



MASTER THESIS | MASTER'S THESIS

Titel | Title

Angewandte Digitaltechnik mit "Physical Computing" Illustration und Erprobung von
Grundlagen der Technischen Informatik für Lehrpersonen und Schüler*innen der Sekun-
darstufe 1 und AHS-Sekundarstufe 2

verfasst von | submitted by

Dipl.-Ing. (FH) Mario Wolfgang Wüschner MA MSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Arts (MA)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree pro-
gramme code as it appears on the student
record sheet:

UA 992 499

Universitätslehrgang lt. Studienblatt | Post-
graduate programme as it appears on the
student record sheet:

Studium Generale - Das nachberufliche Studium an
der Universität Wien (MA)

Betreut von | Supervisor:

Univ.-Prof. Dipl.-Math. Dr. Peter Reichl Privatdoz.

Abkürzungsverzeichnis

ACK	<i>Acknowledgment</i>	MISO	<i>Master In Slave Out</i>
ADC	<i>Analog-Digital-Converter</i>	MOSFET	<i>Metal Oxide Semiconductor Field-Effect Transistor</i>
BBC	<i>British Broadcasting Corporation</i>		
BIT	<i>Binary Digit</i>	MOSI	<i>Master Out Slave In</i>
BMBWF	<i>Bundesministerium für Bildung, Wissenschaft und Forschung</i>	MSB	<i>Most Significant Bit</i>
CC	<i>Creative Commons</i>	NACK	<i>Negative Acknowledgment</i>
CPU	<i>Central Processing Unit</i>	OER	<i>Open-Educational-Resources</i>
CS	<i>Chip-Select</i>	PDF	<i>Portable Document Format</i>
CSGP	<i>Citizen Science Global Partnership</i>	PWM	<i>Pulsweiten-Modulation</i>
DGB	<i>Digitale Grundbildung</i>	RAM	<i>Random Access Memory</i>
DIY	<i>Do It Yourself</i>	REPL	<i>Read-Eval-Print-Loop</i>
DRAM	<i>Dynamic Random Access Memory</i>	ROM	<i>Read-only Memory</i>
GPIO	<i>General Purpose-Input/Output</i>	RS-Flip-Flop	<i>Reset-Set-Flip-Flop</i>
GPS	<i>Global Positioning System</i>	RTC	<i>Real-Time-Counter</i>
GPU	<i>Graphics Processing Unit</i>	SCL	<i>Serial Clock</i>
HDD	<i>Hard-Disk-Drive</i>	SCLK	<i>Serial Clock</i>
I/O-Ports	<i>Input-Output-Ports</i>	SDA	<i>Serial Data Line</i>
I2C	<i>Inter-Integrated-Circuit</i>	SFR	<i>Special Function Register</i>
IC	<i>Integrated Circuit</i>	SPI	<i>Serial Peripheral Interface</i>
IoT	<i>Internet of Things</i>	SRAM	<i>Static Random-Access Memory</i>
IPCEI	<i>Important Project of Common European Interest</i>	SS	<i>Slave-Select</i>
LED	<i>Light-emitting Diodes</i>	SSD	<i>Solid-State-Disc</i>
LSB	<i>Least Significant Bit</i>	STEAM	<i>Science, Technology, Engineering, Arts and Mathematics</i>
MAC	<i>Media Access Controller</i>	STEM	<i>Science, Technology, Engineering and Mathematics</i>
MCU	<i>Mikrocontroller-Unit</i>	UART	<i>Universal Asynchronous Receiver/Transmitter</i>
MINT	<i>Mathematik, Informatik, Naturwissenschaften, Technik</i>	VPL	<i>Visual Programming Language</i>

Vorwort

Das Suchen und Finden eines passenden Themas für die Masterthesis ist oft eine der größten Herausforderungen. Dies galt nicht für mich, denn spätestens seit dem Modul „Informatik“ bei Univ.-Prof. Dipl.-Math. Dr. Peter Reichl war mir klar, in welche Richtung ich gerne forschen will. Mich hat als Informatik-Lehrperson die Vorlesung deswegen besonders beeindruckt, da Dr. Reichl die Technische Informatik von vornherein als Kombination von Mathematik, Physik, Informatik und Elektronik gesehen und dies tatsächlich auch derart interdisziplinär unterrichtet hat. Anders ist es jedoch im Schulwesen der Sekundarstufe 1 und Sekundarstufe 2 an Gymnasien, denn hier werden diese Themen auf unterschiedliche Lehrfächer verteilt. Dies leider ohne konkrete Priorisierung oder Abstimmung mit anderen Fachschaften. Die Folge davon ist, dass Informatik speziell an Gymnasien nicht unmittelbar als interdisziplinär wahrgenommen und unterrichtet wird.

Meine Idee ist es nun, die Grundlagen der Technischen Informatik bzw. der Digitaltechnik mit dem interdisziplinären Ansatz von Dr. Reichl und einem populären Mikrocontroller zu verbinden. Die konkrete didaktische Umsetzung soll mit Ansätzen von Making, Tinkering, Physical Computing und visuellem bzw. textuellem Programmieren erfolgen. In Summe also Digitaltechnik bzw. Technische Informatik als spannender Unterrichtsmix aus elektronischem Basteln, individuellem, explorativem Lernen mitsamt kreativer Digitalität.

Ich danke somit Dr. Reichl für seine großartige Vorlesung im Zuge des Studium Generale, für sein inspirierendes Lehrbuch „*Eine kurze Geschichte der Technischen Informatik*“ und natürlich für seine Unterstützung für das Zustandekommen dieser Masterarbeit. Ebenfalls möchte ich die Pädagogische Hochschule Vorarlberg erwähnen, die es mir ermöglichte, die Inhalte dieser Masterthesis mit zwei verschiedenen Gruppen an Lehrpersonen durch zwei Fortbildungen zu testen bzw. zu evaluieren. Eine große Hilfe war mir auch Frau Dr. Andrea Christen durch Ihre Beratung beim wissenschaftlichen Design dieser Masterarbeit. Und schlussendlich möchte ich mich ganz besonders bei meiner Frau Susanne Wüschner bedanken, die mir eine wertvolle Hilfe war und mich immer bedingungslos unterstützt.

