino Cookbooks“ (siehe Margolis et al., 2020).

Ein Arduino-Sketch besteht zwingend aus zwei Funktionen: Der Funktion „setup()“, die beim Start einmalig durchlaufen wird, und der Loop-Funktion, die anschließend als Dauerschleife durchlaufen wird. In der Setup-Funktion werden alle einmaligen Initialisierungsfunktionen aufgerufen. Hierzu gehört beispielsweise das Starten der seriellen Kommunikation mit allen beteiligten digitalen Komponenten, insbesondere dem seriellen Monitor des Computers, oder die Festlegung eines Pin-Modus (Output/Input). Auch wenn keine Befehle in der Setup-Funktion aufgerufen werden, muss sie zwingend implementiert sein (Brühlmann, 2019). Die Funktion „loop()“ beinhaltet die eigentlichen Aufgaben des Sketches, das heißt hier werden alle gewünschten Funktionsweisen implementiert. Der Arduino arbeitet die Anweisungen von oben nach unten ab und beginnt dann erneut am Anfang der Loop-Funktion. Hierbei ist es meistens sinnvoll die Funktion „delay(X)“ zu verwenden, die den Arduino X-Millisekunden warten lässt, bevor er weitere Anweisungen abarbeitet.

Für eine bessere Übersichtlichkeit des Sketches empfiehlt es sich speziell im Schulkontext, den geschriebenen Code in vier Teile zu gliedern. Im ersten Abschnitt werden alle benötigten Bibliotheken eingebunden (mehr hierzu in Abschnitt 2.3) und wichtige Variablen definiert. Eine Liste der unterstützten Datentypen findet sich bei Brühlmann, 2019. Anschließend folgen erst Setup- und dann Loop-Funktion. Im letzten Teil finden sich selbstgeschriebene Funktionen, die im Laufe des Programmcodes verwendet werden. Nutzer ohne Programmiererfahrung

können sich für den Anfang an einfachen Projekten mit Beispielcode orientieren und diesen nach den eigenen Wünschen anpassen. Dies ist einfacher als direkt den gesamten Sketch selbst zu entwerfen.

2.3 Arduino-Bibliotheken

In diesem Kapitel soll eine weitere Arduino-Lösung vorgestellt werden, die den Programmieraufwand in der Arduino IDE für Anfänger möglichst gering hält. Die so genannten „Arduino-Libraries“ bzw. Arduino-Bibliotheken erweitern einen Sketch mit Funktionen, die bereits von anderen Usern geschrieben wurden. Beispiele hierfür sind das Auslesen verschiedener Sensoren, das Einbinden digitaler Kommunikationsprotokolle und vieles mehr. Die öffentlichen Libraries ermöglichen es daher, unzählige Funktionen in ein Projekt einzubinden, ohne diese selbst programmieren zu müssen.

Das Einbinden der Bibliotheken kann auf zwei verschiedene Arten erfolgen: Bibliotheken, die über den Library-Manager direkt in der IDE verfügbar sind, können hier direkt gesucht und installiert werden. Im Library-Manager finden sich alle gängigen Bibliotheken, weshalb diese Möglichkeit für die meisten Anwendungen ausreicht. Da es mittlerweile aber Bibliotheken für nahezu jede denkbare Erweiterung gibt, sind nicht alle Erweiterungen über die IDE direkt abrufbar. Auf der Website GitHub (github.com) finden sich weitere Arduino-Bibliotheken und können von dort heruntergeladen werden. Die Einbindung einer externen Bibliothek gestaltet sich etwas komplexer, weshalb sie hier nicht weiter vorgestellt werden soll. Detaillierte Anleitungen zu den verschiedenen Installationswegen finden sich unter: <https://docs.arduino.cc/software/ide-v1/tutorials/installing-libraries>.

2.4 Arduino ganz ohne Programmieren - Geht das?

Beim Programmieren im Schulunterricht stellt die Syntax der Programmiersprachen und damit das selbstständige Programmieren ein zentrales Problem dar (Modrow, 2011). Dieses lässt sich auf die Mikrocontrolleranwendung übertragen, sodass gerade jungen Lernenden ohne Programmierkenntnisse der potenzielle Mehrwert der Mikrocontroller dadurch ein Stück weit vorenthalten wird. Das Problem liegt oft darin, dass der eigentliche Zweck der Mikrocontrolleranwendung schnell in den Hintergrund geraten kann, wenn zu viel Zeit auf die Programmierung verwendet werden muss. Verschiedene Programme und Projekte setzen

deshalb an dieser Problematik an und vereinfachen das „Schreiben von Code“ zum so genannten „visuellen Programmieren“.

Beim visuellen Programmieren gibt es vorgegebene Bausteine, die benutzerdefiniert angepasst werden können. Durch das Aneinanderreihen verschiedener Code-Bausteine entsteht puzzleartig das fertige Programm. Das visuelle Baustein-Programm wird automatisch in Arduino-Code übersetzt und kann anschließend an den Arduino weitergegeben werden (PGLU, 2022). Derartige Programme können die Verwendung von Mikrocontrollern wesentlich vereinfachen, ohne den Schritt der Programmierung gänzlich zu umgehen. Den Lernenden wird durch die grafische Darstellung des Codes der logische Ablauf des Programmes klar, weshalb der Einsatz solcher Programme im Schulkontext potenziell einen großen Mehrwert bieten kann (Modrow, 2011). Beispielprogramme für das visuelle Programmieren sind PGLU oder Visuino, wobei auch die Online Plattform Tinkercad eine derartige Funktion liefert. Gerade PGLU punktet hierbei mit einer einsteigerfreundlichen Benutzeroberfläche und auf das Programm abgestimmten Hardware-Boxen für den Einsatz in Schulen. Des Weiteren ermöglicht PGLU einen so genannten Hybridmodus, in dem klassischer und visueller Code kombiniert werden kann, sowie eine Simulation des Sketches in der seine Funktionsweise getestet werden kann (PGLU, 2022). Ein PGLU-Beispielsketch ist in Abbildung 10 dargestellt.



Abbildung 10: Beispielsketch in der visuellen Programmieroberfläche von PGLU. Am linken Bildrand können vorgefertigte Bausteine aus den dargestellten Kategorien ausgewählt werden. Diese Bausteine können im Hauptfeld zu einem Sketch zusammengepuzzelt werden. In der oberen Funktionsleiste kann man zwischen visuellem und klassischem Programmieren sowie der Simulation wechseln.

Bildquelle: Eigene Aufnahme, Sketch online erstellt auf <https://de.mach.pglu.ch/>

2.5 Arduino und das Internet of Things

„Es wird in wenigen Jahrzehnten kaum mehr Industrieprodukte geben, in welche die Computer nicht hineingewoben sind“ (Steinbuch, 1966 aus (Mattern & Flörkemeier, 2010, S.1)). Das Zitat aus dem Jahr 1966 zeigt, dass die Entwicklung des Internet of Things (IoT) schon vor über 55 Jahren absehbar war. Heute sind „smarte“ Haushaltsgeräte wie Staubsaugerroboter, per Smartphone steuerbare Lampen oder ein Kühlschrank, der automatisch Lebensmittel nachbestellt, in unserem Alltag angekommen. Eine Studie der Online-Plattform Statista prognostiziert für das Jahr 2027 eine Smart-Home Nutzungsrate von über 76% (Statista, 2022). All das zeigt, dass das Thema „Internet der Dinge“ eine immer wichtiger werdende Rolle im Alltagsleben einnimmt. Gerade jungen Menschen sollte deshalb eine differenzierte Sichtweise auf das Thema mit all seinen Chancen und Risiken ermöglicht werden.

2.5.1 Was ist das Internet of Things?

Unter dem Begriff Internet of Things versteht man *„eine Vision, in der das Internet in die reale Welt verlängert wird und viele Alltagsgegenstände ein Teil des Internets werden“* (Mattern & Flörkemeier, 2010, S.1). Dadurch werden technische Innovationen in nahezu allen Bereichen des menschlichen Lebens, aber auch die Automatisierung in vielen Bereichen der Industrie ermöglicht. Möchte man das Thema in der Schule behandeln, ist aber eine Betrachtung verschiedener Facetten nötig, um dessen Komplexität gerecht zu werden. Neben den technischen Innovationen bringt das Internet der Dinge nämlich auch gesellschaftliche und politische Herausforderungen mit sich (De Florio-Hansen, 2020). Gerade die Datenschutzproblematik ist ein öffentlich oft kritisierte Punkt des „neuen“ Internet of Things.

Viele Mikrocontroller ermöglichen die Einbindung eines Projektes in das IoT. Lernenden kann so ein einfacher Zugang in das Thema ermöglicht werden, der sowohl technische Grundlagen als auch Datenschutz- oder Infrastruktur-Probleme aufgreift. Im nächsten Teilkapitel wird eine kurze Einführung gegeben, wie sich IoT-Projekte mit einem Arduino umsetzen lassen. Im Verlauf der Arbeit wird in Kapitel 3.1 dann auf die Notwendigkeit der Thematisierung des Internet of Things in der Schule eingegangen.

2.5.2 Umsetzung von IoT-Projekten mit dem Arduino

Eine wesentliche Voraussetzung für die Einbindung eines Mikrocontrollers in das Internet of Things besteht in der Möglichkeit der Kommunikation mit anderen Geräten. Neben der üblichen Internetanbindung stehen auch andere Schnittstellen wie Bluetooth, Radioübertragung oder Near Field Communication (NFC) mittels Hardwareerweiterungen zur Verfügung (Margolis et al., 2020). In diesem Abschnitt sollen allerdings ausschließlich die Möglichkeiten zur Internetanbindung behandelt werden, da die Mikrocontroller hierüber direkt mit IoT-Services kommunizieren können.

Wie in Kapitel 1.1.2 bereits thematisiert, besitzen viele Arduinos standardmäßig keine Schnittstelle zur Internetanbindung, weshalb oft alternative Boards oder Hardwareerweiterungen von Drittanbietern verwendet werden müssen. Für die jeweiligen Hardware-Komponenten gibt es öffentliche Libraries, sodass sie ohne großen Programmieraufwand genutzt werden können. Im Folgenden werden drei verschiedene Internet-Schnittstellen vorgestellt.

Die erste Möglichkeit der Internetanbindung liegt in der Verwendung einer WiFi-Schnittstelle, wodurch das Board direkt mit dem Heim- oder Schul-WLAN verbunden werden kann. Eine komfortable Lösung den Mikrocontroller mit dem WLAN zu verbinden, liefert die Arduino-Bibliothek „WiFi-Manager“. Startet man nach Einbinden dieser Bibliothek das Programm auf dem Mikrocontroller, versucht sich dieser automatisch mit dem zuletzt benutzten WLAN zu verbinden. Sollte dies fehlschlagen, startet der Controller ein eigenes Netzwerk, in dem man sich mit jedem mobilen Endgerät einloggen kann. An diesem Endgerät öffnet sich nach der Verbindung mit dem Arduino-Netzwerk ein Anmeldefenster, in dem ein verfügbares WLAN-Netzwerk ausgewählt und dessen Passwort eingegeben werden kann. Wenn die Anmeldedaten korrekt sind, startet der Mikrocontroller neu und loggt sich automatisch in das vom Benutzer ausgewählte Netzwerk ein. Hierdurch kann vermieden werden, dass bei temporären Passwörtern oder der Benutzung verschiedener WLAN-Netzwerke Quellcode geändert werden muss.

Die zweite, kabelgebundene Lösung liegt in der Verwendung eines Ethernet-Kabels. Sie ist die wohl am einfachsten zu realisierende Möglichkeit, schränkt das Projekt allerdings auf die Nähe eines LAN-Anschlusses ein.

Die dritte Möglichkeit zur Internetverbindung besteht in der Verwendung des Mobilfunknetzes. Über eine SIM-Karte kann der Mikrocontroller mit dem mobilen Internet verbunden werden

und so völlig standortunabhängig Daten senden. Da hierzu die SIM-Karte eines Mobilfunkanbieters nötig ist, ist diese Lösung am aufwendigsten zu realisieren. Sie ermöglicht allerdings Messungen an Standorten, die keine WLAN- oder LAN-Verbindung ermöglichen.

Wenn ein Projekt Internetzugang hat, steht der IoT-Nutzung prinzipiell nichts mehr im Wege. Über digitale Kommunikationsprotokolle wie HTTP oder MQTT kann mit beliebigen anderen Geräten und Servern kommuniziert werden. Für die Implementierung dieser Kommunikation sind jedoch tiefere Programmierkenntnisse erforderlich, weshalb im Folgenden zwei Onlineservices vorgestellt werden, die auch für Anfänger geeignet sind.

Die einfachste Möglichkeit der Teilnahme am IoT besteht in der Nutzung der Arduino Cloud. Diese offizielle Arduino IoT-Lösung bietet eine sehr gute Dokumentation, weshalb sie auch für den Schuleinsatz geeignet scheint. Darüber hinaus liefert die Arduino Cloud bei Verwendung eines Arduino-Boards fertigen Code, um das Board mit dem Onlineservice zu verbinden. Man kann das Board also in die Cloud einbinden, ohne eigenen Code implementieren zu müssen. Neben beispielhaften Einsteigerprojekten bietet die Arduino Cloud konfigurierbare Dashboards, die Sensordaten und ausführbare Aktionen darstellen, sowie eine Smartphone-App, um die IoT-Projekte von überall überwachen zu können (*Arduino-Cloud*, o. D.). Um diese Features nutzen zu können, werden verschiedene „Plans“ angeboten. Je nach Funktionsumfang reicht deren Preis von 0 bis 19,99\$ monatlich (*Arduino-Cloud*, o. D.). Da die kostenlose Version allerdings nicht alle Features zur Verfügung stellt, ist man schnell auf ein kostenpflichtiges Abonnement angewiesen. Für Schulen bietet Arduino auch hier einen speziellen „School Plan“ an (*Arduino-Cloud*, o. D.). Als Fazit lässt sich festhalten, dass die Arduino Cloud eine geeignete Lösung für Schulen darstellt. Sollte aber der Low-Cost-Aspekt in einem Projekt entscheidend sein, scheint der nächste Service geeigneter.

Einen weiteren IoT-Service bietet die Firma Adafruit mit AdafruitIO. Auch diese Cloud bietet einen „Beginners Guide“, der das Einbinden der Cloud auch für Anfänger möglich macht. Wie die Arduino Cloud bietet auch AdafruitIO viele Beispielprojekte sowie vorgefertigte Dashboards, in der Sensordaten und Aktionen verwaltet werden können (Adafruit, o. D.). Leider sind sämtliche Informationen ausschließlich in englischer Sprache verfügbar und die Dokumentation ist auf Nutzer mit gewissen Mikrocontroller-Vorkenntnissen ausgelegt. Der große Vorteil von AdafruitIO liegt darin, dass die kostenlose Version des Services schon die Aktualisierung von 30 Datenpunkten pro Minute und eine Online-Datenspeicherung von 30

Tagen beinhaltet (Adafruit, o. D.). Diese Version ist daher für viele Projekte schon ausreichend und eignet sich gut für Low-Cost-Projekte. Bei der Nutzung von AdafruitIO hilft es allerdings, wenn die Nutzer schon eine gewisse Erfahrung mit Mikrocontrollern mitbringen. Ein Beispiel für die Nutzung von AdafruitIO mit einem ESP8266 basierten D1 mini wird in Kapitel 5 vorgestellt.

Neben diesen anfängerfreundlichen Versionen gibt es noch eine Vielzahl anderer Lösungen. Diese gehen von kostenlosen Services wie beispielsweise „dweet“ bis hin zu professionellen Lösungen wie der Verwendung des Amazon Web Services (AWS) oder Microsoft Azure. Die Nutzung dieser Lösungen erfordert allerdings tiefere Technikenntnisse, weshalb sie für den möglichst einfachen Einsatz in der Schule ungeeignet scheinen.

3 Didaktische Perspektive: Was bringt der Einsatz im Schulkontext?

Nachdem die ersten beiden Kapitel einen Einblick in die verschiedenen Bestandteile, Voraussetzungen und Möglichkeiten eines Arduino-Projektes gegeben haben, soll in den nächsten beiden Kapiteln konkreter auf deren Einsatz in der Schule eingegangen werden. Bevor im nächsten Kapitel verschiedene Einsatzmöglichkeiten vorgestellt werden, soll dieses Kapitel verschiedene Blickweisen auf den Mikrocontroller-Einsatz in Schulen aufzeigen. Hierbei sollen vor allem die Chancen des Einsatzes dargelegt werden, zum Schluss des Kapitels aber auch auf Kritikpunkte eingegangen werden.

3.1 Ein Beitrag zur Bildung in der digitalen Welt

Die Digitalisierung bedingt Veränderungen in allen Bereichen des Lebens. Aufgrund dieser gravierenden Neuerungen sprechen viele Publikationen in Anlehnung an die industrielle Revolution heute von einer „digitalen Revolution“ (De Florio-Hansen, 2020). Mit dem zunehmenden Wandel zu einer digitalisierten Gesellschaft ergeben sich auch für den Bildungsbereich neue Herausforderungen und Chancen. Auf diese geht die Kultusministerkonferenz (KMK) mit ihrem Strategiepapier „Bildung in der digitalen Welt“ ein. Das Papier befasst sich mit den verschiedenen Bereichen von Schule im digitalen Wandel und

thematisiert dabei neben der Unterrichtsentwicklung auch infrastrukturelle und rechtliche Rahmenbedingungen sowie die Aus- und Weiterbildung von Lehrkräften (KMK, 2017).

Im KMK-Strategiepapier werden Kompetenzen genannt, die für eine aktive und selbstbestimmte Teilhabe in einer digitalen Welt erforderlich sind (KMK, 2017). Der Erwerb dieser Kompetenzen geht über rein informatische Grundkenntnisse hinaus und kann daher nicht auf ein Fach beschränkt werden (KMK, 2017). Es ist deshalb nötig, dass alle Fächer im Rahmen ihrer jeweiligen Möglichkeiten zum Erwerb dieser Kompetenzen beitragen. Aus Sicht des Faches Physik bietet das Thema digitale Messtechnik eine physikspezifische Möglichkeit zur Bildung digitaler Kompetenzen. Durch den Einsatz von Mikrocontrollern kann neben diesem eher physikalischen Thema auch das „Internet der Dinge“ im Physik- und Informatikunterricht behandelt werden. So bieten sich verschiedene Optionen, die digitalen Kompetenzen zu stärken. Im Folgenden soll der Beitrag von Mikrocontroller-Anwendungen zur Entwicklung digitaler Kompetenzen an konkreten Kompetenzbereichen aus dem KMK-Strategiepapier dargelegt werden (Alle Kompetenzbereiche übernommen aus (KMK, 2017, S.16-19)):

1. *„(Digitale) Werkzeuge bedarfsgerecht einsetzen“*

Dieser Kompetenzbereich umfasst die vielfältige und kreative Anwendung verschiedener digitaler Werkzeuge, wie auch die Anpassung der Werkzeuge zum persönlichen Gebrauch. Über offene Schülerprojekte und Schülerexperimente mit Mikrocontrollern kann genau dieser Kompetenzbereich gestärkt werden.