Inhaltsverzeichnis

1. Einleitung	1
I. Theoretischer Teil	5
2. Allgemeines und Definitionen	5
2.1 Digitaltechnik und Technische Informatik	5
2.2 Herausforderung Medien- und Informatikunterricht	7
2.3 Making, Tinkering	8
2.4 Physical Computing	9
2.5 Internet of Things	10
3. Grundlagen der Digitaltechnik und Technischen Informatik.....	11
3.1 Informationen.....	11
3.2 Analoge und digitale Informationen	12
3.3 Binäre Darstellung und Stellenwerte	13
3.4 Digitalisierung von Signalen, Pulsweiten-Modulation.....	14
3.5 Boolesche Logik, Logik-Gatter, Integrated Circuits (ICs)	17
3.6 Taktgeber und Timer	21
3.7 Kommunikationsschnittstellen	23
3.7.1 Universal Asynchronous Receiver/Transmitter (UART).....	23
3.7.2 Serial Peripheral Interface (SPI)	24
3.7.3 Inter-Integrated-Circuit (IC2)	25
3.7.4 Radio-Protokoll	26
3.8 Elektronische Speicher	27
3.8.1 Static Random-Access Memory (SRAM)	29

3.8.2 Dynamic Random-Access Memory (DRAM)	29
3.8.3 Register	30
3.8.4 Solid-State-Disc (SSD) und Flash-Speicher.....	30
3.8.5 Read-only Memory (ROM)	32
3.9 Mikrocontroller versus Mikroprozessor	33
4. Digitale Grundlagen im AHS-Lehrplan und IT-Initiativen	35
4.1 Lehrplan DGB für AHS-Unterstufe.....	35
4.1.1 Konzepte und Kompetenzen	36
4.1.2 Digitaltechnik, Programmierung und Making	37
4.2 Lehrplan Informatik AHS Oberstufe	40
4.3 Initiative MINT.....	43
4.3.1 MINT im Hochschulbereich	43
4.3.2 MINT in der Pädagog*innen-Ausbildung.....	44
4.3.3 MINT-Gütesiegel.....	44
4.3.4 Citizen Science	45
4.3.5 Programmieren als schulische Kernkompetenz	46
4.3.6 EU Code Week	48
4.3.7 Micro:bit - Das Schulbuch.....	48
4.3.8 EU-Initiative "Important Project of Common European Interest" (IPCEI)	49
4.3.9 STEM- bzw. STEAM-Initiative	49
5. Mikrocontroller und Programmiersprachen	51
5.1 Arduino Uno R3	51
5.2 Arduino Nano RP2040 Connect.....	52

5.3 ESP32-Mikrocontroller einzeln oder als CyberPi	52
5.4 Circuit Playground Express von Adafruit	53
5.5 BBC-Micro:bit V2 Go	54
5.6 Programmiersprachen	55
5.6.1 Python	56
5.6.1.1 MicroPython	56
5.6.1.2 CircuitPython	57
5.6.2 C/C++ bzw. Arduino-C	58
5.6.3 Visuelle Programmiersprachen VPL	58
II. Praxisprojekte	59
6. Beschreibung der Praxisprojekte	59
6.1 Projekt 1 – Reihen- und Parallelschaltung	60
6.2 Projekt 2 – Stromschalten mit PIN	60
6.3 Projekt 3 – Ampelschaltung	61
6.4 Projekt 4 – Potentiometer	62
6.5 Projekt 5 – Pulsweiten-Modulation (PWM)	63
6.6 Projekt 6 – Binärschaltung	64
6.7 Projekt 7 – interner Lichtsensor	65
6.8 Projekt 8 – LED-Stripes (NeoPixel)	66
6.9 Projekt 10 – Modul Radio	67
6.10 Projekt 11 - MOSFET	68
III. Methodischer Teil	70
7. Auswahl Mikrocontroller, Programmiersprache, Schaltungssoftware	70

7.1 Mikrocontroller	71
7.2 Programmiersprache.....	73
7.3 Schaltungssoftware	75
8. Evaluation der Schulung für Gruppe 1 und Gruppe 2	76
8.1 Datenerhebung, Fragen und Thesen	76
8.2 Ergebnisse	80
9. Zusammenfassung und Ausblick	100
10. Literatur- und Quellenverzeichnis	103
11. Abbildungsverzeichnis	109
12. Tabellenverzeichnis	113
13. Anhang	114
13.1 Legistischer Vorentwurf Lehrplan Informatik AHS-Oberstufe	114
13.2 Praxisprojekt 1 (Reihen-Parallelschaltung)	130
13.3 Praxisprojekt 2 (Stromschalten per PIN)	136
13.4 Praxisprojekt 3 (Ampelschaltung)	143
13.5 Praxisprojekt 4 (Potentiometer)	152
13.6 Praxisprojekt 5 (PWM)	160
13.7 Praxisprojekt 6 (Binärschaltung)	170
13.8 Praxisprojekt 7 (Interner Lichtsensor).....	180
13.9 Praxisprojekt 8 (NeoPixel)	189
13.10 Praxisprojekt 10 (Radio-Protokoll)	197
13.11 Praxisprojekt 11 (MOSFET)	203
14. Abstract.....	214

1. Einleitung

Technische Informatik, Digitaltechnik bzw. binäre Grundlagen sind informatische Themen, die in der Sekundarstufe 1 (SEK1 - Gymnasium Unterstufe, Mittelschulen und sonderpädagogische Schulen) im neuen Unterrichtsfach „Digitale Grundbildung – DGB“ bzw. in der Sekundarstufe 2 (SEK2) der Gymnasien im Lehrplan ansatzweise und ohne exakte Details genannt sind. Ein Problemfeld ist, dass die zur Verfügung stehenden Unterrichtsstunden mit je einer Wochenstunde DGB über vier Jahre in der SEK1 bzw. in der SEK2 verpflichtend insgesamt nur zwei Wochenstunden einmalig in der 9. Schulstufe für die tatsächliche Bedeutung von DGB bzw. Informatik mit Sicherheit viel zu wenig sind. In der SEK2 der Gymnasien ist eine Verpflichtung von lediglich zwei Unterrichtsstunden Informatik (mit Ausnahme von individuell schulautonomen Regelungen oder dem Anbieten eines Wahlpflichtfachs Informatik) mit Sicherheit nicht ausreichend. Dies ist sogar eine bildungstechnische Benachteiligung von AHS-Schüler*innen im Vergleich zu Schüler*innen an berufsbildenden Schulen. Vorausblickend scheint es aber ab dem Schuljahr 2025/2026 in der SEK2 zu einer Erhöhung des Stundenausmaßes im Zuge eines neuen Lehrplans zu kommen. Ein vorläufiger, legislatisch unveröffentlichter Vorentwurf ist im Anhang beigelegt. Das zweite Problemfeld besteht darin, dass Technische Informatik, Digitaltechnik und binäre Grundlagen zwar in den Lehrplänen DGB bzw. Informatik genannt sind, jedoch nicht als bewusst interdisziplinär aus den Bereichen Mathematik, Physik, Elektronik und Schaltungstechnik samt Programmierung wahrgenommen werden. Diese fünf Bereiche werden im Schulbetrieb auf unterschiedliche Jahrgänge bzw. auf unterschiedliche Lehrfächer verteilt. Hier gibt es keinen einheitlich logischen Aufbau und keine verpflichtende, fächerübergreifende Zusammenarbeit von Lehrpersonen.

Die vorliegende Masterthesis befasst sich mit zehn einzelnen Projekten mit dem Mikrocontroller Micro:bit samt elektronischen Bauteilen und ausprogrammierten Funktionen. Damit sollen die Themen Technische Informatik bzw. Digitaltechnik mit ihren wichtigsten Grundlagen aus Mathematik, Physik, Elektronik, Schaltungstechnik und Programmierung direkt in den Fächern DGB und Informatik gebündelt verbleiben. Dies dient

dazu, den Schüler*innen auf diese Art und Weise die technischen Zusammenhänge besser und vor allem kompakt vermitteln zu können. Die zehn Projekte teilen sich in unterschiedliche Schwierigkeitsstufen. Sie sollten in der SEK2 zur Gänze mit einer textuellen Programmiersprache bzw. in der SEK1 zu großen Teilen unter Einsatz einer visuellen Programmiersprache umsetzbar sein.

Die übergreifende Forschungsfrage dieser Masterthesis beschäftigt sich konkret damit, ob ein Einstieg in die sehr komplexen Themen Technische Informatik bzw. Digitaltechnik über eine ganztägige Fortbildung für DGB- und Informatiklehrpersonen mitsamt zehn dokumentierten Referenzprojekten erfolgreich geschafft werden kann. Erwartet wird, dass die geschulten Lehrpersonen diese Thematik tatsächlich nach erfolgreicher Evaluierung in SEK1- und SEK2-Schulen erfolgreich implementieren können.

Im Zentrum der Evaluierung von zwei Fortbildungen sind sieben Thesen zur konkreten Forschungsfrage relevant:

Die Kombination von Design-Entwurf mit TinkerCAD, praktischem Aufbau am Steckbrett und visueller und textueller Programmierung wird von mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 als erleichternd wahrgenommen (These 1).

Die visuelle Programmierung mit MakeCode als Einstieg in die Programmierung wird von mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 als erleichternd wahrgenommen (These 2).

Die visuelle Programmierung mit MakeCode erleichtert für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 das Verständnis für die textuelle Programmierung mit MicroPython (These 3).

Die Kombination aus Unterricht und angeleitetem Nachbau samt Erklärung der einzelnen Projekte ist für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 eine Hilfe für das Verständnis der Thematik "Computertechnik" (These 4).

Die fertig programmierten und dokumentierten Projektbeispiele als gespeicherte Sicherung (Backup) stellen für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 eine Erleichterung für den eigenen Unterricht dar (These 5).

Der Gesamtzusammenhang von „Technischer Informatik“ mit der Kombination von Digital- und Analogtechnik und Programmierung ist für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 aus den Projekten erkennbar. (These 6)

Das Übungsbeispiel X gibt mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 Gelegenheit für Adaptierungen und neue Ideen. (These 7)

Es wird erwartet, dass alle sieben Thesen mit mindestens 50 % Zustimmung der befragten Teilnehmenden angenommen werden. Damit wird auch die Forschungsfrage nach der Eignung der Schulung bzw. der Schulungsunterlagen beantwortet. Die teilnehmenden Lehrpersonen sind somit Evaluator*innen und gleichzeitig mögliche Multiplikator*innen an ihren Schulen.

Die vorliegende Masterthesis ist wie folgt aufgebaut:

Der Theorieteil beinhaltet die Kapitel 2 bis 5. Kapitel 2 beleuchtet technisch-pädagogische Betrachtungen für den Unterricht von Digitaltechnik bzw. Technische Informatik. Dieser Teil ist zentral für den didaktischen Bildungsansatz durch Physical Computing. Sehr Technik lastig ist das Kapitel 3, in dem wichtige technische Grundlagen der Digitaltechnik erklärt werden, die auch in den Praxisprojekten implementiert sind. Von Bedeutung ist es auch, den Unterschied zwischen Mikrocontroller und Mikroprozessor zu erklären und zu verstehen. Kapitel 4 befasst sich mit den Lehrplänen für DGB und Informatik an Österreichs Gymnasien und versucht hier, die relevanten Lehrplaninhalte von Technischer Informatik bzw. Digitaltechnik zu orten. Aber auch österreich- bzw. weltweite Initiativen wie MINT oder STEAM werden in diesem Kapitel erklärt. Eine Vorauswahl von vier Mikrocontrollern zur späteren Evaluierung des geeignetsten Lehrgerätes bzw. dessen Programmiersprache wird im Kapitel 5 thematisiert. Kapitel 6 stellt zehn Projekte vor, die als Referenzprojekte für den Unterricht zu Digitaltechnik dienen sollen. Dieses Kapitel ist sogleich auch der Praxisteil. Der methodische Teil beginnt mit dem Kapitel 7, wo mit dem Micro:bit V2 samt den Programmiersprachen MakeCode und MicroPython die Auswahl des am besten geeigneten Mikrocontrollers getroffen wird. Sämtliche Bewertungskriterien bzw. deren Auswertung sind hier angeführt, ebenso die Begründung für die Wahl der Programmiersprachen MicroPython und MakeCode. Die

zweimal durchgeführte Online-Umfrage mit zwei Gruppen an Lehrpersonen wird im Kapitel 8 erklärt, beschrieben und schlussendlich interpretiert. Mit der Zusammenfassung und dem Ausblick beschließt das Kapitel 9 die vorliegende Masterthesis.