2. *„Persönliche Daten und Privatsphäre schützen“*

Dieser in einer modernen Gesellschaft immer wichtiger werdende Bereich ist eng mit dem Schlagwort „Internet der Dinge“ verbunden. Sobald Daten über eine drahtlose Schnittstelle übertragen werden, stellt sich das Problem der Datensicherheit. Über „Smart-Home“-Schülerprojekte, bei denen Daten mit einem Mikrocontroller online gespeichert werden, lässt sich dieses Thema gut aufgreifen.

3. *„Speichern und Abrufen“*

Das Ausgeben, Speichern und Abrufen von Messdaten stellt eine zentrale Fragestellung jedes Mikrocontroller-Projektes dar. Auch im Rahmen des Physikunterrichts können verschiedenste digitale Möglichkeiten der Datenspeicherung und Aufbereitung

thematisiert werden. So können die Kompetenzen dieses Bereichs aus physikalischer Sicht entwickelt werden.

4. „An der Gesellschaft aktiv teilhaben“

Neben der Nutzung des Internets der Dinge bieten sich im Fach Physik weitere Möglichkeiten der digitalen Teilhabe an der Gesellschaft an, beispielsweise so genannte „Citizen Science“ Projekte. Heute gibt es viele Projekte dieser Art, bei denen man mit einer Mikrocontroller-Messstation aktiv an wissenschaftlichen Projekten teilhaben kann.

Neben diesen vier Kompetenzbereichen gibt es noch weitere, bei denen Mikrocontroller zur Kompetenzgewinnung beitragen können, beispielsweise der Bereich „*Algorithmen erkennen und formulieren*“ oder „*Technische Probleme lösen*“ (KMK, 2017, S.18). Allerdings sollte schon durch die vorgestellten Beispiele klar geworden sein, dass der Einsatz von Mikrocontrollern in der Schule eine Möglichkeit zur Berücksichtigung der veränderten Anforderungen an die Lernenden und den Unterricht darstellt.

3.2 Der Praxiseinsatz im Physikunterricht

Im folgenden Kapitel soll auf den konkreten Praxiseinsatz eingegangen werden. Hierbei sei schon zu Beginn des Kapitels klar vermerkt, dass alle Funktionalitäten eines Mikrocontroller-Projektes auch von gängigen Messwerterfassungssystemen oder digitalen Messgeräten erfüllt werden können. Der Einsatz eines Mikrocontrollers liefert also keine technisch neuen Funktionsweisen. Ihr offensichtlicher Vorteil liegt im Wesentlichen in der hohen Variabilität und dem niedrigen Preis. Mikrocontroller-Messsysteme können äußerst günstig umgesetzt werden. Das Beispiel eines Bluetooth Thermometers zeigt den massiven Unterschied: Ein smarter Temperatur-Sensor mit Bluetooth kostet bei den Firmen PASCO⁴ oder PHYWE⁵ etwas über 75€. Ein Thermometer, das die Daten über Bluetooth überträgt, lässt sich mit einem Mikrocontroller allerdings für unter 15€ realisieren. Natürlich bringen die teureren Modelle der Lehrmittelfirmen die Vorteile, dass sie sofort einsatzbereit sind und auf den Einsatz im Unterricht spezialisiert, weshalb sich die direkte Einbindung in den Unterricht hier einfacher

⁴ PASCO Smart Temperatursensor für 76,15€ auf schuchardt-lehrmittel.de, Stand: 6.2.2023

⁵ Cobra SMARTsense Temperature (Bluetooth) für 79€ auf phywe.de, Stand: 6.2.2023

gestaltet als bei der Mikrocontroller-Lösung. Der niedrige Preis ermöglicht allerdings eine Vielzahl an neuen Möglichkeiten. Es ist beispielsweise denkbar, dass jeder Schüler und jede Schülerin ein kleines Sensorkit mit nach Hause nehmen kann. So könnten geeignete Messungen oder Forschungsaufträge zuhause übernommen werden und man schafft es, Praxis statt Theorie in die Hausaufgaben zu integrieren. Des Weiteren bieten die günstigen Sensoren die Möglichkeit, dass die Lernenden allein experimentieren und nicht, wie oft üblich, mehrere SchülerInnen an einer Messapparatur arbeiten. Dadurch müssen sich Lernende beim Experimentieren nicht mehr abwechseln und jeder Einzelne kommt zu einer höheren aktiven Experimentierzeit. Eng zusammenhängend mit diesem Vorteil ist auch die hohe Variabilität. Basis-Hardwaresets ermöglichen schon eine Vielzahl von Messungen mit verschiedenen Möglichkeiten der Datenausgabe. Die Variabilität zeigt sich auch darin, dass alle Mikrocontroller-Komponenten miteinander kompatibel sind, weshalb man nicht an das System einer bestimmten Firma gebunden ist und bestehende Hardwaresets günstig erweitert werden können.

Neben diesen beiden Vorteilen nennen Raimund Girwidz und Bianca Watzka in einem Artikel zum Einsatz digitaler Werkzeuge im Physikunterricht fünf Kategorien, die den didaktischen Mehrwert von Mikrocontroller-Anwendung beschreiben sollen: (Girwidz & Watzka, 2018)

1. *Flexible Ausgabe*
2. *Mobilität*
3. *Schülerexperimente*
4. *Experimentelle Kompetenzen*
5. *Kontextorientierung*

Die einzelnen Kategorien sollen nachfolgenden kurz erläutert werden:

Der erste didaktische Mehrwert liegt nach Girwidz und Watzka in der flexiblen Ausgabe von Messwerten. Durch die digitalen und analogen Ausgänge eines Mikrocontrollers lassen sich die Signale eines Sensors auf verschiedenste Weise weiterverarbeiten. Qualitative Anzeigen wie das Aufleuchten einer LED reduzieren die Komplexität eines Messsystems und können daher den Einstieg in ein Thema vereinfachen. Neben der qualitativen Betrachtung eines Zusammenhangs ist meist auch eine quantitative Betrachtung notwendig, die den Vergleich von Messwerten und das Erkennen von Gesetzmäßigkeiten ermöglicht (Girwidz & Watzka, 2018). Auch in diesem Fall lässt sich die Datenausgabe auf verschiedene Weisen realisieren: Für einen

Demonstrationsversuch ist beispielweise die Ausgabe der Messwerte auf einem Monitor oder dem Beamer geeignet. Für Schülerversuche bietet sich die Datenaufzeichnung über das eigene Smartphone an. Je nach Anwendungskontext lassen sich gleiche Messungen also flexibel umsetzen.

Der zweite Vorteil liegt in der Mobilität der realisierbaren Projekte. Die Energieversorgung der Mikrocontroller ist über eine Batterie oder Power-Bank möglich, sodass die Messapparaturen völlig ortsunabhängig sind. Der Bezug zwischen dem Unterricht und der Lebenswelt der SchülerInnen kann gestärkt werden, indem physikalische Zusammenhänge im Alltag erkundet werden (Girwidz & Watzka, 2018). Als konkrete Beispiele sind die Aufnahme von Bewegungen (beispielsweise dem Nachhauseweg mit dem Fahrrad), die Datenerfassung mit einem Wetterballon oder eine Messung der Bodenfeuchtigkeit an mehreren Standorten denkbar.

Girwidz und Watzka führen die Realisierung von Schülerexperimenten, die aufgrund der Vielzahl an Sensoren aus verschiedensten Themenbereichen gewählt werden können, als weiteren Vorteil an. Anders als viele gängige Messwerterfassungen, die nur auf eine begrenzte Anzahl von physikalischen Zusammenhängen spezialisiert sind, können Schülerprojekte oder -experimente mit einem Mikrocontroller sehr frei gewählt werden. Bei der Themenwahl können daher lernförderliche, affektive Faktoren wie die Motivation oder Kontextwahl der Lernenden berücksichtigt werden (Girwidz & Watzka, 2018).

Die Realisierung eines Messsystems durch die Lernenden selbst bietet weitere didaktische Vorteile im Bereich der experimentellen Kompetenzen. Da eine selbstgebaute Apparatur seine Funktionsweise nicht verbirgt, wird der „Black-Box-Charakter“ vieler gekaufter Messwerterfassungssysteme aufgebrochen (Girwidz & Watzka, 2018). Lernende können so ein Verständnis für die Messtechnik hinter vielen Systemen erlangen. Des Weiteren zwingt eine selbstgebaute Messapparatur zum kritischen Hinterfragen der Messwerte, was einen wichtigen Schritt in der wissenschaftlichen Erkenntnisgewinnung darstellt.

Der letzte didaktische Mehrwert, den Girwidz und Watzka in ihrem Artikel aufgreifen, besteht in der Kontextorientierung. Dieser Vorteil bildet gleichzeitig eine Art Voraussetzung zum gewinnbringenden Einsatz von Mikrocontrollern. Da dieser viele Facetten hat, kann der Fokus auf verschiedene Aspekte wie die digitale Messtechnik, die Programmierung oder das

Verkabeln der Sensoren gelegt werden. Diese Aspekte haben oftmals nichts mit dem eigentlich zu erarbeitenden physikalischen Inhalt des Unterrichts zu tun. Die Nutzung der Technik sollte daher stets an den physikalischen Lerninhalten ausgerichtet sein. Der Fokus auf die zu untersuchenden physikalischen Zusammenhänge sollte stets gewahrt werden. Dann können Mikrocontroller eine Bereicherung durch die Einbindung physikalischer Zusammenhänge in attraktiven Kontexten darstellen (Girwidz & Watzka, 2018).

3.3 Authentizität des Experimentierens im Unterricht

Die Erkenntnisgewinnungskompetenz ist einer von vier zentralen Kompetenzbereichen in den Bildungsstandards des Fachs Physik (KMK, 2020). Die Kompetenzen des Bereichs zeigen sich beispielsweise durch die Kenntnis naturwissenschaftlicher Denk- und Arbeitsweisen, wobei die Bildungsstandards im Wesentlichen zwei Bereiche nennen: Die theoretische Beschreibung der Natur und die empirischen Methoden, vor allem das Experimentieren (KMK, 2020). Im Hinblick auf die zentrale Rolle des Experimentierens sollte man daher einen Blick darauf werfen, ob die Umsetzung des Experimentierens in der Schulpraxis die eigentliche naturwissenschaftliche Denk- und Arbeitsweise widerspiegelt.

„Wenn Schüler/innen experimentieren, wissen sie oft gar nicht was und warum sie etwas tun“ (Höttecke & Rieß, 2015, S.130). Das Zitat lässt den Schluss zu, dass die Umsetzung des Experimentierens in der Realität die in den Bildungsstandards verankerte Aufgabe nicht in vollem Umfang erfüllt. Dietmar Höttecke und Falk Rieß gehen 2015 in einem Beitrag in der Zeitung für Didaktik der Naturwissenschaften darauf ein, dass das Experimentieren im Schulkontext nicht viel mit dem Experimentieren in der wissenschaftlichen Praxis gemeinsam hat (Höttecke & Rieß, 2015). Da viele Lehrmittel stark vorstrukturiert sind und detaillierte Angaben zur Durchführung und Auswertung eines Versuches geben, kann man in der Schule eher von einem „Abarbeiten eines Kochrezeptes“ als von selbstständigem Experimentieren sprechen. Des Weiteren sind auch die verwendeten Messinstrumente darauf spezialisiert, Gesetzmäßigkeiten möglichst klar und zeitökonomisch darzustellen, sodass Messfehler oder eine kritische Hinterfragung der Messwerte oft vernachlässigt werden (Höttecke & Rieß, 2015). Genau an diesen Punkten kann der Einsatz von Mikrocontroller-Messsystemen hilfreich sein, da diese nicht didaktisch vorstrukturiert sind. SchülerInnen müssen sich daher tiefere Gedanken über die Anforderungen an ein geeignetes Messgerät sowie dessen Aufbau und Funktionsweise machen. Auch die Notwendigkeit einer kritischen Hinterfragung der Messwerte oder deren

Vergleich mit anderen Messgeräten wird Lernenden bei selbstkonstruierten Messsystemen eher bewusst.

Erfolgsversprechende Konzepte in der Physikdidaktik sind schon länger das selbstständige Forschen und Entdecken sowie offene Experimentierumgebungen (Höttecke & Rieß, 2015). Dennoch finden diese Konzepte aufgrund fehlender didaktischer Strategien wenig Beachtung in der Unterrichtspraxis, sodass sowohl Lehrkräfte als auch Lernende in einer Umfrage immer noch eine klare Unterscheidung zwischen Schulexperimenten und Experimenten in der Forschung machen (Höttecke & Rieß, 2015). Der Einsatz von Mikrocontrollern im Schulunterricht oder außerschulischen Kontexten wie Schüler-Forschungswettbewerben, die Gelegenheit zum selbstständigen Forschen bieten, kann einen Teil dazu beitragen, Konzepte wie das selbstständige Forschen und Entdecken in den Fokus zu rücken. So könnte das Experimentieren zumindest wissenschaftsähnlich gestaltet werden. Dies wäre ein Ansatz, den in den Bildungsstandards festgelegten Auftrag über die Vermittlung von wissenschaftlichen Denk- und Arbeitsweisen zu erfüllen.

3.4 Das Schulbuch der Zukunft

Der letzte didaktische Ansatz, der in diesem Kapitel behandelt werden soll, dreht sich um das so genannte „Schulbuch der Zukunft“. Hierbei handelt es sich um eine Vision, bei der sich erst zeigen muss, ob sie sich in Zukunft etablieren wird. Als Beispiel für ein Schulbuch der Zukunft wird das „Technology Enhanced Textbook“ (tet.folio) der Freien Universität Berlin verwendet, das eine webbasierte universell einsetzbare Lehr-Lern-Plattform liefert. Das digitale Schulbuch ermöglicht durch die Kombination von Elementen eines klassischen Schulbuches und neuen, multimedialen Angeboten neue Möglichkeiten für das Lernen, die über das reine Ersetzen der analogen Medien hinausgehen (Haase et al., 2021). Beispiele für diese neuen Lernangebote sind die Einbindung von Simulationen oder interaktiven Bildschirmexperimenten (IBE). Da jeder Schüler und jede Schülerin auf einer eigenen Version des Schulbuches arbeitet, können Aufgaben oder Notizen direkt in das digitale Schulbuch geschrieben werden, sodass im Laufe des Unterrichts eine personalisierte Version des vorgegebenen Schulbuches entsteht (Haase et al., 2016). Dies ermöglicht eine individuelle Schwerpunktsetzung der Lernenden in einem von der Lehrkraft vorgegeben Rahmen.

Viele dieser „Schulbücher der Zukunft“ sind in ihrer aktuellen Version noch nicht in der Breite einsetzbar. Neben vielen weiteren Hindernissen könnte das ein Grund sein, weshalb digitale Schulbücher bisher nur in wenigen Fällen genutzt werden. Eine Studie der Universität Göttingen zur Digitalisierung im Schulsystem aus dem Jahr 2021 zeigt, dass digitale Schulbücher in etwa der Hälfte aller Klassen gar nicht eingesetzt werden und sie von nur ca. 16% aller Klassen zumindest oft eingesetzt werden (Mußmann & Hardwig, 2021, S.18). Dennoch gibt es Grund zur Annahme, dass sich die Nutzung digitaler Schulbücher in Zukunft stark verbreitet. Das Saarland ist hierbei Vorreiter: Als erstes Bundesland deutschlandweit sollen digitale Bücher zum Schuljahr 2022/2023 in der Sekundarstufe I und II flächendeckend zur Verfügung stehen (Zierer & Gottfried, 2023). Aufgrund dieser Entwicklung scheint die Vision des Schulbuches der Zukunft wie sie tet.folio verfolgt auch weiterhin denkbar.

3.4.1 Mikrocontroller im Schulbuch der Zukunft

Inwiefern können Mikrocontroller zu einem Schulbuch der Zukunft beitragen? Bei vielen der neuen Lernarrangements arbeiten die SchülerInnen überwiegend oder ausschließlich digital. Bei einer übermäßigen Fokussierung auf rein digitale Experimentierangebote wie Simulationen oder Bildschirmexperimente besteht daher die Gefahr, dass praktische Experimentierkompetenzen immer mehr in den Hintergrund rücken. Da das Experimentieren aber ein wesentlicher Bestandteil der Physik ist und damit auch seine zentrale Rolle im Physikunterricht behalten sollte, muss man eine Möglichkeit finden reale Experimente in das Schulbuch der Zukunft einzubinden.

Durch die Möglichkeit der Internetanbindung können Mikrocontroller eine Schnittstelle zwischen dem digitalen Schulbuch und einem realen Experiment darstellen, indem die Messdaten vom Mikrocontroller direkt in das digitale Schulbuch eingebunden werden. So ist es möglich, Messwerte im Schulbuch digital zu protokollieren und auszuwerten. Der Vorteil gegenüber anderen Messgeräten, die eine Internetverbindung ermöglichen, liegt darin, dass diese oftmals nicht programmierbar sind und daher nur mit dafür vorgesehenen Online-Services kommunizieren können. Aufgrund der Offenheit des Mikrocontroller-Systems können diese hingegen mit etwas Programmierkenntnissen in nahezu allen serverbasierten Anwendungen eingebunden werden. So können weitere Beispiele für die

Zusammenarbeit digitaler Lernarrangements mit realen Experimenten, wie der Vergleich der Messdaten mit Daten einer Simulation oder ein digitales Planungstool für reale Experimente umgesetzt werden.