I. Theoretischer Teil

2. Allgemeines und Definitionen

In diesem Kapitel werden wichtige Funktionen und Definitionen der Technischen Informatik erläutert. Dabei teilen sich die Erläuterungen in pädagogische und technische Begrifflichkeiten, deren Kombination schlussendlich wichtig für das Zustandekommen eines zeitgemäßen Informatik-Unterrichts ist. Speziell in der Technischen Informatik gilt es, den klassischen Frontalunterricht durch eine Lehrperson durch Schüler*innen-zentrierte Methodiken zu ergänzen.

2.1 Digitaltechnik und Technische Informatik

Die Definition von Digitaltechnik ist in der Literatur nicht einheitlich festgelegt. Digitaltechnik ist ein umfassendes und vielschichtiges Gebiet, das weit über die einfache Darstellung von Informationen in Form von "1" und "0" hinausgeht. Es vereint verschiedene Disziplinen und ist ein zentrales Element moderner Technologie.

„Die Digitaltechnik ist ein Teil der Elektronik und Informationstechnik zur Erfassung, Darstellung, Verarbeitung und Übertragung digitaler Signale. Sie gliedert sich in die Bereiche mathematische Grundlagen (Theorie), Schaltkreistechnik in der Elektronik und Anwendungen in der Informatik und Kommunikationstechnik.“ (SIEMERS & SIKORA, 2007, S. 19)

Eine ganzheitliche Betrachtung und Vermittlung der Digitaltechnik fordern das Einbeziehen mehrerer Teilbereiche gemäß der folgenden Abbildung 1:

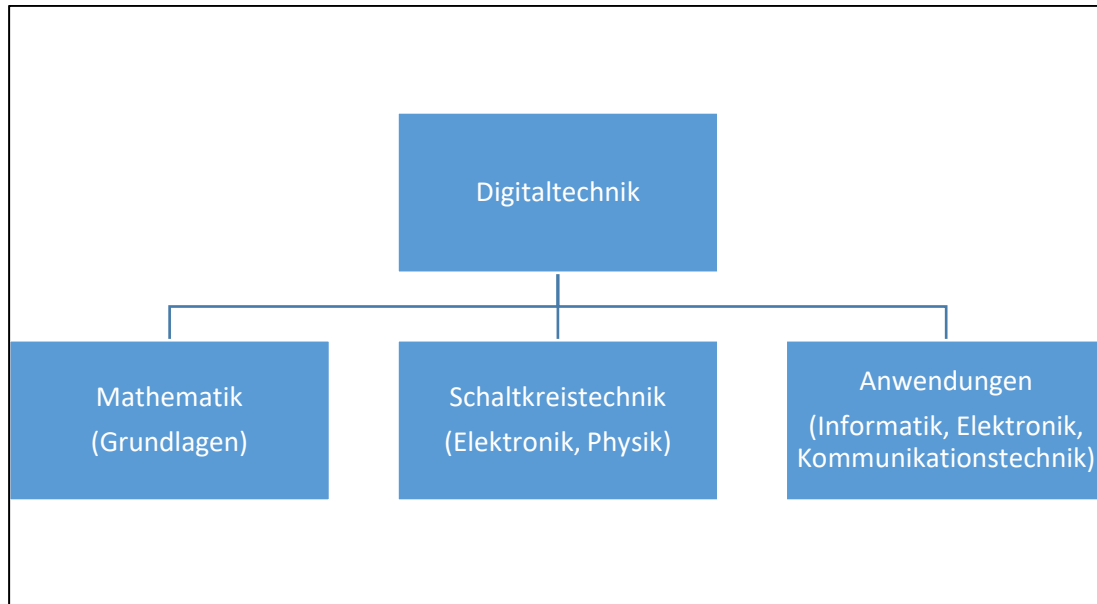


Abbildung 1: Teilbereiche der Digitaltechnik (eigene Abbildung)

Mathematik

Die Grundlage der Digitaltechnik ist die Mathematik, insbesondere die diskrete Mathematik, die sich mit endlichen oder abzählbaren Mengen beschäftigt. Konzepte wie Boolesche Algebra und Zahlensysteme (binär, oktal, dezimal, hexadezimal) sind die Grundvoraussetzung für das Verständnis und die Entwicklung digitaler Systeme.

Physik

Die Physik, speziell die Elektronik und die Halbleiterphysik, spielt eine entscheidende Rolle. Das Verständnis von elektrischen Schaltkreisen, Transistoren, Halbleitermaterialien, Speichermedien und deren Verhalten unter verschiedenen Bedingungen ist grundlegend für das Erstellen von digitalen Schaltungen.

Informatik

Informatik, hier speziell die Technische Informatik, bildet das Herzstück der Digitaltechnik. Sie umfasst die Entwicklung von Software, Algorithmen, Datenstrukturen und die Implementierung von Programmiersprachen. Die Umsetzung erfolgt in Mikrocontrollern, Mikroprozessoren, digitalen Speichern und den Logikbausteinen (SIEMERS & SIKORA, 2007, S. 19). Da der Begriff „Digitaltechnik“ oft nur für digitale sowie logische Schaltungen verwendet wird, ist die Technische Informatik wichtig, denn diese erweitert die Digitaltechnik mit Hardware, Software und vor allem Programmierung.

Elektronik

Die Elektronik beschäftigt sich mit der Gestaltung und Analyse von Schaltungen, die digitale Signale verarbeiten. Hierzu gehören logische Gatter, Flip-Flops, Zähler, Register und komplexere integrierte Schaltungen (ICs) wie Mikroprozessoren, Mikrocontroller sowie Speicherchips. Ebenso sind elektronische Komponenten wie Widerstände, Kondensatoren, LEDs, Sensoren und Schalter von großer Bedeutung. Diese Bauteile werden in digitalen Schaltungen verwendet, um Signale zu steuern, zu verstärken, zu speichern oder anzuzeigen.

Kommunikationstechnik

Digitale Kommunikation ist ein wesentlicher Bestandteil der modernen Informationsübertragung. Dies umfasst die Modulation und Demodulation von Signalen, die Datenübertragung über verschiedene Medien (drahtlos, drahtgebunden) mit Netzwerktechnologien.

Die fächerübergreifende Integration dieser Disziplinen in den Informatikunterricht bzw. die Festigung dieser Wissensbereiche auch im Schulfach *Informatik* bzw. *Digitale Grundbildung* (DGB) ist essenziell, um ein umfassendes Verständnis der Digitaltechnik zu vermitteln. Es ist wichtig, den Schüler*innen zu zeigen, wie diese Teilbereiche ineinandergreifen und zusammenarbeiten in der Bildung, im Beruf oder im privaten Umfeld. Ein möglicher, erfolgreicher Ansatz ist hier *Physical Computing* (siehe Kapitel 2.4).

2.2 Herausforderung Medien- und Informatikunterricht

Eine spezielle Herausforderung im Schulwesen ist mit Sicherheit der Unterricht von informatischen Inhalten. Hier gibt es einerseits die Vorgabe, Theorie- und Grundlagenwissen analog zu anderen Unterrichtsfächern zu sichern. Zusätzlich wird das erworbene Wissen auch auf einem digitalen Standardgerät (iPad, Notebook, Schul-PC) in der Schule angewandt. Speziell bei digital-technischen Grundlagen entsteht jedoch die Notwendigkeit, das digitale Standardgerät zu erweitern. Im konkreten Fall durch den Mikrocontroller Micro:bit samt externer elektronischer Komponenten wie Steckbrett, Logik-Gatter, Widerstände, *Light-emitting Diodes* (LED) oder gar Sensoren. Hürzeler u.a. (2023) nennen elf Prinzipien eines guten Medien- und Informatik-Unterrichts. Interessant ist dabei

das vierte Prinzip, das eine kreative und aktive Lernumgebung fordert.

*„Auch im Bereich der Informatik sollen Schüler*innen durch aktives Handeln Autonomie erleben und ‚Informatikprodukte‘ nicht nur konsumieren, sondern selbst produzieren. Dabei soll das Erleben der Informatik als kreative und gestaltende Fachwissenschaft als Ziel mitgedacht werden.“ (HÜRZELER u. a., 2023, S. 11)*

Der Einsatz von zusätzlichen analogen und digitalen Lehrmitteln ist ausdrücklich zu begrüßen (HÜRZELER u. a., 2023, S. 24). Dabei ist es nicht notwendig, IT-Wissen über zusätzliche, teure Schulinfrastruktur zu sichern, auch digitale Simulationen oder ergänzende, einfache Lehrgeräte (beispielsweise der Mikrocontroller *Micro:bit*) sind für die Veranschaulichung von digitalen Vorgängen völlig ausreichend. Sogar analoge Unterrichtsmethoden für Informatik haben sich etabliert. Besonders bekannt ist die Organisation *CS Unplugged*¹, ein Projekt der Computer Science Education Research Group der University of Canterbury in Neuseeland, die kosten- und nutzungsfrei Lern- und Lehrinhalte zum Thema Informatik online zur Verfügung stellt.

2.3 Making, Tinkering

Making wird als „die Verbindung von digitaler und kreativer Arbeit“ beschrieben, fortgeschrittenes Making kann aber auch „das Bauen und Programmieren eines eigenen Roboters bedeuten“ (INTERNET-ABC, 2020). Hier stoßen scheinbar zwei Welten aufeinander: eine streng hierarchische Programmierung auf der einen Seite und ein kreatives Produzieren mittels Making. Making im Schulbereich führt schlussendlich zum Begriff *Maker Education*. Hierbei handelt es sich um ein medienpädagogisches Konzept, „das sich an das Machen anschließt: der Reflexion des Tuns“ (KNAUS & SCHMIDT, 2020, S. 2). Dabei wird der Begriff *Making* um wichtige Komponenten aus Schüler*innen-Sicht erweitert. Die selbst erlebte Erfahrung, der Weg zum Endprodukt samt den Hindernissen und der Austausch mit anderen vollziehen einen Wechsel weg von der Lehrperson-Fokussierung hin zu einer selbst erlebten Lernerfahrung.

Eng verbunden mit Making ist der Begriff *Tinkering*. Es gibt zahllose Definitionen für Tinkering, besonders treffend erscheinen hier aber die Beschreibungen Experimentieren,

¹ Online verfügbar unter <https://www.csunplugged.org/de/> (abgerufen am 01.08.2023)

Lernen durch Fehler (ohne negative Konsequenzen), Innovation durch Änderungen, Auseinanderbauen oder ähnliche Erklärungen.

„Tinkering is about hands-on experiences, learning from failures, and unstructured time to explore and invent. And through the processes of exploration and invention lies the potential for innovation.” (RACHELLE, 2012)

Zusammenfassend sind Tinkering und Making (bzw. Maker Education) wichtige Instrumente, um informatische Inhalte besser verstehen zu können. Dabei wird Lernerfahrung nicht primär über Lehrpersonen vermittelt, sondern über das eigene Lernen durch aktives Tun (*Learning by Doing*).

2.4 Physical Computing

Physical Computing beschreibt im schulischen Bereich die Beschäftigung der Schüler*innen mit technischen Systemen, die eine Beziehung zwischen dem Menschen und der digitalen Welt ermöglichen (meist eine IoT-Anwendung – siehe Kapitel 2.5). Dabei kann es sich um künstlerische Bereiche handeln oder aber um ein Projekt nach dem Motto *Do It Yourself* (DIY). Oft werden Sensoren und Mikrocontroller (oder Mikroprozessoren) verwendet, um Schaltkreise zu bauen, anzusteuern oder eigene Funktionen zu programmieren. Dies erfolgt jedoch nicht nach einem strengen Frontal-Unterrichts-Ansatz, sondern mit Hilfe von kreativen Ideen der Schüler*innen selbst. Besonders gefragt sind hier Making und Tinkering (siehe Kapitel 2.3), also ein individuelles, exploratives Lernen, das auch Spaß machen sollte.