3.4.2 Die Einbindung von Mikrocontrollern in tet.folio

Je nach den individuellen Anforderungen und dem Einsatzzweck gibt es mehrere Möglichkeiten, einen Mikrocontroller in ein digitales Schulbuch, beispielsweise tet.folio, einzubinden. Da eine genaue Anleitung die Grenzen dieses Unterkapitels sprengen würde, wird lediglich ein Überblick über zwei Anbindungsmöglichkeiten gegeben und dabei auf weiterführende Literatur verwiesen.

Die erste direkte Möglichkeit der Anbindung besteht darin, dass der Mikrocontroller direkt mit dem tet.folio-Server kommuniziert. Experimente, in denen viele Daten in einer kurzen Zeit aufgezeichnet werden sollen und die nicht auf langfristige Messungen ausgelegt sind, sind hierüber gut realisierbar. Die direkte Anbindung bedeutet allerdings einen höheren Programmieraufwand, da nur bedingt auf fertige Lösungen zurückgegriffen werden kann. Der Ablauf der Datenübertragung ist schematisch auf der nächsten Seite dargestellt. Die Sensoren kommunizieren mit dem Mikrocontroller und dieser wiederum über eine Internetschnittstelle mit dem tet.folio-Server. Von dort aus können beliebige Clients auf die online gespeicherten Daten zugreifen. Genauere Informationen zur technischen Umsetzung und der Programmierung der jeweils benötigten Schnittstellen finden sich in einer Bachelorarbeit von Jannes Siems zur „Webbasierten Erfassung und Visualisierung von Messwerten“. Diese ist auf tet.folio abrufbar unter: <https://tetfolio.fu-berlin.de/tet/850527>.

Eine andere Möglichkeit der Einbindung lässt sich über den Zwischenschritt eines IoT-Servers (siehe hierzu auch Kapitel 2.5.2) realisieren. Diese Option hat technische Einschränkungen wie die Limitierung der Messwerte pro Minute und eine beschränkte Anzahl an Messvariablen, liefert allerdings schon fertige Schnittstellen, sodass große Programmierarbeit vermieden werden kann. Die zur Anbindung benötigten Codezeilen können meist aus Onlineanleitungen der IoT-Services übernommen werden. Wenn die Daten auf einem IoT-Server gespeichert sind, können sie von dort über ein so genanntes „Application Programming Interface (API)“ vom tet.folio-Server abgefragt werden. So gelangen die Messdaten über den Umweg eines IoT-Servers auf die gewünschten tet.folio-Seiten. Die Möglichkeit bietet sich gerade für

Schülerprojekte an, da sich die Anbindung über diesen Umweg wesentlich einfacher gestaltet als die direkte Anbindung. Ein Vergleich der Datenübertragungswege ist in nachfolgender Abbildung dargestellt.

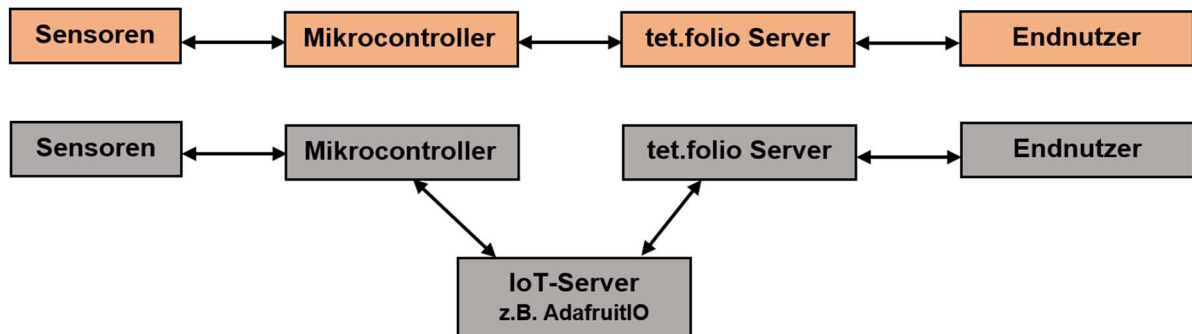


Abbildung 11: Vergleich der Datenübertragungswege bei direkter Anbindung des Mikrocontroller an tet.folio (orange) und der indirekten Anbindung über einen IoT-Server (grau). Beim grauen Datenübertragungsweg, kann für die zweite und teilweise auch für die dritte Schnittstelle auf fertige Lösungen zurückgegriffen werden, weshalb dieser potenziell einfacher zu realisieren ist.

Bildquelle: Eigene Grafik, Teile vereinfacht übernommen von Siems, 2018 (abrufbar unter <https://tetfolio.fu-berlin.de/tet/850527>)

Beide Optionen erfordern allerdings mindestens Grundkenntnisse im Programmieren. Für eine breitere Nutzung der Mikrocontroller im Kontext eines digitalen Schulbuches muss es daher „ready to use“ Anwendungen geben. Diese sollen Lehrkräften und Lernenden ermöglichen, ihre Projekte in ein digitales Schulbuch einzubinden, ohne sich Gedanken über Programmierung oder sonstige technische Feinheiten machen zu müssen.

3.5 Kritische Hinterfragung des Schuleinsatzes

Neben den oben genannten Argumenten für einen Einsatz von Mikrocontrollern im Schulunterricht, gibt es selbstverständlich auch Punkte, die gegen einen Einsatz sprechen. Der wohl größte Punkt, der vom Mikrocontrollereinsatz in der Schule abhält, ist der schwierige Einstieg. Für einen gewinnbringenden Einsatz brauchen SchülerInnen Erfahrung im Umgang mit den Mikrocontrollern, weil für Anfänger gerade die Programmierung eines Sketches oder das Verkabeln der ersten Projekte eine große Hürde darstellt. Will man allerdings das volle Potenzial der Mikrocontroller ausschöpfen, sodass SchülerInnen selbstständig arbeiten können und eben kein Kochrezept Schritt für Schritt durcharbeiten, genügt es nicht, Sketche und Anleitungen für die Lernenden vorzubereiten. Es stellt sich daher die Frage, ob die Zeit, die man darauf verwendet, die SchülerInnen an die neue Technik heranzuführen, in einem sinnvollen Kosten-Nutzen-Verhältnis zum didaktischen Mehrwert steht. Wenn die Nutzung der

Mikrocontroller auf wenige Anwendungen während der gesamten Schullaufbahn beschränkt ist, ist die zu investierende Zeit mit Sicherheit zu groß. Schafft es eine Schule hingegen, die Mikrocontroller schon früh und fächerübergreifend in den Unterrichtsalltag zu integrieren, sodass sich vielseitige Anwendungsmöglichkeiten in mehreren Fächern bieten, kann der Einsatz den Unterricht stark bereichern. In diesem Fall schafft man es auch, die Einstiegszeit auf mehrere Fächer zu verteilen. So ist es beispielweise denkbar, die elektronischen Grundkenntnisse im Fach Physik und die benötigten Programmierkenntnisse im Informatikunterricht zu vermitteln. Für einen möglichst einfachen Einstieg spielen auch die Nutzung der stetig weiterentwickelten Systeme auf Hard- und Softwareseite, zum Beispiel das Grove-System (siehe Kapitel 1.2.5) oder die Programmieroberfläche PGLU (siehe Kapitel 2.4), eine wichtige Rolle.

Ein weiterer Kritikpunkt bezieht sich auf den oft genannten Vorteil des „Aufbrechens der Black-Box“. Die Lernenden können mit einem selbstgebauten Messsystem einiges über den Aufbau und die Datenübertragung in einem digitalen Messgerät lernen. Aber wie viel lernen sie über die wirkliche Funktionsweise des Gerätes? Betrachtet wird dies am Beispiel eines digitalen Thermometers: Auch beim Aufbauen des Mikrocontroller-Messsystems bleibt die eigentliche Funktionsweise des Sensors verborgen. Erfolgt in der Praxis keine Reflexion darüber, wie die gemessene Temperatur im Sensor in eine Spannung umgewandelt wird, kann man maximal von einer „Verkleinerung der Black-Box“, nicht aber von einem „Aufbrechen der Black-Box“ sprechen. Ein weiterer problematischer Punkt sind hierbei Hilfssysteme wie das „Grove-System“. Diese vereinfachen den Einstieg und die Anwendung der Mikrocontroller deutlich, wodurch sie allerdings die Komplexität reduzieren und verschiedene Funktionsweisen wie die eines A/D-Wandlers verbergen. Um die Black-Box der Messsysteme im Unterricht möglichst weit aufbrechen zu können, muss eine Lehrkraft daher je nach Vorkenntnissen und Alter der Lernenden abwägen, welche Vereinfachungen sie zulässt.

Aus dem oben genannten Kritikpunkt kann man bereits schließen, dass der Erfolg des Mikrocontroller-Einsatzes wesentlich von der Lehrkraft abhängt. Klaus Zierer geht in seinem Buch „Lernen 4.0 - Pädagogik vor Technik“ auf genau diesen Sachverhalt ein. Laut Zierer kann der Einsatz von neuer Technik allein nicht zu einer Revolution des Lernens führen, da die Lehrkraft und dabei insbesondere ihre Kompetenzen und Haltung zu Technik eine entscheidende Rolle für den Erfolg der neuen Medien spielt (Zierer, 2017). Er führt weiter aus,

dass eine Digitalisierung im Unterricht nur dann hilfreich sein kann, wenn sie kein Selbstzweck ist, sondern...

1. *Die Lernausgangslage berücksichtigt*
2. *Herausfordert*
3. *Vertrauen aufbaut und Zutrauen ermöglicht*
4. *Fehler sichtbar macht*
5. *Gespräche über den eigenen Lernprozess initiiert* (Zierer, 2017, S.52).

Soll das volle Potenzial der Mikrocontroller ausgenutzt werden, müssen die oben genannten Anforderungen bei der Verwendung im Schulunterricht erfüllt werden. Werden diese Grundsätze nicht beachtet, besteht die Gefahr, dass die Technik vor die Pädagogik tritt (Zierer, 2017, S.101).

Ob sich der Einsatz von Mikrocontrollern im Schulunterricht allerdings in der Breite durchsetzt, scheint eher fragwürdig. Eine Studie von Wilhelm und Trefzger aus dem Jahr 2010 zeigt, dass etwa ein Drittel aller Gymnasialkräfte nur geringes oder sehr geringes Interesse am Einsatz von PC-Messwerterfassungssystemen hat und etwa zwei Drittel die Systeme höchstens vereinzelt verwenden (Kurth & Wodzinski, 2018). Da diese Ergebnisse in späteren Studien bestätigt wurden, ist es schwer vorstellbar, dass der Einsatz von Mikrocontrollern bei einem Großteil der Lehrkräfte auf große Akzeptanz stößt.

Als Fazit dieses Kapitels lässt sich festhalten, dass für einen gewinnbringenden Mikrocontrollereinsatz in Schulen bestimmte Voraussetzungen erfüllt sein müssen. Eine besondere Rolle spielt hierbei die Lehrperson. Sind diese Voraussetzungen allerdings hinreichend erfüllt, kann der Mikrocontrollereinsatz wesentlich zur Verbesserung des naturwissenschaftlichen Unterrichts beitragen. Beispielhaft sind hierbei der Beitrag zur Bildung digitaler Kompetenzen oder das wissenschaftsähnlichere Experimentieren mit Mikrocontrollern zu nennen. Neben den didaktischen Vorteilen eröffnet der geringe Preis neue Möglichkeiten im Unterrichtsalltag, wie die Ausstattung aller Lernenden mit einem eigenen Messsystem. Um das große Potenzial der Mikrocontroller-Anwendungen ausnutzen zu können, bedarf es dennoch weiterer didaktischer Konzepte, die sich nicht auf einzelne Kontexte beschränken, sondern ein Gesamtkonzept über mehrere Jahrgangsstufen und Fächer liefern.

4 Wie lassen sich Mikrocontroller in der Schule konkret einsetzen?

Nachdem im letzten Kapitel die theoretischen Vorteile des Einsatzes von Mikrocontrollern diskutiert wurden, soll dieses Kapitel Ideen zu konkreten Einsatzmöglichkeiten liefern. Da dieses Thema allein eine ganze Arbeit füllen könnte, wird bei den Beispielen auf detaillierte Anleitung und die Angabe von konkreten Codezeilen verzichtet. Das Ziel des Kapitels ist es nicht, die jeweiligen Experimente Schritt für Schritt zu erklären, sondern eine Vorstellung zu liefern, in welchen Kontexten Mikrocontroller eingesetzt werden können. Die verschiedenen Einsatzgebiete werden für dieses Kapitel in drei Bereiche untergliedert: Der erste Bereich umfasst Themen der klassischen Physik, bevor im zweiten Abschnitt auf Einsatzmöglichkeiten in der modernen Physik eingegangen wird. Der letzte Bereich befasst sich mit Ideen für fachübergreifende Projekte.

4.1 Klassische Physik

In diesem Kapitel sollen ausgewählte Mikrocontroller-Experimente aus den Bereichen der klassischen Physik vorgestellt werden. Neben den vorgestellten Experimenten finden sich viele weitere Ideen in einschlägiger Literatur oder Internet-Beiträgen. Speziell für den Schulgebrauch soll auf zwei Werke aufmerksam gemacht werden: Zuerst das Buch „Physik-Experimente mit Arduino – 36 faszinierende Projekte für Schule und Hobby“ von Willem von Dreumel, in dem auch auf die Funktionsweise der verwendeten Sensoren eingegangen wird. Speziell für den Unterricht ist das Heft „Arduino, Rasperry Pi & Co. – Alltagsphysik und Messtechnik verstehen mit digitalen Werkzeugen“ aus der Serie „Naturwissenschaften im Unterricht Physik“ zu empfehlen, da hier neben den Grundlagen der Experimente auch Einbindungsmöglichkeiten in den Unterricht behandelt werden.

Im Themenbereich der Mechanik bieten sich Experimente zu Fallbewegungen mit Lichtschranken oder die Analyse verschiedener Pendelbewegungen an. Beispielhaft soll hier eine Messung am Federpendel vorgestellt werden (Dreumel, 2016): Über einen Arduino-Abstandssensor, der auf dem Tisch unter dem Federpendel platziert wird (vgl. Abbildung 12), kann die Schwingung des Pendels verfolgt werden. Über eine im Sketch implementierte Stoppuhr lässt sich aus dem Abstand des Pendels zum Sensor anschließend die Schwingung in einem Graph visualisieren (vgl. Abbildung 13).

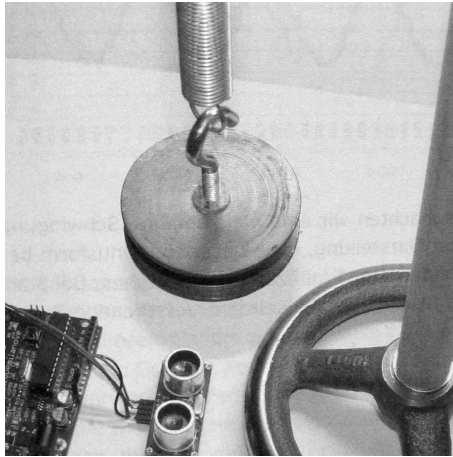


Abbildung 12: Versuchsaufbau zur Analyse der Schwingung eines Federpendels mit dem Arduino

Bildquelle: (Dreumel, 2016, S.49)

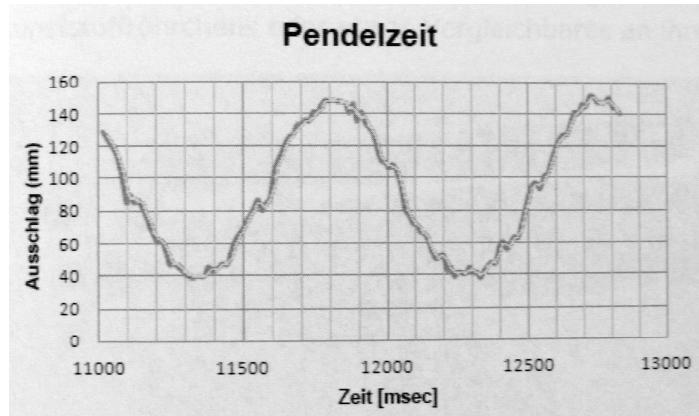


Abbildung 13: Messgraph der Schwingung eines Federpendels mit dem Arduino. Der Abstand wird mit einem Ultraschallabstandssensor gemessen, die Zeit mittels einer im Sketch implementierten Stoppuhr. Bildquelle: (Dreumel, 2016, S.51)

So können unter anderem die Schwingungsdauer und Amplitude der Pendelschwingung bestimmt werden. Hieraus lässt sich die Federkonstante der verwendeten Feder bei bekannter Masse des angehängten Gewichts ermitteln und beispielsweise mit dem in Ruhe über die Auslenkung der Feder bestimmten Wert vergleichen. Das Experiment bietet den Vorteil, dass der Versuchsaufbau relativ einfach ist, sodass Lernende das Experiment bei Vereinfachung der Programmierung auch in Einzelarbeit oder zuhause durchführen können. Im Buch „Physik-Experimente mit Arduino“ finden sich neben diesem Beispiel noch weitere Messungen aus der Mechanik wie die Realisierung eines Kraftmessers, Messungen zur Elastizität oder verschiedene Ballistik-Experimente.

Das Thema erneuerbare Energien spielt in unserer modernen Welt eine immer wichtiger werdende Rolle. Auch bei der Erforschung dieses Themas können Mikrocontroller behilflich sein. Sie ermöglichen es beispielsweise, die Leistung einer Solarzelle oder eines kleinen Windrades über mehrere Tage zu beobachten. So lässt sich die abgegebene Leistung in Abhängigkeit der Tageszeit und den Parametern Helligkeit beziehungsweise Windstärke untersuchen. Des Weiteren können die SchülerInnen den optimalen Standort der Kraftwerke suchen, welcher eine Maximierung der abgegebenen Leistung pro Tag ermöglicht. Neben der Optimierung des Standortes lässt sich mit Mikrocontrollern auch ein so genannter „Solar Tracker“ realisieren. Dieses fortgeschrittene Experiment ermöglicht es, über Schrittmotoren die

Ausrichtung des Solarpanels automatisch so anzupassen, dass dessen Leistung maximiert wird.⁶ Ein schematischer Versuchsaufbau ist in Abbildung 14 dargestellt. Durch die beiden Servomotoren kann die Ausrichtung der Solarzelle zweiachsig angepasst werden. Die jeweils optimale Ausrichtung wird aus den Signalen der vier Fotodioden (LDR $\triangleq$ Light Dependent Resistor) am Mikrocontroller berechnet. Im Bereich der Elektrizitäts-Lehre bieten sich neben diesem Experiment auch Messungen zur Induktion an einer Spule (Dreumel, 2016) oder dem Ein- und Ausschaltvorgang einer Spule und eines Kondensators an (Kurth & Wodzinski, 2020).