“Physical computing involves interacting with the physical world in addition to a screen. You take in data from light, motion, or temperature sensors, and control devices such as motors, speakers, and lights. Physical computing is fun, engaging, and makes learning to code easier and more understandable. Two well-supported and easily found platforms are Arduino and micro:bit. Both are great for students and adults to learn on and make interactive electronic projects.” (MIT, 2024)

Auch HODGES, SENTANCE, FINNEY und BALL (2020) betonen die Wichtigkeit von Physical Computing im Informatikunterricht. Als besonders wertvoll wird erachtet, dass die Motivation der Schüler*innen offensichtlich zunimmt. Auf der einen Seite sind Lernerfolge und Ergebnisse durch die praktische Nutzung des Geräts sofort sichtbar, auf der

anderen Seite gibt es als Belohnung immer auch einen praktischen Nutzen des Ergebnisses (eine neue Funktion). Auch das „Lernen durch Tun“ animiert im Vergleich zur virtuellen Ergebnisfindung. Ebenso spannend ist das Erkennen, dass es oft unterschiedliche Lösungswege gibt oder dass auch Versuch und Irrtum wichtig für den Erkenntnisgewinn sind. Oder das Arbeiten in Gruppen, das ungemein motivierend sein kann. Ganz wichtig ist auch das Erkennen, dass es beim Physical Computing nicht um spezialisiertes Einzelwissen geht, sondern dass Schüler*innen immer mehrere Einzeldisziplinen wie etwa Schaltungsaufbau, Elektronik oder Programmierung beherrschen sollten.

Das Konzept von Physical Computing scheint vor allem Mädchen besonders zu tangieren. Eine Umfrage der *British Broadcasting Corporation* (BBC) unter Siebtklässlerinnen hat nach dem Gebrauch eines Micro:bits dazu geführt, dass 70 % der befragten Mädchen sich geneigt zeigten, in Zukunft weiterführende, informatische Studienrichtungen zu belegen (HODGES, SENTANCE, FINNEY & BALL, 2020, S. 3).

2.5 Internet of Things

Der Datenbank-Spezialist Oracle definiert das *Internet of Things* (IoT) wie folgt:

„Das Internet of Things (IoT) ist die Bezeichnung für das Netzwerk physischer Objekte (,Things'), die mit Sensoren, Software und anderer Technologie ausgestattet sind, um diese mit anderen Geräten und Systemen über das Internet zu vernetzen, sodass zwischen den Objekten Daten ausgetauscht werden können.“ (ORACLE, 2023)

Tatsächlich ist die Bedeutung von IoT bahnbrechend. Bereits 2008 wurde IoT vom US National Intelligence Council (2008) als eine von sechs bahnbrechenden Technologien mit nachhaltiger Beeinflussung eingestuft. Die besondere Bedeutung liege an der Tatsache, dass diese Technologie tatsächlich vom Berufsleben bis in den privaten Bereich der Konsument*innen vordringe. Die Folge ist eine einzigartige Vernetzung von mobilen Geräten und Sensoren mit enormen Potentialen, aber auch nicht vorhersehbaren Sicherheitsrisiken (THE NATIONAL INTELLIGENCE COUNCIL, 2008, S. 27–30).

3. Grundlagen der Digitaltechnik und Technischen Informatik

Da das Forschungsziel dieser Masterthesis die Visualisierung von digitalen Grundlagen mit Hilfe eines Mikrocontrollers ist, soll an dieser Stelle definiert werden, welche Themengebiete davon betroffen sind. Natürlich ist es nicht möglich, alle relevanten Fachgebiete wie etwa Elektronik, Schaltungstechnik, Automatisierung, technische Mathematik, Physik oder gar Detailwissen über sämtliche Vorgänge eines Computers zu erfassen. Aus diesem Grund orientiert sich diese Masterthesis zu großen Teilen am logischen Aufbau des Lehrwerks *„Eine kurze Geschichte der Technischen Informatik – Wie Computer funktionieren – für alle verständlich erklärt“* (REICHL, 2023). Dieses Kapitel beschreibt analog zum genannten Lehrwerk die wichtigsten Grundlagen der Digitaltechnik mit einem Fokus auf den Praxisprojekten im Kapitel 6. Aufbauend auf diesen Grundlagen werden somit zur besseren Visualisierung zehn praktische Projekte mit einem Mikrocontroller erstellt (siehe Kapitel 6). Diese Referenzprojekte erweitern und vertiefen den technisch-theoretischen Hintergrund durch den Entwurf von einfachen Schaltungen, den Aufbau und Anschluss der elektronischen Komponenten, meist mittels eines Steckbretts, und schlussendlich durch das Programmieren einer gewünschten Funktion am Mikrocontroller. Dabei soll aber sowohl den Lehrpersonen als auch den Schüler*innen genügend Raum für Making und Tinkering (siehe Kapitel 2.3) eingeräumt werden.

3.1 Informationen

Es gibt keine einheitliche Definition des Begriffs „Information“. Gemäß Lipp & Becker (2011) sind Informationen Daten, die im Prinzip aus Zeichen oder kontinuierlichen Funktionen bestehen. Zeichen wiederum sind Symbole wie etwa Buchstaben, Ziffern, Morsezeichen etc. Aufgrund von definierten Regeln („Abmachungen“) stellen diese Daten bzw. Zeichen Informationen dar, die weiterverarbeitet werden oder die bereits ein Ergebnis darstellen. Informationen sind somit Träger von realen oder gedanklichen Sachverhalten oder Vorgängen (LIPP & BECKER, 2011, S. 23–24). Reichl (2023) deutet die Information als Beschreiben der Welt mit Symbolen. Diese Symbole sind etwa Zahlen, mit deren Hilfe die Welt seit jeher sehr gut zu beschreiben war. Der Begriff „digital“ leitet

sich direkt vom lateinischen *digitus* (der Finger) ab (REICHL, 2023, S. 3). „*Digital rechnen heißt daher eigentlich nichts anderes als ,Rechnen wie mit Fingern‘*“ (REICHL, 2023, S. 3).