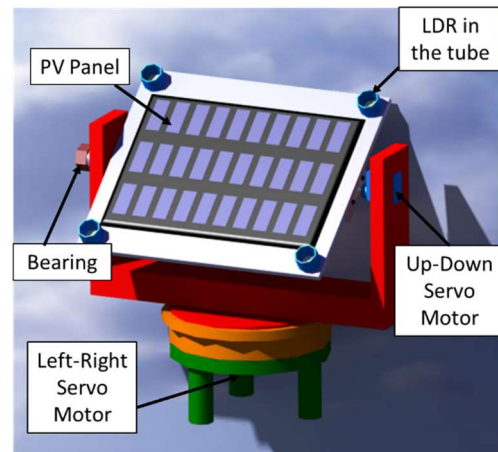


Abbildung 14: Schematischer Aufbau des Projektes „Solar Tracker“ zur automatischen Maximierung der abgegebenen Leistung einer Solarzelle. Bildquelle: https://projecthub.arduino.cc/Aboubakr_Elhammoumi/77347b69-2ade-4a44-b724-3bb91e954188 (Abgerufen am 1.3.2023)

Die nächsten Experimente behandeln den Themenbereich Optik und sollen erläutern wie moderne LEDs funktionieren. Neben den Grundlagen der additiven Farbmischung können SchülerInnen das Prinzip der Pulsweitenmodulation in verschiedenen Einzelversuchen erkunden. Die Pulsweitenmodulation stellt durch Variation der Länge des „An“-Zustandes bei schnellem Ein- und Ausschalten eine effiziente Möglichkeit dar, eine LED zu dimmen. Die Helligkeit der LED ist dabei in guter Näherung direkt proportional zur Dauer des „An“-Zustandes. Da die Frequenz der Schaltvorgänge deutlich über 200Hz ist, kann das menschliche Auge das Flackern der LED nicht wahrnehmen (Hoyer et al., 2018). Über die Pulsweitenmodulation kann aus den drei Grundfarben Rot, Blau und Grün additiv jede andere Farbe gemischt werden (Hoyer et al., 2018). Die Anleitungen für fünf Versuche zu diesem Themengebiet finden sich im Heft „Arduino, Raspberry Pi & Co“.

„Woher wissen Funkuhren, wie spät es ist?“ (Richtberg, 2018). Mit dieser Frage beschäftigt sich das letzte Experiment, das in diesem Abschnitt thematisiert werden soll. In diesem Versuch können die Grundlagen der Funkübertragung von Zeitsignalen und des offenen Schwingkreises erlernt werden. Das Zeitsignal für Funkuhren wird vom Zeitzeichensender DCF77 in der Nähe von Frankfurt am Main auf einer Frequenz von 77,5kHz ausgesendet (Richtberg, 2018). Durch

⁶ Detaillierte Anleitung beispielsweise unter: https://projecthub.arduino.cc/Aboubakr_Elhammoumi/77347b69-2ade-4a44-b724-3bb91e954188

eine Absenkung der Sendeleistung können dadurch 59Bits pro Minute zur Datums- und Zeit-Synchronisation gesendet werden (Richtberg, 2018). Aber wie kann dieses Signal empfangen und ausgelesen werden? Genau das kann mit einem Mikrocontroller und dem passenden Empfangsmodul gezeigt werden. Das hierfür benötigte Empfangsmodul besteht meist aus einer mit einer Spule umwickelten Ferritantenne und einem parallel dazu geschalteten Kondensator. Die Spule und der Kondensator bilden einen offenen Schwingkreis, der bei gekauften Modulen auf die Sendefrequenz des Zeitsignals abgestimmt ist (Richtberg, 2018). In einem ersten Versuch kann die Abschwächung des Zeitsignals durch eine LED visualisiert werden, bevor der Arduino in einem zweiten Versuch zur wirklichen Decodierung des Signals verwendet wird. Lernende können durch diese Versuche einiges zur Informationsübertragung über Funk in einem praxisnahen Kontext lernen. Auch zu diesem Versuch finden sich konkrete Anleitungen im Heft „Arduino, Raspberry Pi & Co“.

Neben den vorgestellten Versuchen sind auch Experimente zur Wärmeabsorption, der Messung der Schallgeschwindigkeit oder verschiedenen Interferenzphänomenen denkbar. Anleitungen hierzu und weitere Ideen finden sich beispielsweise in den anfangs genannten Werken.

4.2 Moderne Physik

Neben den Einsatzgebieten in der klassischen Physik findet man auch in der modernen Physik einige Experimentideen. Aufgrund der meist eingeschränkten Verfügbarkeit von kommerziellen Lehrmitteln zur modernen Physik an Schulen (Bernstein et al., 2021) wurden im Schülerlabor S’Cool Lab des CERNs Experimente entwickelt, die Methoden der modernen Experimentalphysik widerspiegeln (Bernstein et al., 2021). Um den Preis der benötigten Teile möglichst gering zu halten, werden in diesen Experimenten Mikrocontroller-Komponenten verwendet und viele Bestandteile 3D-gedruckt.

Das erste Beispiel ist ein Mikrocontroller basiertes Analogieexperiment zur Detektion von Gravitationswellen. Es soll Lernenden die prinzipielle Funktionsweise von Gravitationswellendetektion über das Modell eines Michelson-Interferometers verdeutlichen (Bernstein et al., 2021). Der Aufbau des Versuches ist in Abbildung 15 auf der nächsten Seite dargestellt. Als Lichtquelle dient im Versuchsaufbau ein Laserpointer und alle Halterungen sind 3D-gedruckt. Die quantitative Auswertung des Experiments erfolgt über ein Fotodioden-Modul. Ein Piezoelement, das über den Mikrocontroller angesteuert wird, ermöglicht die Bewegung



Abbildung 15: Aufbau eines Analogieexperiments zur Gravitationswellendetektion des S'Cool Labs Am CERN

Bildquelle:
<https://scoollab.web.cern.ch/laserlab3> (Abgerufen am 8.3.2023)

eines Spiegels auf einer Größenordnung von einigen hundert Nanometern (Bernstein et al., 2021). Die Bewegung des Spiegels bewirkt eine Änderung der Weglängendifferenz und so eine Änderung des Interferenzmusters. Diese ist über die Intensitätswerte an der Fotodiode detektierbar (Bernstein et al., 2021). Weitere Informationen, sowie detaillierte Aufbaupläne und fertige Dateien für den 3D-Druck finden sich auf der Homepage des S'Cool Labs unter <https://scoollab.web.cern.ch>.

Neben diesem Experiment wurden das Modell eines Wechselspannung-Linearbeschleunigers (LINAC) und Experimente zum äußeren Fotoeffekt konzipiert. Das Prinzip des LINAC-Modells besteht darin, dass ein graphitbeschichteter Tischtennisball über elektrostatische Felder in einer Plexiglasröhre beschleunigt wird (Bernstein et al., 2021). Wie man in Abbildung 16 sehen kann, ist die Verkabelung des Versuchsaufbaus wesentlich komplexer als beim ersten Experiment. Zur Vereinfachung des Aufbaus und der Reduzierung der Fehleranfälligkeit, gibt es deshalb für dieses und das folgende Experiment fertige Shields zu kaufen, die auf einen Arduino UNO aufgesetzt werden können und alle erforderlichen Komponenten enthalten. Das Shield zum äußeren Fotoeffekt beinhaltet unter anderem ein

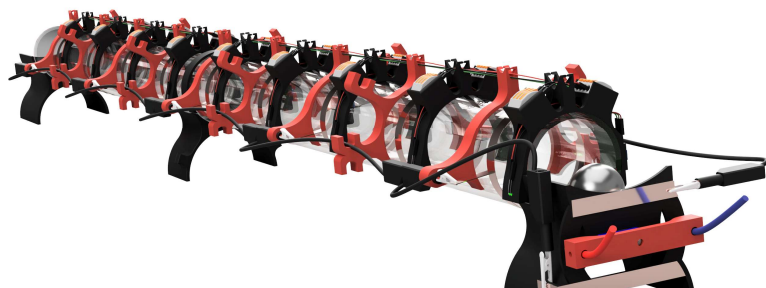


Abbildung 16: Versuchsaufbau des LINAC zum Beschleunigen eines graphitbeschichteten Tischtennisballs.

Bildquelle: Homepage S'Cool Lab (<https://scoollab.web.cern.ch/linac3D>) (Abgerufen am 8.3.2023)

Nanoamperemeter, sodass Messungen zur Intensitätsunabhängigkeit und Frequenzabhängigkeit der angelegten Gegenspannung durchgeführt werden können. Durch Auftragung der maximalen Elektronen-Energie über der Lichtfrequenz lässt sich mit diesem Aufbau auch das Planck'sche Wirkungsquantum bestimmen (Bernstein et al., 2021). Der große Vorteil des 3D-gedruckten Experiments mit der Mikrocontroller Messwerterfassung liegt erneut im Preis, da dieser laut den Entwicklern um mehr als eine Größenordnung reduziert wurde (Bernstein et al., 2021), sodass es jetzt auch als Schülerexperiment durchgeführt werden kann. Auch zu den Versuchen zum äußeren Fotoeffekt und dem LINAC gibt es detaillierte Baupläne und weitere Informationen auf der Homepage des S'Cool LAB.

4.3 Fachübergreifende Projektideen

Das volle Potenzial der Mikrocontroller kann, wie in Kapitel drei beschrieben, nur ausgeschöpft werden, wenn es vielfältige Anwendungsmöglichkeiten in mehreren Schulfächern gibt. Nur so steht der Mehrwert des Einsatzes in sinnvollem Verhältnis zu der zu investierenden Zeit. Im Folgenden werden Projektideen vorgestellt, die Zugänge und Anschlussmöglichkeiten in mehreren Schulfächern zulassen.

Ein schon mehrfach realisiertes Projekt ist das Starten eines Stratosphären-Ballons. Über eine Mikrocontroller-Messvorrichtung können beispielweise Höhe, Luftfeuchtigkeit, Temperatur oder die GPS-Koordinaten gemessen und gespeichert werden. Mögliche Umsetzungen dieses Projektes finden sich im „StratoSat“ Schülerwettbewerb mit dem Beispiel des „StratoIno“-Datenloggers. Eine detaillierte Anleitung zu diesem Datenlogger findet sich auf der Website der LMU München, wobei der Datenlogger für die eigene Mission nach individuellen Wünschen und Anforderungen angepasst werden kann. Für dieses Projekt sind Zugänge im Physik-, Informatik oder dem Geographieunterricht denkbar.

Ein weiteres Projekt ist der Aufbau einer „Schul-Wetterstation“. Diese kann neben der Temperatur und der Luftfeuchtigkeit auch die Windstärke oder den CO₂-Gehalt der Luft messen. Online finden sich des Weiteren einige Anleitung zu einem „Do-it-yourself“ Niederschlagsmesser⁷, sodass die Wetterstation ein vollständiges Klimadiagramm im Laufe des Schuljahres aufzeichnen könnte. Auch dieses Projekt bietet mehrere Anschlussmöglichkeiten,

⁷ Eine Anleitung findet man beispielsweise unter <https://tutorials-raspberrypi.de/esp8266-regensensor-selber-bauen-wie-viel-pro-qm/> (Stand: 10.2.2023)

wie die Auswertung des Klimadiagramms im Geographieunterricht oder die Teilnahme an verschiedenen Citizen-Science-Projekten. Stattet man die Station zusätzlich mit einem Helligkeitssensor aus, kann auch das Thema „Lichtverschmutzung“ im Fach Astrophysik thematisiert werden.

Neben diesen beiden Projektideen, sind noch viele weitere denkbar. Beispiele sind unter anderem das Bauen einer Lärm- oder CO₂-Ampel für das Klassenzimmer, das Aufzeichnen verschiedenster Bewegung im Sport- und Physikunterricht oder das Aufzeichnen eines EKGs im Biologieunterricht. Hierzu gibt es ein spezielles EKG-Modul AD8232, das im Rahmen dieser Arbeit allerdings nicht genauer getestet wurde. Dessen Funktionalität kann daher nicht bewertet werden. Genauere Erläuterungen zu diesem Modul findet man unter: <https://learn.sparkfun.com/tutorials/ad8232-heart-rate-monitor-hookup-guide>.

Mit dem „Gläsernen Ökosystem“ wird im nächsten Kapitel ein weiteres Projekt vorgestellt, dass fächerübergreifend umsetzbar ist.

5 Projekt: Das gläserne Ökosystem

Die Vorgänger heutiger Ökosysteme im Glas sind die so genannten „Ward’schen Kästen“, die bereits Mitte des 19. Jahrhunderts zu verschiedenen Zwecken eingesetzt wurden. Der Botaniker Dr. Nathaniel Ward entdeckte im Jahr 1829 durch Zufall das Prinzip der nach ihm benannten Kästen. Er beobachtete, dass unter einer luftdicht verschlossenen Glasglocke Pflanzen keimen und wachsen können (Bäck, 2017). Da sich innerhalb der geschlossenen Gefäße autarke Stoffkreisläufe einstellen, können die Pflanzen so im Idealfall über Monate und Jahre ohne jegliche Pflege, also insbesondere ohne Gießen, überleben.

Aufbauend auf dem Modell des versiegelten Ökosystems im Glas wurde im Rahmen des Labs4Future an der Universität Würzburg das fächerübergreifende Schülerlabor „Das gläserne Ökosystem“ entworfen, das neben fachlichen Inhalten auch überfachliche Arbeitsweisen vermitteln soll (Grothaus et al., 2023). Hierzu wird ein verschlossenes Ökosystem im Glas mit Sensorik versehen, sodass verschiedene Umweltparameter über einen längeren Zeitraum beobachtet werden können. Zum Zeitpunkt der Abgabe dieser Arbeit sind viele technische Komponenten bereits realisiert, das Projekt selbst wurde allerdings noch nicht mit SchülerInnen

durchgeführt. Im Laufe dieses Kapitels werden die Lernziele sowie der angedachte Ablauf des Schülerlabors vorgestellt. Darüber hinaus wird anhand des Projektes die exemplarische Anbindung eines Mikrocontroller-Messsystems an tet.folio, wie in Kapitel 3.4.2 beschrieben, demonstriert.

5.1 Die Lernziele

Das vorgestellte Schülerlabor soll zukünftig in das Labs4Future der Universität Würzburg integriert werden. Das zentrale Ziel des Labs4Future besteht in der Minimierung des so genannten Knowledge-Action-Gaps im Bereich des Klimawandels (Grothaus et al., 2023). Dieser Begriff beschreibt die oft gravierenden Unterschiede zwischen dem Wissen über den Klimawandel und dem daraus resultierenden Handeln. Das Schülerlabor ist ausgelegt auf die neunten und zehnten Klassen der Realschulen und Gymnasien in Bayern.

Die Lernziele des Schülerlabors gliedern sich in zwei Bereiche: Die überfachlichen Arbeitsweisen und die fachbezogenen Inhalte (Grothaus et al., 2023). Im Folgenden sollen ausgewählte Lernziele der beiden Bereiche vorgestellt werden. Hierzu werden zuerst die fachübergreifenden Arbeitsmethoden, anschließend die fachlichen Inhalte betrachtet.

Die Lernenden sollen im Schülerlabor möglichst selbstständig an eigenen Forscherfragen arbeiten. Hierzu müssen sie die wesentlichen Schritte des naturwissenschaftlichen Experimentierens durchlaufen: Sie stellen während des Projektes eigene Forscherfragen auf, formulieren Hypothesen zu diesen Fragen und entwerfen anschließend geeignete Experimente, um diese Hypothesen experimentell zu testen. Die Experimente sollen selbstständig durchgeführt, ausgewertet und kritisch hinterfragt werden (Grothaus et al., 2023). Die Stufen dieses Prozesses sind zentrale Bestandteile experimenteller Erkenntnisgewinnung und stellen daher wichtige Lernziele des Projektes dar. Insbesondere das selbstständige, offene Experimentieren wird in den Schulen oft vernachlässigt (vgl. Kapitel 3.3).

Weitere Arbeitsweisen, die die SchülerInnen im Projekt kennenlernen sollen, stammen aus dem Bereich der „Data Literacy“. Data Literacy umfasst unter anderem das Erfassen, Aufbereiten, Visualisieren und Analysieren von Daten (Grothaus et al., 2023). Im Projekt bieten sich vielfältige Möglichkeiten, Kompetenzen dieses Bereiches zu erlernen. Beispiele hierfür sind die mathematische Weiterverarbeitung der Rohdaten unter Berücksichtigung potenzieller Messfehler oder das Aufbereiten und Visualisieren von Messreihen zur späteren biophysikalischen Argumentation.

Neben den Arbeitsweisen sollen den Lernenden im Projekt auch fachbezogene Inhalte vermittelt werden (Grothaus et al., 2023). Diese können aus verschiedenen Naturwissenschaften wie der Informatik, Physik oder Biologie stammen, wobei eine genaue Zuordnung der Lernziele zu einer einzelnen Naturwissenschaft oft nicht möglich ist. Aus Sicht der Informatik kann im Schülerlabor beispielsweise das Konzipieren eines so genannten „Physical-Computing-Systems“ behandelt werden, das eine Verbindung zwischen analoger Welt und der Computerwelt ermöglicht (Grothaus et al., 2023). Hierzu müssen SchülerInnen mit verschiedenen Informatiksystemen reflektiert umgehen können, um eine begründete Auswahl eines geeigneten Systems für ihr Projekt treffen zu können (Grothaus et al., 2023). Neben diesem Inhalt lassen sich auch verschiedene Aspekte aus der Physik thematisieren: Hierzu zählen beispielsweise messtheoretische Grundlagen (vgl. Kapitel 1.2.1), die genaue Funktionsweise der einzelnen Sensoren oder verschiedene biophysikalischen Inhalte wie der Ablauf des Kohlenstoffkreislaufes. Erst diese Kenntnisse ermöglichen die Interpretation der Messdaten und das Ziehen wissenschaftlicher Schlüsse. Durch das Diskutieren von Stärken und Schwächen des Modells eines gläsernen Ökosystems und dessen Parallelen zur realen Welt, können weitere biologische oder geografische Inhalte in das Projekt integriert werden.

Die Vielfältigkeit der Lernziele insbesondere im Hinblick auf die zu vermittelnden Fachinhalte lassen eine individuelle Schwerpunktsetzung im Schülerlabor zu, sodass es bei der Durchführung möglich ist, auf Interessen der Lernenden differenziert einzugehen.

5.2 Ablauf des Schülerlabors

In der praktischen Durchführung soll das Schülerlabor in fünf aufeinander aufbauende Phasen unterteilt werden (Grothaus et al., 2023). Die Phasen werden nachfolgend anhand einer Realisierung des gläsernen Ökosystems genauer erläutert.

5.2.1 Erste Phase: Vorüberlegungen mit biologischem Fokus

Das komplexe Erdsystem wird durch eine Vielzahl an Systemgrößen und Umweltparametern sowie deren gegenseitigen Wechselwirkungen bestimmt. Durch verschiedenste Messgrößen wie dem Temperaturgradienten in der Atmosphäre, der Feinstaubbelastung oder dem Luftdruck lassen sich in diesem komplexen System kurzfristig Wetterprognosen und langfristig Entwicklungen des Weltklimas berechnen (Grothaus et al., 2023). In der ersten Phase des Projektes soll das gläserne Ökosystem als Elementarisierung des Erdsystems vorgestellt

werden. Dabei soll auch auf die hierfür nötigen Vereinfachungen und daraus resultierenden Grenzen des Modells eingegangen werden. In dieser Phase werden auch die biologischen Vorüberlegungen für die nächsten Phasen getroffen. Hierzu gehören unter anderem die Bestimmung relevanter Umweltparameter oder die Festlegung der Bepflanzung des Glases (Grothaus et al., 2023). Im Folgenden werden biologische Grundlagen des versiegelten Ökosystems und dessen Parallelen zu realen Ökosystemen behandelt. Anschließend wird genauer auf die Pflanzen und den Boden der gläsernen Ökosysteme eingegangen. Die verschiedenen Themen ermöglichen eine individuelle Schwerpunktsetzung durch die Lernenden in der Umsetzung dieser Phase.

Für das lange Bestehen der verschlossenen Ökosysteme im Glas sind im Wesentlichen zwei Stoffkreisläufe verantwortlich: Der Kohlenstoff- bzw. Sauerstoffkreislauf, der sich durch die Photosynthese, Zellatmung und verschiedene Zersetzungsprozesse einstellt, und der Wasserkreislauf (Bäck, 2017). Aufgrund der zum Erhalt der Kreisläufe stetig nötigen Energiezufuhr durch die Sonne, können die beiden Kreisläufe allerdings nicht als geschlossen betrachtet werden (Krohn, 2010). In einem realen Ökosystem werden drei Grundtypen des Lebens unterschieden: Die Photosynthese betreibenden Produzenten, die Destruenten, wie Boden- und Mikroorganismen, und die Konsumenten (Bäck, 2017). Da bei den Zersetzungsprozessen der Destruenten Kohlenstoff frei wird, der für die Photosynthese der Produzenten benötigt wird, sind die beiden Grundtypen essenziell für den Erhalt eines versiegelten Ökosystems. Allerdings sind in einem Ökosystem im Glas meistens ausschließlich Produzenten und Destruenten vertreten, was auf reale Ökosysteme nur bedingt zutrifft. In der Realität gibt es stets Konsumenten, die ein Ökosystem beeinflussen, weshalb sich reale Ökosysteme nicht wie im Glas durch ein statisches Gleichgewicht erhalten, sondern durch eine Dynamik, die kleinere Störungen aushalten kann (Krohn, 2010). Des Weiteren sind Ökosysteme in der Praxis prinzipiell nicht geschlossen, weshalb sie nicht separiert betrachtet werden können (Krohn, 2010). Das gläserne Ökosystem wird hingegen oft als abgeschlossen betrachtet. Trotz der Schwächen des Modells kann beispielsweise die Rolle des Kohlenstoffkreislaufes im globalen Klimawandel anhand des Modells gut dargestellt werden.

Vor dem eigenen Aufbau eines gläsernen Ökosystems sollten vor allem die Pflanzen und der Boden genauer betrachtet werden. Für ein Ökosystem im Glas kann prinzipiell jede Pflanzenart genutzt werden. Je nach Zielsetzung und der angepeilten Lebensdauer des Ökosystems eignen sich allerdings unterschiedliche Arten. Da viele Pflanzen die dauerhafte Luftfeuchtigkeit von

nahezu 100% nicht tolerieren, empfiehlt der Biologe Ulf Soltau für langfristig angelegte Gläser die Verwendung von kleinbleibenden Pflanzenarten tropischer Herkunft. Auf seiner Website führt er eine Liste von Pflanzenarten, die sich für dauerhafte Ökosysteme eignen (Soltau, o. D.). Die Liste findet sich unter: <https://ulfsoltau.wordpress.com/pflanzen/>.

Die gläsernen Ökosysteme im Projekt sind aber darauf ausgelegt, schnell auf Änderungen der Umweltparameter zu reagieren und die Auswirkungen des Pflanzenwachstums sichtbar zu machen. Deshalb werden hierfür schnellwachsende Pflanzenarten verwendet, die daher aber nicht für ein langfristiges Bestehen des Ökosystems geeignet sind. Beispiele hierfür sind Kresse- oder Radieschensamen. Die gesäte Kresse keimt im Glas innerhalb von einem Tag und erreicht schon nach etwa fünf Tagen ihre maximale Höhe (vgl. Abbildung 17). So können schon in kurzer Zeit verschiedene Pflanzenstadien untersucht werden.

Neben den Pflanzen sollte auch der Boden des Ökosystems verschiedene Anforderungen erfüllen. Er sollte chemisch neutral, nährstoffarm und grobkörnig sein, um den Gasaustausch zu gewährleisten. Des Weiteren sollten für dauerhafte Ökosysteme keine organischen Materialien wie herkömmliche Blumenerde verwendet werden, da sich diese im Laufe der Zeit verdichten (Soltau, o. D.). Da im Projekt Pflanzensamen verwendet werden, wird zumindest eine kleine Schicht Blumenerde benötigt, die das Keimen der Samen sicherstellt. Im dargestellten Glas in Abbildung 17 wurde deshalb etwas Blumenerde auf einer Schicht des Pflanzensubstrates Blähton verwendet. Das Pflanzensubstrat speichert Wasser besser als die Erde und beugt so der Bildung von Schimmel im Ökosystem vor.



Abbildung 17: Mit Kressesamen bepflanzt Glas an verschiedenen Tagen.
Bildquelle: Eigene Aufnahme

Ein speziell auf die Schule ausgelegtes Ökosystemkonzept findet sich bei Krohn (2010) auf Seite 40. Zwei Ökosysteme sollen hier verschiedene Klimazonen nachahmen: Das erste Glas stellt die feuchten Tropen dar, das zweite Glas die trockenen Tropen. Durch die verschiedenen Klimazonen treten andere Bedingungen für den Wasser- und Kohlenstoffkreislauf auf, sodass SchülerInnen völlig unterschiedliche Ökosysteme miteinander vergleichen können.

5.2.2 Zweite Phase: Physical Computing

Aufbauend auf den Überlegungen der ersten Phase soll in der zweiten Phase das Messwerterfassungssystem im Fokus stehen. Die SchülerInnen machen sich hierzu Gedanken zur Erfassung verschiedener Umweltparameter und den Anforderungen an die im Glas verbauten Sensoren. Exemplarisch sollen während dieser Phase einige Sensoren an den verwendeten Mikrocontroller angeschlossen werden (Grothaus et al., 2023). Für die im Laufe des Projekts folgenden Messungen soll aufgrund der deutlichen Zeitersparnis und der Vergleichbarkeit der Messdaten allerdings auf von der Uni gestellte Gläser zurückgegriffen werden. Eine weitere Herausforderung des Physical Computings besteht in der Modellierung eines Deckels für das Ökosystem im Glas. Dieser sollte das Glas möglichst luftdicht verschließen, die Durchführung der Sensorkabel zulassen und auf weitere Anforderungen des Projektes angepasst sein (Grothaus et al., 2023).

Neben den Überlegungen zur Umsetzung der Messwerterfassung auf Hardwareseite, müssen sich Lernende aber auch Gedanken zur Softwareseite des Projektes machen. Hierbei ist insbesondere die Realisierung der Datenübertragung, Datenspeicherung und die Weiterverarbeitung der Daten relevant (Grothaus et al., 2023).

Die zweite Phase des Projektes lässt sich je nach Vorwissen der Lernenden und der zur Verfügung stehenden Zeit höchst flexibel gestalten. Die verschiedenen Aspekte einer möglichen Umsetzung des Physical Computings werden im Folgenden vorgestellt.

Die erste Herausforderung, um Sensoren in das Ökosystem zu integrieren, besteht darin die Sensorkabel aus dem Inneren des Glases nach außen zu führen und das Gefäß dennoch luftdicht zu verschließen. Die zugehörigen Gefäßdeckel lassen oft keine Kabeldurchführungen zu. Eine Lösung kann ein 3D gedruckter Deckel liefern. Ein eigener Deckel bietet die Möglichkeit neben den Kabeldurchführung auch Aus-

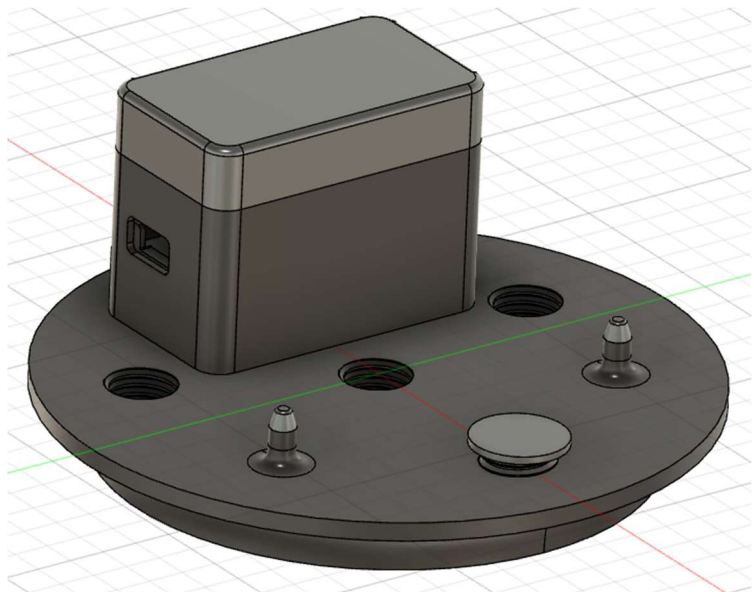


Abbildung 18: CAD-Modell des im Projekt verwendeten Deckels. Neben der Messbox bietet er Schlauchanschlüsse zum Austauschen der Luft im Ökosystem und Löcher zum Nachgießen.
Bildquelle: Eigenes CAD-Modell

sparungen für die Befestigung der Sensoren sowie Löcher zum Nachgießen und dem Gasaustausch zu integrieren. Eine mögliche Umsetzung des Deckels ist in Abbildung 18 dargestellt. Neben der Messbox, die im nächsten Abschnitt genauer thematisiert wird, sieht man zwei Gasanschlüsse, über die die Luft im Ökosystem ausgetauscht werden kann. Vier weitere Löcher lassen das Nachgießen an verschiedenen Stellen zu. Über angepasste Gewinde sollen diese möglichst gut verschlossen werden können.

Für die hier vorgestellte Realisierung des gläsernen Ökosystems wurden folgende Umweltparameter und Sensoren gewählt:

1. Lufttemperatur (DS18B20)
2. Luftfeuchtigkeit (BME280)
3. CO₂-Gehalt in der Luft (SCD30)
4. Bodentemperatur (SMT50)
5. Bodenfeuchtigkeit (SMT50)
6. Helligkeit (BH1750)

Neben diesen Parametern wäre auch eine Messung des Sauerstoffgehaltes in der Luft interessant, um das Entstehen von Sauerstoff bei der Photosynthese nachzuweisen. Leider ist die Änderung der Sauerstoffkonzentration zu klein, um sie mit den gängigen Sauerstoffsensoren messen zu können. Eine weitere Herausforderung an die verwendeten Sensoren besteht in der hohen Luftfeuchtigkeit. Die meisten Sensoren sind auf diese nicht ausgelegt und können daher beschädigt werden. Sie wurden deshalb, bis auf die für die Messung benötigten Öffnungen, mit einer Schicht Urethan Schutzlack besprüht. Langfristige Schäden aufgrund der hohen Luftfeuchtigkeit können dennoch nicht ausgeschlossen werden, weshalb die Kalibrierung der Sensoren ab und zu überprüft werden sollte.

Neben den oben genannten Sensoren werden ein Mikrocontroller und ein externer A/D-Wandler für die Messwerterfassung benötigt. Bei der Verwendung eines Mikrocontrollers mit mehreren analogen Input-Pins, kann auf den externen A/D-Wandler verzichtet werden. In dieser Umsetzung wurden die ESP8266-Version des D1mini als Mikrocontroller und der I²C-fähige externe A/D-Wandler ADS1115 verwendet. Die Verkabelung aller Komponenten erfolgt in der auf der Deckeloberseite integrierten Box (vgl. Abbildung 18), sodass keine unnötigen Kabel aus dieser herausgeführt werden müssen. Bis auf den Temperatursensor DS18B20 und

den analogen Sensor SMT50 im Boden kommunizieren alle Komponenten mit dem I²C-Übertragungsprotokoll. Eine gelötete Platine mit vier verbundenen Leitungen reicht deshalb für die Verkabelung des Projektes aus. Den Aufbau der Platine mit DuPont-Steckern sieht man in Abbildung 19. Die im Bild gekennzeichneten Leitungen sind an der

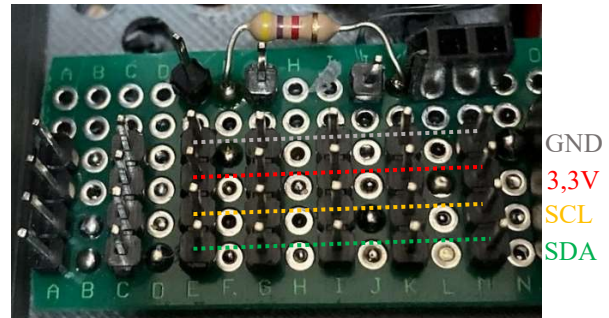


Abbildung 19: Für die Verkabelung der Sensoren verwendete Platine. Je zwei Leitungen werden zur Stromversorgung und Datenübertragung verwendet.

Bildquelle: Eigene Aufnahme

Unterseite der Platine verbunden. Über Jumper-Kabel können die Komponenten direkt an die Platine angeschlossen werden. Der Widerstand, den man am oberen Bildrand in Abbildung 19 sieht, wird für den Anschluss des Temperatursensors DS18B20 benötigt. Dieser Sensor ist nicht mit dem I²C-Protokoll kompatibel, weshalb er nicht wie die anderen Komponenten angeschlossen werden kann. Je nach Version des Sensors muss für das daher bei diesem Sensor verwendete „One-Wire-Protokoll“ ein geeigneter Widerstand verbaut werden.

Ein weiterer Aspekt des Physical Computings besteht in der Versiegelung des Deckels. In der vorgestellten Umsetzung geschieht dies mit Knetmasse. Für langfristig angelegte Ökosysteme sollte der Deckel hingegen mit Silikon abgedichtet werden. Den Aufbau des Ökosystems und die Befestigung der Sensoren im Deckel sieht man in den folgenden Abbildungen.



Abbildung 20: Realisierung des gläsernen Ökosystems mit Messwerterfassung.

Bildquelle: Eigene Aufnahme

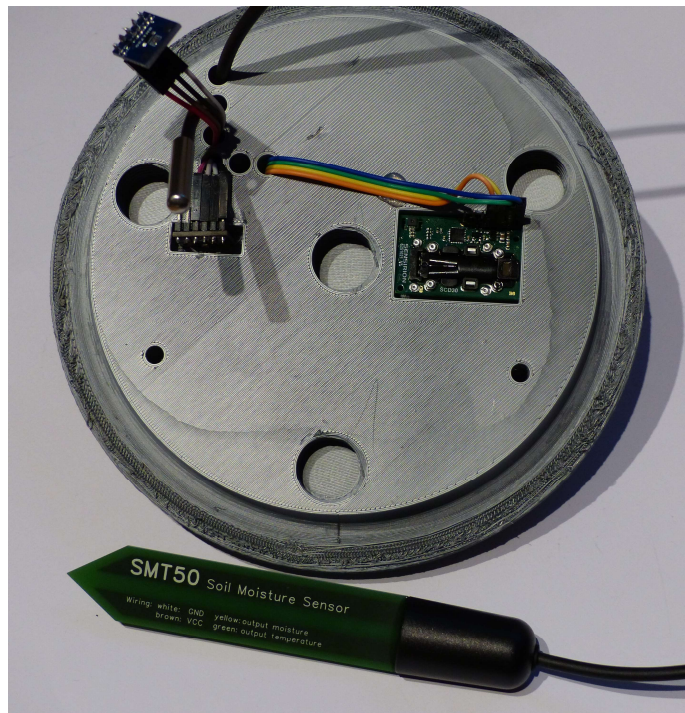


Abbildung 21: Deckelunterseite mit Befestigung der Sensoren sowie verschlossenen Löchern zum Gießen und Gasaustausch.

Bildquelle: Eigene Aufnahme

Nachdem die physische Verbindung der Sensoren mit dem Mikrocontroller behandelt wurde, soll der Blick nun auf eine mögliche Umsetzung der Software gelegt werden. Für die Datenspeicherung in der vorgestellten Realisierung wurde die kostenlose Version des online IoT-Service AdafruitIO gewählt (siehe Kapitel 3.4.2). Der WiFi-fähige Mikrocontroller ESP8266 schickt die Messdaten automatisch an AdafruitIO, wo jede Messvariable in einem eigenen Feed gespeichert wird. Hierzu wurde eine Rate von einer Messung alle fünf Minuten gewählt. In ihrem Feed werden die Messdaten 30 Tage online gespeichert und können dort jederzeit eingesehen und heruntergeladen werden (Adafruit, o. D.). Um den Verlauf der Umweltparameter sowie den aktuellen Zustand des Ökosystems auf einen Blick sehen zu können, wurde ein so genanntes Dashboard angelegt, das Messverläufe und alle aktuellen Messdaten anzeigt. Die Dashboard-Lösung im Projekt ist in nachfolgender Abbildung dargestellt. Links werden die aktuellen Messwerte angezeigt, rechts werden ausgewählte Messgraphen dargestellt.