3.2 Analoge und digitale Informationen

Vor allem in der Technik handelt es sich bei Informationen um physikalische Größen. Hier spricht man nicht von Informationen, sondern von Signalen. Signale sind „*die Darstellung von Nachrichten oder Daten mit physikalischen Mitteln*“ (LIPP & BECKER, 2011, S. 25). Diese Signale verändern sich durch Zeit und Raum nach bestimmten Regeln und ändern Informationen ihrerseits unter kontrollierten Umständen. Abbildung 2 zeigt einige physikalische Größen:

Physikalische Größe	Veränderbarer Parameter
Spannung, Strom	Amplitude, Kurvenform, Frequenz, Phasenlage
Widerstand	Widerstandswert
Feldstärke	Betrag, Richtung
Reflexionsfähigkeit	Reflexionsfaktor
Durchlässigkeit	Transmissionsfaktor

Abbildung 2: Informationen als physikalische Größe mit möglicher Änderung der Parameter (LIPP & BECKER, 2011, S. 25)

Entscheidend ist die Tatsache, dass physikalische Größen in der Natur immer als analoge Werte vorliegen. Diese Werte sind kontinuierlich, sie haben also zwischen dem Mindest- und dem Maximalwert unendlich viele Zwischenwerte. Bei analogen Werten können Überlappungen geschehen. Es ist auch stets möglich, dass Messfehler oder Unschärfen bei Messungen auftreten. Im Gegensatz dazu sind digitale Werte stets diskret. Die Werte halten also einen klar definierten Abstand zueinander ein (REICHL, 2023, S. 5). Dies wird durch eine klare Definition von Intervallen zwischen Mindest- und Maximalwert erreicht. „*Ein Digitalwert steht dann jeweils für ein Intervall*“ (LIPP & BECKER, 2011, S. 27). Damit ergibt sich allerdings das Problem der exakten Intervallgrenze. Fallen Werte genau in diesen Übergangsbereich, müsste hier ganz genau gemessen werden. Stattdessen

wird an diesen Grenzen mit Absicht ein „undefinierter Zustand“ eingebaut, der entweder eine Fehlermeldung generiert oder eine willkürliche Zuordnung zu einem der benachbarten Intervalle durchführt („Weiche“ Diskrimination an den Intervallgrenzen) (LIPP & BECKER, 2011, S. 27).

3.3 Binäre Darstellung und Stellenwerte

Wichtig ist ein grundlegendes Verständnis für eine generelle Zahlendarstellung. Dezimale Zahlen könnten beispielsweise auch mit Potenzen zur Basis 10 dargestellt werden, wobei der sogenannte „Stellenwert“ von entscheidender Bedeutung ist. Abbildung 3 zeigt die Darstellung der Zahl 583 als Summe der Stellenwerte:

Hunderter		Zehner		Einerstelle
500	+	80	+	3
$5 \cdot 100$	+	$8 \cdot 10$	+	$3 \cdot 1$
$5 \cdot 10^2$	+	$8 \cdot 10^1$	+	$3 \cdot 10^0$

Abbildung 3: Darstellung der Zahl 583 durch Potenzen zur Basis 10 (BERNSTEIN, 2019, S. 2)

Dies gilt nicht nur für die dezimale Darstellung von Zahlen. Auch eine duale Darstellung arbeitet etwa mit 2^{er} -Potenzen. Die technische Umsetzung des Dualsystems (Potenzen zur Basis 2) erfolgt technisch über das sogenannte Bit. „*Das Bit ist die Einheit für ein binäres (zweiwertiges) Signal*“ (BERNSTEIN, 2019, S. 7–8). Ein Binary Digit (BIT) gilt als kleinste Informationseinheit und hat den Wert 0 oder 1. Diese beiden Werte können Zustände darstellen, wie etwa *Schalter offen/geschlossen, Strom fließt/fließt nicht, Transistor leitend/gesperrt* etc. (BERNSTEIN, 2019, S. 7–8). Erhöht man die Anzahl der Bits, erhöht man zugleich die Möglichkeiten an Zuständen. Fasst man vier Bits zusammen, spricht man von einer *Tetrade* oder meist von einem *Nibble*. Ein Nibble kann insgesamt 2^4 Zustände (in Summe 16) abbilden. Zwei Nibble (8 Bit) entsprechen einem Byte und können somit 2^8 Zustände (256) abbilden. Das linksstehende Bit wird dabei *Most Significant Bit* (MSB) genannt und stellt den höchsten Wert dar (PLATT, 2022, S. 241). Das

rechtsstehende Bit ist ein *Least Significant Bit* (LSB) und repräsentiert den niedersten Wert. Die folgende Abbildung 4 zeigt die Stellenwerte eines Nibble:

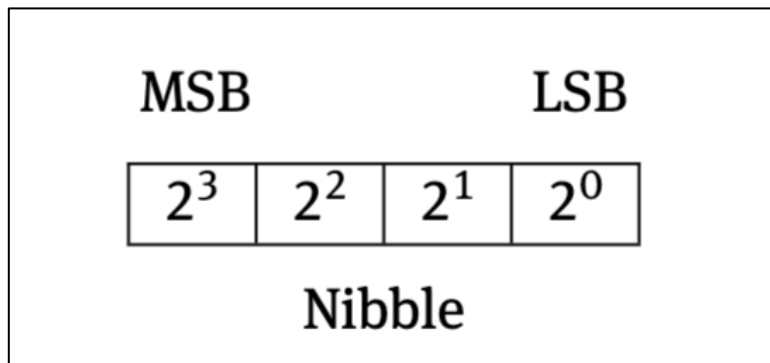


Abbildung 4: Darstellung eines Nibble, links das MSB, recht das LSB (BERNSTEIN, 2019, S. 8)

Die Anordnung der Bits als Stellenwerte ist wesentlich für die Funktion digitaler Systeme. Sie ermöglicht es, Daten effizient zu speichern bzw. zu verarbeiten, indem jedes Bit einen spezifischen Stellenwert hat, der in Registern und Speicherzellen genutzt wird.