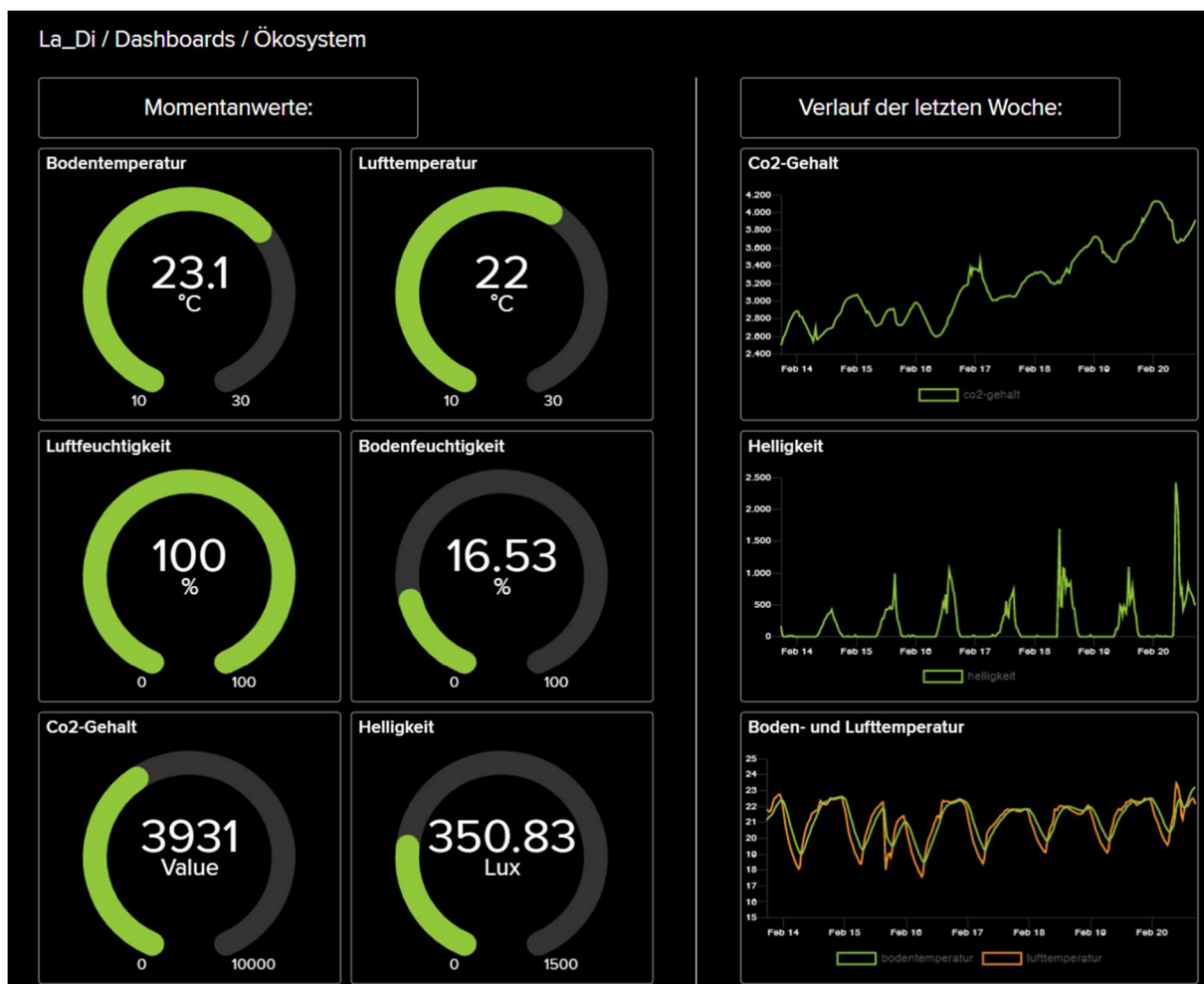


Abbildung 20: Beispiel eines möglichen AdafruitIO Dashboards zur Überwachung verschiedener Umweltparameter. Die verwendeten Bausteine können individuell angepasst werden.

5.2.3 Dritte Phase: Datenauswertungen und -präsentation

In der dritten Phase des Projektes geht es um den Erwerb grundlegender Data Literacy Kompetenzen. Neben der Aufnahme eigener Messdaten soll hierzu eine Datenbank mit Rohdaten alter Ökosysteme zur Verfügung stehen (Grothaus et al., 2023). Diese können zum Erlernen verschiedener Methoden der Datenanalyse oder dem Vergleich mit den eigenen Messdaten verwendet werden. Wichtige Bestandteile dieser Phase sind auch das Bereinigen fehlerhaft übertragener Messwerte, das kritische Hinterfragen der Messergebnisse sowie das Auswählen und Visualisieren geeigneter Messwerte zu einer bestimmten Hypothese (Grothaus et al., 2023).

Im Folgenden sollen ausgewählte Messgraphen eines mit Kresse bepflanzten Ökosystems mögliche Ergebnisse dieser Phase darstellen. Die dazu verwendeten Daten sind über einen Zeitraum von etwa einem Monat aufgezeichnet und online gespeichert worden. Hierzu wurde eine Messrate von einer Messung alle fünf Minuten gewählt. Innerhalb des Messzeitraums sind die Kressesamen gekeimt, die Kresse gewachsen und wieder eingegangen. Es wurde also ein gesamter „Lebenszyklus“ des Kresse-Ökosystems aufgezeichnet. Da der Fokus dieser Arbeit allerdings auf der Anwendung der Mikrocontroller liegt, wird auf eine tiefergreifende biophysikalische Analyse verzichtet.

In Abbildung 21 sieht man den Verlauf des CO₂-Gehaltes in der Luft über einen Zeitraum von

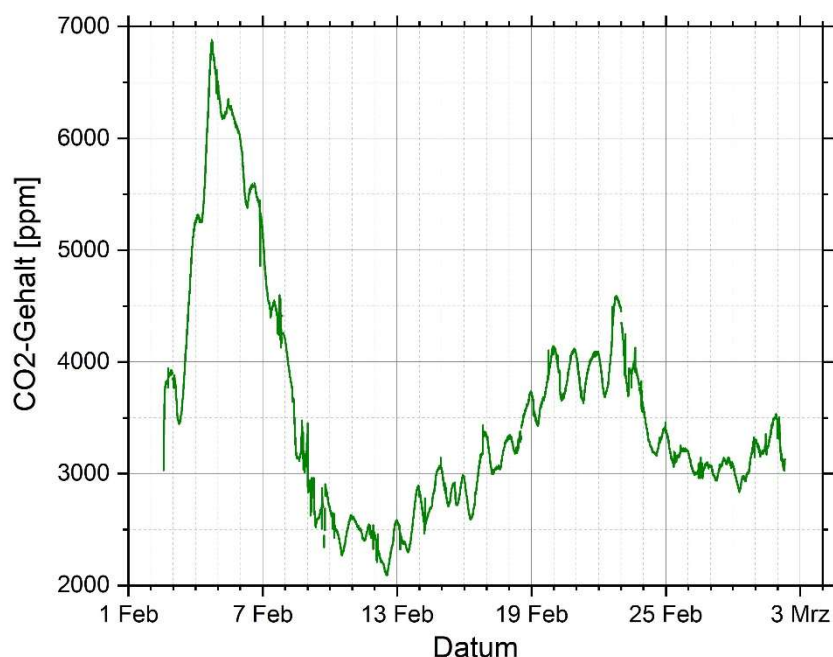


Abbildung 21: Verlauf des CO₂-Gehaltes in der Luft im versiegelten Ökosystem über einen Zeitraum von etwa einem Monat.

etwa einem Monat. Neben den täglichen Schwankungen durch die Photosynthese kann man in den ersten Tagen einen starken CO₂-Anstieg feststellen. Dieser lässt sich erklären, da die Kresse in diesem Zeitraum noch nicht gewachsen ist und daher keine Photosynthese betreibt. Sobald die Kressesamen gekeimt und die ersten Pflanzen gewachsen

sind, wird der CO₂-Überschuss wieder abgebaut. Der vorliegende Messverlauf lässt die Vermutung zu, dass der CO₂-Gehalt überkompensiert wurde und deshalb wieder angestiegen ist. Bei einem längerfristigen System scheint es möglich, dass sich der CO₂-Gehalt irgendwann auf einem, bis auf die Tag-Nacht-Schwankungen, konstanten Niveau einpendelt. Bei der Auswertung der CO₂-Daten ist es besonders wichtig unphysikalische Messdaten auszusortieren. Diese können durch eine fehlerhafte Messung oder einen Fehler bei der Datenübertragung zustande kommen.

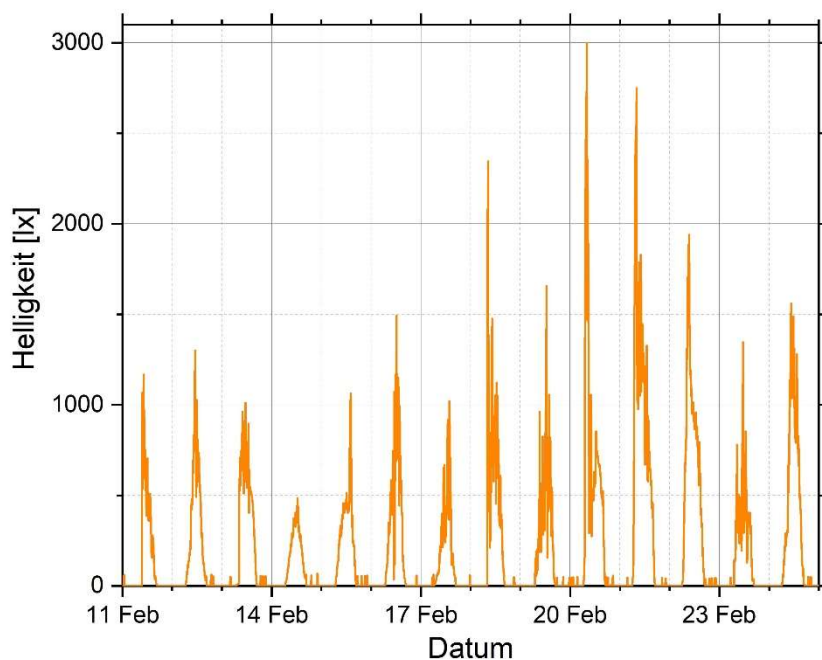


Abbildung 22: Gemessener Helligkeitsverlauf über einen Zeitraum von zwei Wochen

Die nebenstehende Abbildung veranschaulicht die Daten des Helligkeitssensors über einen Zeitraum von zwei Wochen. Bis auf kleinere Schwankungen misst der Sensor nachts eine Helligkeit von 0 Lux. Im Laufe des Tages erreicht sie ihr Maximum. Aufgrund verschiedener Wetterverhältnisse tritt das Helligkeitsmaximum im

Messzeitraum zu verschiedenen Uhrzeiten auf. Die verschiedenen Höhen der Maxima können die SchülerInnen im Projekt mit echten Wetterdaten verifizieren: An einem bewölkten Tag tritt ein niedrigeres Helligkeitsmaximum auf als an einem sonnigen Tag.

Im nächsten Diagramm soll ein genauerer Blick auf die Boden- und Lufttemperatur gelegt werden. Der in Abbildung 23 zu sehende Verlauf wird wesentlich durch die Zeitschaltung der Raumheizung beeinflusst. Nachts wird der Raum, in dem sich das Ökosystem befindet, nicht geheizt, weshalb die Temperatur im Glas abfällt. Hierbei kann man anhand der Messgraphen belegen, dass der Boden aufgrund unterschiedlicher Wärmekapazitäten langsamer abkühlt als die Luft. Beim Vergleich der Daten zweier verschiedener Temperatursensoren kann das Problem eines Offset-Fehlers thematisiert werden. Dieser Fehlertyp tritt auf, wenn die beiden Sensoren nicht identisch kalibriert sind, das heißt bei einer identischen Umgebungstemperatur zeigen die beiden Sensoren unterschiedliche Werte an. Um die Werte der beiden Sensoren

allerdings vergleichen zu können, muss deren Offset in der Auswertung gegebenenfalls beachtet werden.

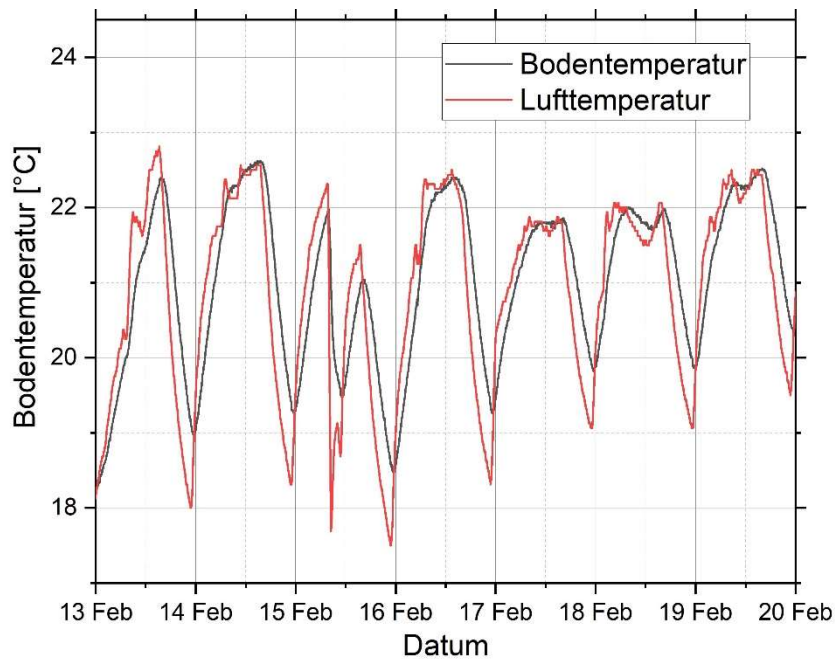


Abbildung 23: Vergleich der gemessenen Boden- und Lufttemperaturen über einen Zeitraum von einer Woche. Die Temperaturschwankungen kommen im Wesentlichen von der Zeitschaltung der Heizung, da der Raum des Ökosystems nachts nicht geheizt wird.

Die folgenden beiden Messgraphen sollen den Verlauf des CO₂-Gehaltes aus Abbildung 21 genauer erläutern:

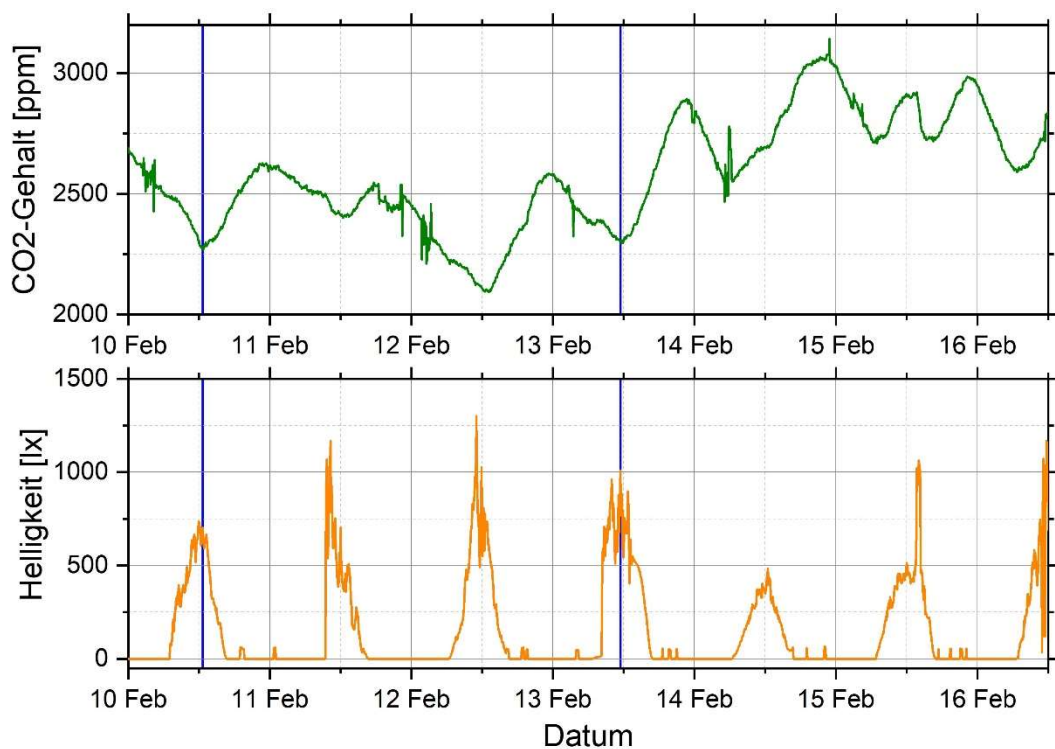


Abbildung 24: Gegenüberstellung der Verläufe des CO₂-Gehaltes und der Helligkeit. Ein Helligkeitsmaximum fällt hierbei stets mit einem Minimum des CO₂-Gehaltes zusammen (vgl. blaue Markierungen)

Abbildung 24 zeigt eine Gegenüberstellung des Helligkeits- und CO₂-Verlaufes. Anhand der blauen Markierungen erkennt man, dass ein Minimum des CO₂-Gehaltes auftritt, wenn die Helligkeit ihr Maximum erreicht. Dies zeigt, dass der Abfall des CO₂-Gehaltes über die Helligkeit des Lichtes tatsächlich mit der Photosynthese der Kresse zusammenhängt. Betrachtet man allerdings den Verlauf zu Beginn des 11. Februars, fällt auf, dass der CO₂-Gehalt sinkt, obwohl eine Helligkeit von etwa 0 Lux gemessen wird. Da dieser Abfall nicht über die Photosynthese der Pflanzen erklärbar ist, scheint es, als ob bei der Messung weitere Effekte auftreten.

Abbildung 25 verdeutlicht daher ein wesentliches Problem des CO₂-Sensors: Der verwendete NDIR-Sensor besitzt eine eingebaute Temperaturkompensation. Anhand der Messdaten liegt allerdings die Vermutung nahe, dass diese Kompensation nicht ausreicht. In der Grafik sind die Verläufe direkt nach Bepflanzung des Ökosystems zu sehen. Am Anfang des dargestellten Zeitraums findet daher noch keine Photosynthese statt. Im rot markierten Bereich tritt dennoch eine Abnahme des CO₂-Gehaltes auf. Da die Temperatur abfällt (vgl. blaue Kurve), könnte dieser Effekt an einer nicht vollständig kompensierten Temperaturabnahme liegen.

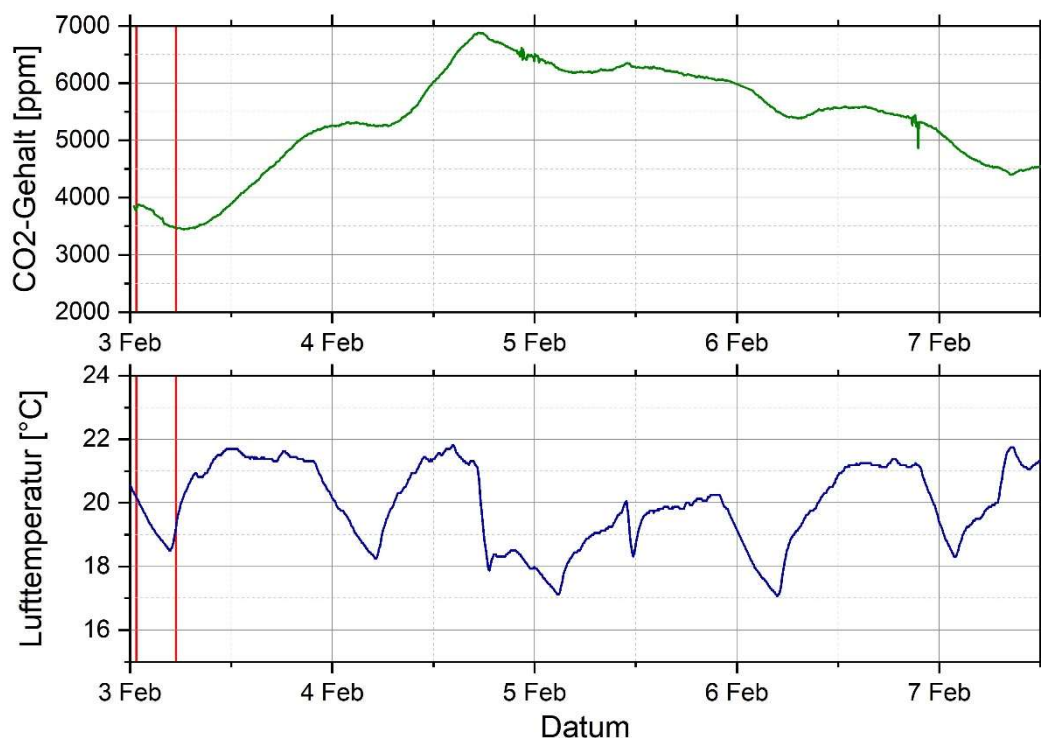


Abbildung 25: Vergleich des gemessenen CO₂-Gehaltes und der Lufttemperatur. Der rot markierte Bereich dient der Visualisierung eines potenziellen Sensoreffekts aufgrund einer nicht vollständig kompensierten Temperaturabnahme.

5.2.4 Vierte Phase: Naturwissenschaftliches Experimentieren

Aufbauend auf den bisher gewonnen Erkenntnissen entwickeln die Lernenden in dieser Phase eigene Fragestellungen. Das Ziel dieser Phase besteht darin ein zur eigenen Forscherfrage passendes Experiment zu planen und dieses vorzubereiten (Grothaus et al., 2023). In der Planung des Experiments soll auch auf die Prozesse der Datenspeicherung, -aufbereitung und -verarbeitung eingegangen werden. Die Forscherfrage, das geplante Experiment und die Techniken zur Datenaufbereitung stellen die Lernenden am Ende dieser Phase über ein Poster vor (Grothaus et al., 2023).

Nachfolgende Liste soll potenzielle Forscherfragen liefern und dabei auf die für die Planung relevanten Variablen eingehen (Tabelle nach Grothaus et al., 2023).

Forscherfrage	Relevante Variablen
Welche Lichtparameter (Intensität, Frequenz) sind besser für Wachstum?	Mehrere Vergleichsgläser, Helligkeit
Welche Pflanzen „bauen am meisten CO ₂ ab“?	Verschiedene Pflanzenarten, CO ₂ -Gehalt
Wie beeinflussen Schadstoffe das Ökosystemwachstum?	Schadstoff (z.B. Unkrautvernichter)
Wachsen Pflanzen bei höheren Temperaturen besser?	Mehrere Vergleichsgläser, Temperatur

Tabelle 5: Potenzielle Forscherfragen, die an einem „gläsernen Ökosystem“ bearbeitet werden können.

5.2.5 Fünfte Phase: Verallgemeinerung auf das globale Erdsystem

In der letzten Phase des Projektes werden Überlegungen aus der ersten Phase erneut aufgegriffen. Da die SchülerInnen im Laufe des Projektes verschiedene Einflussfaktoren auf ein Ökosystem am Modell untersucht haben, lassen sich diese Erkenntnisse auf das Erdsystem übertragen (Grothaus et al., 2023). Hierbei können die SchülerInnen feststellen, dass die natürlichen Gleichgewichtsprozesse essenziell für den Erhalt eines Ökosystems sind: Wird ein bestimmter Prozess im Ökosystem irreversibel aus dem Gleichgewicht gebracht, gehen die Pflanzen im gläsernen Ökosystem ein. Anhand dieser Erkenntnis können beispielsweise verschiedene Kipp-Punkte des Erdklimasystems behandelt werden (Grothaus et al., 2023).

Ein zweiter Aspekt dieser Phase besteht in der Anwendung der erlernten Data Literacy Kompetenzen auf das Ökosystem der Erde. Über die Auswertung realer Umweltdaten sollen die Konsequenzen der Klimakrise den SchülerInnen bewusst gemacht werden (Grothaus et al., 2023). So kann das Projekt potenziell zum Überwinden des Knowledge-Action-Gaps beitragen.

5.3 Einbindung in tet.folio

In diesem Abschnitt soll die Anbindung einer Mikrocontroller-Messwerterfassung an tet.folio, wie in Abschnitt 3.4 und 3.4.2 beschrieben, anhand des vorgestellten Schülerlabors demonstriert werden. Hierzu wurde ein tet.folio-Buch zum Projekt „Das gläserne Ökosystem“ erstellt. Dieses ist online abrufbar unter: <https://tetfolio.fu-berlin.de/tet/1790823>.

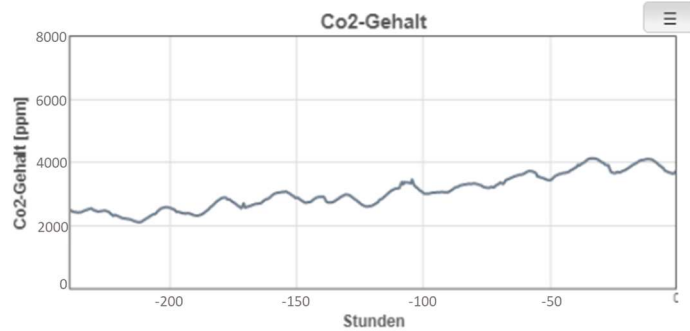
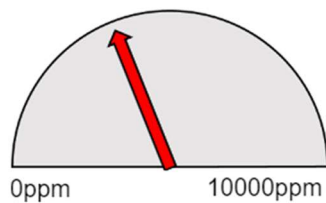
Über Java Script lassen sich die Messdaten vom Adafruit-Server abrufen und auf der tet.folio-Seite einbinden. Neben dem bloßen Anzeigen der Messwerte lassen sich technisch weitere Funktionen wie das Zeichnen eines Messgraphen realisieren. Zusätzlich können die Elemente auf der tet.folio-Seite mit den Messwerten verknüpft werden, sodass diese responsiv auf die Messwerte reagieren. Im vorgestellten Beispiel werden die aktuellen Werte einer Messvariable abgerufen und ein Messgraph der letzten 250 Stunden geplottet. Zusätzlich dreht sich der Pfeil der „analogen Anzeige“ in Abhängigkeit von einem Minimal- und Maximalwert. So kann beispielsweise der Füllstand eines Gefäßes oder der momentane Stand bezüglich eines kritischen Wertes verdeutlicht werden. Die Umsetzung in tet.folio sieht man in Abbildung 26 auf der nächsten Seite.

Das oben genannte tet.folio-Buch soll erste Umsetzungsbeispiele aufzeigen. Es bleibt allerdings Platz für die technische Optimierung. Eine verbesserte Darstellung der Zeitachse in den Messdiagrammen, eine adaptive Skala in der „analogen“ Anzeige oder die Möglichkeit des Herunterladens der Messdaten über die tet.folio-Seite sind erste Verbesserungsansätze. Ein weiteres Problem besteht darin, dass die Daten nur abgefragt und nicht auf der tet.folio-Seite gespeichert werden. Die Messgraphen werden daher nur geplottet, wenn im dargestellten Zeitraum eine aktive Messung stattgefunden hat. Das Zurückgreifen auf alte Daten oder Messgraphen ist daher nicht möglich. Darüber hinaus ist die Einbindung der AdafruitIO-Daten in tet.folio bisher nur mit den geeigneten Programmierkenntnissen in HTML, JavaScript und CSS möglich. Gerade weil die technischen Möglichkeiten aber nahezu unbegrenzt sind, wäre es wünschenswert die Anbindung in Zukunft sukzessiv zu optimieren. Neben der technischen Optimierung ist es im Rahmen zukünftiger Projekte denkbar, die Anwendung über vorgefertigte Bausteine und Anleitungen auch Anfängern zu ermöglichen.

2. Messwerte:

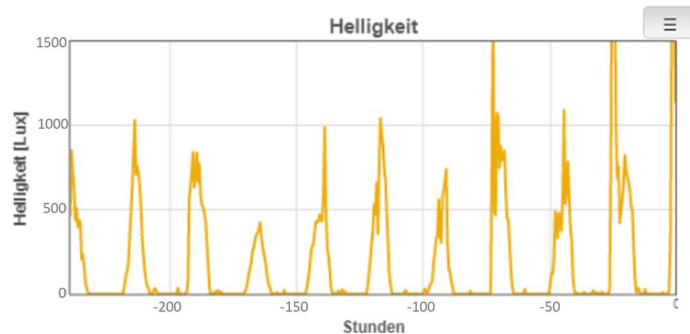
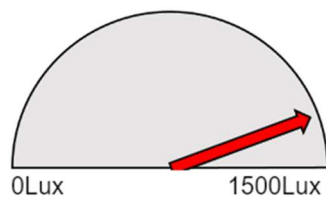
Co2-Gehalt:

Aktueller Messwert: 3786ppm



Helligkeit:

Aktueller Messwert: 1331Lux



Lufttemperatur:

Aktueller Messwert: 22.88°C

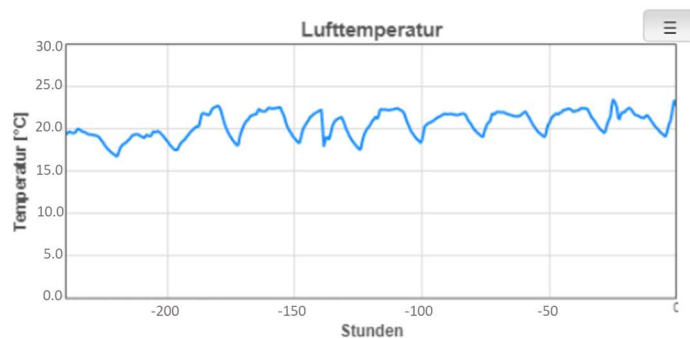
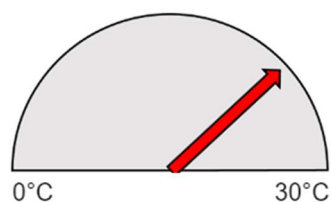


Abbildung 26: Umsetzungsbeispiel der Darstellung von Mikrocontroller-Messdaten in tet.folio. Bei jeder Aktualisierung der Website werden die Messwerte und -graphen automatisch von AdafruitIO abgefragt und auf der tet.folio-Seite angezeigt.

Zusätzlich findet sich eine aktuelle HTML-Version des tet.folio Buches auf der der Arbeit beigelegten Daten-CD. Hierbei soll allerdings darauf hingewiesen werden, dass das Abfragen aktueller Messwerte und Graphen nur bei bestehender Internetverbindung möglich ist. Des Weiteren werden die Messgraphen nur dann geplottet, wenn im gewählten Zeitraum eine aktive Messung stattfindet.

6 Fazit

In der Einleitung dieser Arbeit wurde die These aufgeworfen, dass Mikrocontroller eine Möglichkeit bieten, mit der Lernende auch zu Hause experimentieren können. Ein Grund hierfür ist vor allem der geringe Preis, der es ermöglicht, die Messsysteme in großer Anzahl anzuschaffen. Dadurch können alle Lernende mit eigenen Mess-Sets zuhause experimentelle Fragestellungen bearbeiten und die Messdaten über eine Internetschnittstelle online sichern, sodass die Daten auch in der Schule abrufbar sind.

Im Verlauf dieser Arbeit wird allerdings klar, dass das nur einen kleinen Bereich der Einsatzmöglichkeiten von Mikrocontrollersystemen im Schulkontext darstellt. Neben fachspezifischen Einsatzgebieten im Unterricht lassen sich mit Mikrocontrollern auch viele fachübergreifende Projekte realisieren (vgl. Kapitel 4). Eine weitere Möglichkeit ist der Einsatz im Rahmen von Schülerforschungsprojekten oder Schüler-AGs.

Die Erörterung verschiedener didaktischer Gesichtspunkte zeigt, dass der Einsatz von Mikrocontrollern neben den vielfältigen Einsatzmöglichkeiten und dem geringen Preis auch einen didaktischen Mehrwert liefern kann. Insbesondere der immer wichtiger werdende Erwerb digitaler Kompetenzen kann über die Arbeit mit Mikrocontrollern gefördert werden (vgl. Kapitel 3.1). Neben diesem überfachlichen Aspekt gibt es auch Argumente für den Mikrocontrollereinsatz speziell im Physikunterricht: Durch die selbstständige Arbeit mit Mikrocontrollern kann das Experimentieren im Unterricht zumindest wissenschaftsähnlich werden, wodurch der in den Bildungsstandards geforderte Auftrag zur Vermittlung wissenschaftlicher Arbeitsweisen erfüllt werden könnte (vgl. Kapitel 3.3). Es lässt sich also festhalten, dass die Einschränkung des Mikrocontrollereinsatzes auf die Durchführung von Experimenten zuhause, den vielfältigen Möglichkeiten nicht gerecht werden würde.

Zusammenfassend lässt sich sagen, dass der Mikrocontrollereinsatz in Schulen großes Potenzial hat. Dennoch gibt es in der Umsetzung Hürden, die viele Lehrkräfte vom Einsatz im Unterricht abhalten. Der Ansatz sollte trotzdem weiterverfolgt werden, weil neue Unterrichtsmedien wie digitale Schulbücher in Zukunft einen noch größeren Nutzen ermöglichen. Da es allerdings schon heute viele Hard- und Softwaresysteme gibt, die den Mikrocontrollereinsatz wesentlich vereinfachen, scheint die These naheliegend, dass es vor allem an didaktischen Konzepten zur Einbindung in den Unterricht fehlt. Das verfügbare Material beschränkt sich meist auf einzelne Anwendungsbeispiele, weshalb in Zukunft an einem fach- und jahrgangsstufenübergreifenden Konzept gearbeitet werden sollte.

7 Danksagung

Zum Schluss der Arbeit möchte ich ein besonderes Dankeschön an einige Personen richten, die mich bei der Anfertigung dieser Zulassungsarbeit unterstützt haben.

Ein besonderer Dank geht an Dr. Sebastian Haase der Freien Universität Berlin, der sich bei technischen Fragen zur Programmierung von tet.folio stets Zeit genommen hat, weiterzuhelfen. Ohne ihn wäre die Anbindung der Mikrocontroller an tet.folio im Rahmen dieser Arbeit nicht umsetzbar gewesen.

Ein weiterer Dank geht an Jonathan Grothaus für die Unterstützung bei der Realisierung des gläsernen Ökosystems. Von ihm stammt die Idee, ein verschlossenes Ökosystem mit Mikrocontrollern zu beobachten. Auch bei der Umsetzung der Messwerterfassung konnte er stets mit innovativen Verbesserungsvorschlägen weiterhelfen.

Zuletzt geht ein Dank an Dr. Markus Elsholz für das Betreuen meiner Zulassungsarbeit.

8 Literaturverzeichnis

- Adafruit IO.* (o. D.). Abgerufen am 10. Februar 2023, von <https://io.adafruit.com/>
- analogRead.* (o. D.). Arduino.cc. Abgerufen am 9. Februar 2023, von <https://www.arduino.cc/reference/de/language/functions/analog-io/analogread/>
- Arduino Education.* (o. D.). Arduino.cc. Abgerufen am 9. Februar 2023, von <https://www.arduino.cc/education>
- Arduino Hardware.* (o. D.). Arduino.cc. Abgerufen am 10. Februar 2023, von <https://www.arduino.cc/en/hardware>
- Arduino Software.* (o. D.). Arduino.cc. Abgerufen am 10. Februar 2023, von <https://www.arduino.cc/en/software>
- Arduino-Cloud.* (o. D.). Arduino.cc. Abgerufen am 10. Februar 2023, von <https://cloud.arduino.cc/>
- Arduino-Store.* (o. D.). Arduino.cc. Abgerufen am 10. Februar 2023, von <https://store.arduino.cc/>
- Bäck, K. I. (2017). *Hermetosphären – autarke Ökosysteme in verschlossenen Gläsern* [Diplomarbeit]. Karl-Franzens-Universität Graz.
- Bernstein, F., Keller, O., Schmeling, S. & Wilhelm, T. (2021). 3D-Druck und Mikrocontroller: Ein Dreamteam für Lowcost-Hightech-Experimente? *PhyDid B - Didaktik Der Physik - Beiträge Zur DPG-Frühjahrstagung, 1.* <https://ojs.dpg-physik.de/index.php/phydid-b/article/view/1155/1241>
- Brühlmann, T. (2019). *Arduino: Praxiseinstieg (mitp Professional)* (4. überarbeitete Auflage 2019). mitp.
- Chung, T. & Bagur. (2023, 9. März). *Guide to 3V3 and 5V Power Supplies Differences.* Arduino.cc. Abgerufen am 14. März 2023, von <https://docs.arduino.cc/learn/microcontrollers/5v-3v3>
- Datenblatt D1 mini ESP32.* (o. D.). az delivery. Abgerufen am 9. Februar 2023, von https://cdn.shopify.com/s/files/1/1509/1638/files/D1_Mini_ESP32_Datenblatt_AZ-Delivery_Vertriebs_GmbH.pdf
- Datenblatt D1mini ESP8266.* (o. D.). az delivery. Abgerufen am 10. Februar 2023, von https://cdn.shopify.com/s/files/1/1509/1638/files/Betriebsanleitung-AZ-D1miniV1.2_2.pdf
- De Florio-Hansen, I. (2020). *Digitalisierung, Künstliche Intelligenz und Robotik: Eine Einführung für Schule und Unterricht* (1. Aufl.). utb GmbH.
- Dreumel, V. W. (2016). *Physik-Experimente mit Arduino: 36 faszinierende Projekte für Schule und Hobby* (1. Aufl.). Elektor.
- Duden.de. (2022). Mi-k-ro-con-t-rol-ler. In *Duden online*. Abgerufen am 9. Februar 2023, von <https://www.duden.de/rechtschreibung/Mikrocontroller>
- Duit, R., Tesch, M. & Mikelski-Seifert, S. (2010, Februar). *PIKO-Brief Nr. 7: Das Experiment im Physikunterricht.* Abgerufen am 9. Februar 2023, von https://www.mathphys.uni-freiburg.de/physik/filk/public_html/InfoLehramt/FD17_QhxBz/pikobrief_Experimentieren.pdf

- Girwidz, R. & Watzka, B. (2018). Digitale Werkzeuge im Physikunterricht einsetzen: Mit Micro-Controllern und Mini-Computern einfach, kreativ und motivierend die Physik im Alltag verstehen lernen. *Naturwissenschaften Im Unterricht Physik*, 167, 2–5.
- Grothaus, J., Greubel, A., Dickmeis, L., Herold, A., Elsholz, M., Hennecke, M. & Trefzger, T. (2023). Das gläserne Ökosystem: Vorstellung eines fächerübergreifenden MINT-Projekts [Eingereicht]. 20. GI-Fachtagung „Informatik Und Schule“ (INFOS).
- Haase, S., Kirstein, J. & Nordmeier, V. (2016). tet.folio: Neue Ansätze zur digitalen Unterstützung individualisierten Lernens. *PhyDid B - Didaktik der Physik - Beiträge zur DPG-Frühjahrstagung*. <https://doi.org/10.17169/refubium-24768>
- Haase, S., Sommerer, M., Kirstein, J. & Nordmeier, V. (2021). tet.folio: Eine Online-Plattform für die Produktion innovativer Lehr-Lern-Angebote. *PhyDid B - Didaktik Der Physik - Beiträge Zur DPG-Frühjahrstagung*, 1. <https://ojs.dpg-physik.de/index.php/phydid-b/article/view/1117/1208>
- Henne, A., Beuter, A., Hachfeld, A., Schumann, S. & Möhrke, P. (2021). Naturwissenschaftlicher Unterricht während der Corona-Pandemie. 47. Jahrestagung Gesellschaft für Didaktik der Chemie und Physik (GDGP), 669–672. <https://doi.org/10.25656/01:22661>
- Höttecke, D. & Rieß, F. (2015). Naturwissenschaftliches Experimentieren im Lichte der jüngeren Wissenschaftsforschung – Auf der Suche nach einem authentischen Experimentbegriff der Fachdidaktik. *Zeitschrift für Didaktik der Naturwissenschaften*, 21(1), 127–139. <https://doi.org/10.1007/s40573-015-0030-z>
- Hoyer, C., Thoms, L.-J., Behrens, S. & Girwidz, R. (2018). Moderne RGB-Farbmischung - Pulsweitenmodulation mit dem Arduino. *Naturwissenschaften Im Unterricht Physik*, 167, 12–15.
- KMK. (2017). *Bildung in der digitalen Welt : Strategie der Kultusministerkonferenz*. Abgerufen am 9. Februar 2023, von https://www.kmk.org/fileadmin/pdf/PresseUndAktuelles/2018/Digitalstrategie_2017_mit_Weiterbildung.pdf
- KMK. (2020). *Bildungsstandards im Fach Physik für die Allgemeine Hochschulreife*. Abgerufen am 9. Februar 2023, von https://www.kmk.org/fileadmin/Dateien/veroeffentlichungen_beschluesse/2020/2020_06_18-BildungsstandardsAHR_Physik.pdf
- Krohn, W. (2010). Neues (und Bewährtes) aus der Grünen Schule. *Lynx-Druck*, 38–41. https://www.fs-hamburg.org/wp-content/uploads/PDF/LynxDruck_2010_02.pdf
- Kurth, C. & Wodzinski, R. (2018). Messwerterfassung mit dem Arduino in der Lehramtsausbildung. *PhyDid B - Didaktik Der Physik - Beiträge Zur DPG-Frühjahrstagung*, 1. <https://ojs.dpg-physik.de/index.php/phydid-b/article/view/842/982>
- Kurth, C. & Wodzinski, R. (2020). Veränderung der Einstellungen von Lehramtsstudierenden zum Einsatz digitaler Messwerterfassung: durch die Auseinandersetzung mit dem Arduino im didaktischen Experimentierpraktikum. *PhyDid B - Didaktik Der Physik - Beiträge Zur DPG-Frühjahrstagung*, 1. <https://ojs.dpg-physik.de/index.php/phydid-b/article/view/1068/1158>

- Lerch, R. (2016). *Elektrische Messtechnik: Analoge, digitale und computergestützte Verfahren*. Springer Verlag.
- Margolis, M., Jepson, B. & Weldin, N. (2020). *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects* (3rd edition) [E-Book]. O'Reilly Media.
- Mattern, F. & Flörkemeier, C. (2010). Vom Internet der Computer zum Internet der Dinge. *Informatik-Spektrum*, 33(2), 107–121. <https://doi.org/10.1007/s00287-010-0417-7>
- Modrow, E. (2011). Visuelle Programmierung - oder: Was lernt man aus Syntaxfehlern? *INFOS*, 27–36. <https://dl.gi.de/handle/20.500.12116/18500>
- Mußmann, F. & Hardwig, T. (2021, 1. Juni). *Digitalisierung im Schulsystem: Herausforderung für Arbeitszeit und Arbeitsbelastung von Lehrkräften* [Pressemeldung]. https://www.gew.de/fileadmin/media/sonstige_downloads/hv/Service/Presse/2021/Digitalisierung-im-Schulsystem---Studie.pdf
- Parthier, R. (2019). *Messtechnik: Vom SI-Einheitensystem über Bewertung von Messergebnissen zu Anwendungen der elektrischen Messtechnik* (9., überarb. u. erw. Aufl. 2020). Springer Vieweg.
- PGLU. (2022, 23. Mai). *Was ist visuelles Programmieren?* PGLU Prozessgesteuerte Lernumgebung. Abgerufen am 10. Februar 2023, von <https://pglu.ch/was-ist-visuelle-programmierung/>
- Richtberg, S. (2018). Woher wissen Funkuhren, wie spät es ist?: Funkübertragung von Zeitsignalen als Anwendung des offenen Schwingkreises. *Naturwissenschaft Im Unterricht Physik*, 167, 28–29.
- Siems, J. (2018). *Webbasierte Erfassung und Visualisierung von Messwerten am Beispiel des chaotischen Pohl'schen Rades* [Bachelorarbeit]. FU Berlin.
- Söderby, K. (2023, 7. Februar). *Getting Started with Arduino*. Arduino.cc. Abgerufen am 9. Februar 2023, von <https://docs.arduino.cc/learn/starting-guide/getting-started-arduino>
- Soltau, U. (o. D.). *Hermetosphären*. Abgerufen am 11. Februar 2023, von <https://ulfsoltau.wordpress.com/>
- Statista. (o. D.). *Smart Home - Deutschland : Statista Marktprognose*. Abgerufen am 10. Februar 2023, von <https://de.statista.com/outlook/dmo/smart-home/deutschland>
- Strobel, C. (2019, 6. März). Arduino vs. Raspberry Pi: Wo liegt der Unterschied? *techtag*. Abgerufen am 10. Februar 2023, von <https://www.techtage.de/it-und-hightech/arduino-vs-raspberry-pi-wo-liegt-der-unterschied/>
- Zierer, K. (2017). *Lernen 4.0 Pädagogik vor Technik. Möglichkeiten und Grenzen einer Digitalisierung im Bildungsbereich* (1. Aufl.). Schneider Verlag Hohengehren.
- Zierer, K. & Gottfried, T. (2023, 8. Januar). Digitaler Holzweg. *Zeit Online*. Abgerufen am 9. Februar 2023, von <https://www.zeit.de/gesellschaft/schule/2022-12/saarland-schulbuecher-abschaffung-bildung-digitalisierung>

A Anhang

Der folgende Arduino-Sketch wurde für die Beobachtung verschiedener Umweltparameter im Projekt des gläsernen Ökosystems verwendet (siehe Kapitel 5.2). Der interessierte Bastler kann den Sketch gerne für eigene Projekte kopieren, anpassen und weiterverwenden.

```
/*
 * The MIT License (MIT)

Copyright (c) 2015-2021 Adafruit Industries
Authors: Tony DiCola, Todd Treece, Brad Black

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
*/

/* Quellen:
 *
 * Hier findest du weitere Infos zu den verbauten Komponenten, die auch für diesen Sketch
verwendet wurden.
 *
 * BME280: https://funduino.de/nr-23-bme280-luftdruck-luftfeuchtigkeits-und-temperatursensor
 * DS18B20: https://draeger-it.blog/arduino-lektion-48-temperatursensor-ds18b20/
 * SMT50: https://www.gardenergranny.de/bludit/soil-moisture-sensor-smt50-and-arduino
 * SCD30: https://learn.adafruit.com/adafruit-scd30/arduino
 * ADS1115: https://wolles-elektronikkiste.de/ads1115
 * BH1750: https://create.arduino.cc/projecthub/infoelectorials/project-017-arduino-bh1750-light-sensor-project-640075
 *
 * Für die Einbindung des WiFiManagers und der Anbindung an AdafruitIO wurde auf ein fertiges
Code-Beispiel von Brad Black aus dem Jahr 2022 zugegriffen.
 * Dieses ist abrufbar unter:
https://github.com/adafruit/Adafruit\_IO\_Arduino/blob/master/examples/adafruitio\_27\_wifimanager/adafruitio\_27\_wifimanager.ino
*/

//Ab hier beginnt der eigentliche Sketch!

//Einbinden der benötigten Bibliotheken
#include <WiFiManager.h>
#include "AdafruitIO_WiFi.h"
#include <Wire.h>
#include <Seeed_BME280.h>
#include <DallasTemperature.h>
#include "SparkFun_SCD30_Arduino_Library.h"
#include<ADS1115_WE.h>
#include <BH1750.h>

//Einloggdaten für AdafruitIO (Wifi-Daten werden über WiFiManager übergeben)
char IO_USERNAME[64] = "[Hier deinen AdafruitIO Username eingeben]";
char IO_KEY[64] = "[Hier deinen AdafruitIO Key eingeben]";
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, "", "");
```

```

//Deklarieren der verwendeten digitalen Komponenten
WiFiManager wifiManager; //Setup des WifiManagers
#define ONE_WIRE_BUS 2 //One-Wire-Bus für DS18B20 an GPIO2
OneWire oneWire(ONE_WIRE_BUS); //Setup des One-Wire-Bus
DallasTemperature ds18b20(&oneWire); //DS18B20 mit OneWire verknüpfen
SCD30 scd30; //SCD30 deklarieren
BME280 bme280; //BME280 deklarieren
ADS1115_WE adc = ADS1115_WE(0x48); //ADS1115 deklarieren
BH1750 bh1750; //BH1750 deklarieren

//Deklarieren der für AdafruitIO verwendeten Feeds
AdafruitIO_Feed *Co2gehalt_feed = io.feed("okosystem.co2-gehalt");
AdafruitIO_Feed *Temp_feed = io.feed("okosystem.lufttemperatur");
AdafruitIO_Feed *LuftFeucht_feed = io.feed("okosystem.luftfeuchtigkeit");
AdafruitIO_Feed *BodenTemp_feed = io.feed("okosystem.bodentemperatur");
AdafruitIO_Feed *BodenFeucht_feed = io.feed("okosystem.bodenfeuchtigkeit");
AdafruitIO_Feed *Helligkeit_feed = io.feed("okosystem.helligkeit");

//Definition globaler Variablen
const float VRefer = 3.3;
const int pinO2 = 1;
const int pinTemp = 3;
const int pinMois = 2;
const int pause = 300;

float sensorValueTemperature = 0;
float sensorValueMoisture = 0;
float TempSMT = 0;
float BodenFeuchtSMT = 0;
float O2MIX = 0;
float Druck = 0;
float Pascal = 0;
float hPaBME = 0;
float TempBME = 0;
float LuftFeuchtBME = 0;
float TempDS = 0;
float Co2SCD = 0;
float TempSCD = 0;
float LuftFeuchtSCD = 0;
float Lux = 0;

void setup() //einmalige "Initialisierungs-Funktion"
{
    Serial.begin(9600); // Verbindung zu Serielllem Monitor (Baudrate 9600) (Eher ein Debug-
    Feature)
    delay(500);

    wifiManager.setClass("invert"); // enable "dark mode" for the config portal
    wifiManager.setConfigPortalTimeout(120); // auto close configportal after n seconds
    wifiManager.setAPClientCheck(true); // avoid timeout if client connected to softap

    if (!wifiManager.autoConnect("WiFi Setup")) // connect to wifi with existing setting or
    start config
    {
        Serial.println("failed to connect and hit timeout");
    }
    else
    {
        // if you get here you have connected to the WiFi
        Serial.println("Connected to WiFi.");
        Serial.printf("Connecting to Adafruit IO with User: %s Key: %s.\n", IO_USERNAME, IO_KEY);
        io.connect();

        // wait for a connection
        while ((io.status() < AIO_CONNECTED))
        {
            Serial.print(".");
            delay(500);
        }
        Serial.println("Connected to Adafruit IO.");
    }
}

```

```

// Erst hier startet die Kommunikation mit den digitalen Komponenten
Wire.begin();
bme280.init();
ds18b20.begin();
scd30.begin();
adc.init();
bh1750.begin();
}

void loop() // "Endlosschleife" - Hier passieren die eigentlichen Messungen
{
    io.run(); //muss IMMER als erste Funktion im Loop stehen, um Connection mit AdafruitIO
    sicherzustellen

    //Auslesen BME280
    Pascal = bme280.getPressure();
    hPaBME = Pascal/100;
    TempBME = bme280.getTemperature();
    LuftFeuchtBME = bme280.getHumidity();

    //Auslesen DS18B20
    ds18b20.requestTemperatures();
    TempDS=ds18b20.getTempCByIndex(0);

    //Auslesen SMT50 (Definition der Funktionen siehe unten)
    TempSMT = readSMT50Temp();
    BodenFeuchtSMT = readSMT50Feucht();

    //Auslesen SCD30
    Co2SCD = scd30.getCO2();
    TempSCD = scd30.getTemperature();
    LuftFeuchtSCD = scd30.getHumidity();

    // Auslesen BH1750
    Lux = bh1750.readLightLevel();

    //Übergabe der Werte an AdafruitIO
    Co2gehalt_feed->save(Co2SCD);
    Temp_feed->save(TempDS);
    LuftFeucht_feed->save(LuftFeuchtBME);
    BodenFeucht_feed->save(BodenFeuchtSMT);
    BodenTemp_feed->save(TempSMT);
    Helligkeit_feed->save(Lux);

    delay(pause*1000); //Wert für "pause" in Sekunden ist oben definiert
}

float readSMT50Temp() //Temperaturabfrage SMT50
{
    float sum = 0;

    for(int i=0; i<5; i++)
    {
        sum += readChannel(ADS1115_COMP_3_GND);
        delay(500);
    }
    float ValueTemperature = sum/5;
    float TempSMTRead = (ValueTemperature-0.5)*100.00; //Berechnung der Temperatur
    laut Anleitung

    return TempSMTRead;
}

float readSMT50Feucht() //Bodenfeuchteabfrage SMT50
{
    float sum = 0;

    for(int i=0; i<5; i++)
    {
        sum += readChannel(ADS1115_COMP_2_GND);
        delay(500);
    }

```

```

    }

    float sensorValueMoisture = sum/5;
    float BodenFeuchtSMTRead = sensorValueMoisture *50.00/3.00    //Berechnung der
    Bodenfeuchtigkeit laut Anleitung

    return BodenFeuchtSMTRead;
}

float readChannel(ADS1115_MUX channel) //Auslesen des AD-Wandlers in Abhängigkeit des Channels
{
    float voltage = 0.0;
    adc.setCompareChannels(channel);
    adc.startSingleMeasurement();
    while(adc.isBusy()){ }
    voltage = adc.getResult_V();
    return voltage;
}

```

B Erklärung nach §29 LPO-I

Ich versichere an Eides statt durch meine Unterschrift, dass ich die vorstehende Arbeit selbständig und ohne fremde Hilfe angefertigt habe. Alle Stellen, die wörtlich, annähernd wörtlich oder dem Sinn nach aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Des Weiteren erkläre ich, dass ich mich keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe. Diese Versicherung gilt auch für Abbildungen, Tabellen und ähnliches.

Die vorliegende Arbeit wurde in dieser oder ähnlicher Form an keiner weiteren Prüfungsstelle eingereicht.

A handwritten signature in black ink, consisting of a stylized 'L' followed by a series of loops and a long horizontal stroke.

Würzburg, der 15.3.2023