3.4 Digitalisierung von Signalen, Pulsweiten-Modulation

Die Digitalisierung ist der theoretische Prozess der Umwandlung von kontinuierlichen Signalen bzw. Informationen in diskrete Werte (eindeutige Werte) samt binärer Darstellung. Ein *Analog-Digital-Converter* (ADC) führt als Analog-Digital-Wandler diesen Prozess in der Praxis durch. Ein Digital-Analog-Wandler (DAC) kehrt den Diskretisierungsprozess um, indem er diskrete Werte in ein kontinuierliches Signal umwandelt. Diskrete Werte sind eindeutig abgegrenzte und zählbare Zustände, die von einem Signal oder System angenommen werden können (siehe Kapitel 3.2). Diese Werte treten innerhalb definierter Stufen auf und sind nicht kontinuierlich. Die folgende Tabelle 1 erklärt nochmals die wichtigsten Begriffe der Informationsdarstellung, da diese oft falsch bzw. nicht eindeutig verwendet werden und für das Verständnis der Diskretisierung essenziell sind.

Tabelle 1: Erklärung wichtiger Begriffe der Informationsdarstellung (eigene Tabelle)

Fachbegriff	Erklärung
Dual	Dual ist die Darstellung von Zahlen zur Basis 2 unter Verwendung von Potenzen von 2. Dies ist ein vorwiegend mathematischer Begriff.
Diskret	Diskrete Werte sind eindeutige, gesonderte Werte, die nicht kontinuierlich sind. Diese Werte können sowohl digital als auch analog sein, solange sie in klar abgegrenzten Stufen erkennbar sind (ein Beispiel für einen analog-diskreten Wert wäre ein analoges Thermometer, das nur eine ganze Zahl ohne Zwischenwert anzeigen kann, wie etwa 0° Grad, 1° Grad ... 100° Grad).
Binär	Binär bezeichnet die Verwendung des dualen Systems zur Darstellung diskreter Werte als Bits, wobei jedes Bit entweder den Wert 1 oder 0 hat. Die binäre Darstellung als Bit ist die Basis der digitalen Elektronik und Informatik.

Bei der Digitalisierung wird das analoge Signal durch einen ADC in fest vorgegebene Bereiche geteilt und dann als diskreter, binärer Wert dargestellt. Dabei unterscheidet man zwischen den folgenden Phasen:

- Diskretisierung (Abtastung)

Hierbei wird das Quellsignal als kontinuierliches Signal zu festen Zeitpunkten abgetastet (meist pro Sekunde). Die sogenannte *Sampling Rate* bestimmt die Anzahl der Messvorgänge pro Sekunde, beispielsweise bei einem Audiosignal. Eine mögliche Sampling-Rate wäre hier 44.100 Kilohertz pro Sekunde (KERSKEN, 2016, S. 55).

- Quantisierung

Die abgetasteten analogen Werte werden in diskrete Werte gewandelt, die durch eine begrenzte Anzahl von Bits dargestellt werden.

- Codierung

Die quantisierten Werte werden in eine binäre Zahl umgewandelt, die das digitale Signal repräsentiert. Die Anzahl der Bits bestimmt die Auflösung und damit auch die Genauigkeit der Darstellung.

Ein DAC macht diesen Vorgang wieder rückgängig, indem das digitale, binäre Signal wieder in ein analoges, kontinuierliches gewandelt wird. Die folgende Abbildung 5 zeigt diesen Ablauf:

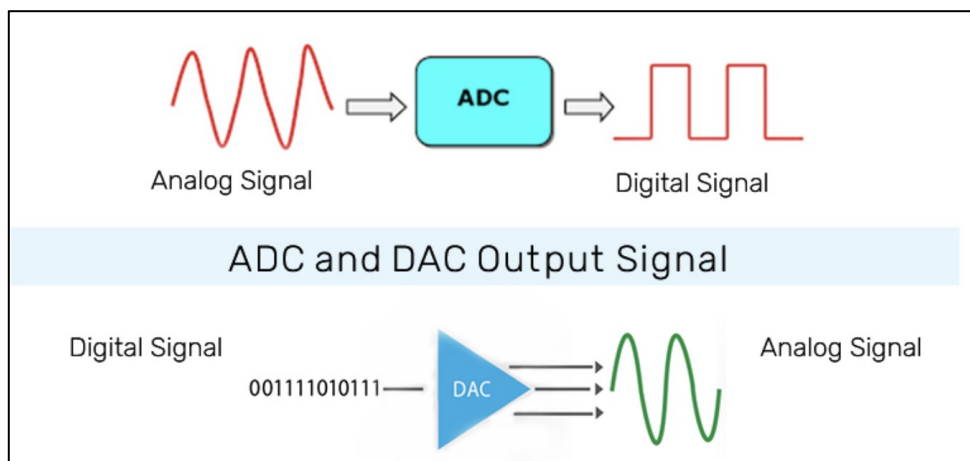


Abbildung 5: Wandlung und Rückwandlung von analogen Signalen (CLARK, 2022)

Murphy (2022) definiert als bekanntes Beispiel einer Digitalisierung einen Telefonanruf über ein Mobiltelefon. Die Stimme des Absenders ist ein analoges Audiosignal, das durch einen im Smartphone integrierten ADC in eine binäre Form gebracht wird. Das binäre Signal wird über Mobilfunkstationen und Satelliten digital an Empfänger*innen des Telefonats geleitet. Hier wird das digitale Signal mit Hilfe eines DAC wieder in ein analoges Signal gewandelt (MURPHY, 2022).

Keine Form der Digitalisierung ist die sogenannte „Pulsweiten-Modulation“ (PWM). PWM ist eine Technik, die digitale Signale verwendet, um analoge Effekte zu erzeugen. Sie kann in Systemen wie Mikrocontrollern eingesetzt werden, die digitale Signale verwenden, um analoge Ausgaben zu simulieren. Dabei wird gemäß Elektronik-Kompandium (2024) durch wechselnde Schaltung von Impulsdauer (repräsentiert eine „1“) und Pausendauer (repräsentiert eine „0“) der Effekt einer Spannungsänderung erzeugt. Bei

gleichbleibender Periodendauer entsteht hier beispielsweise der Eindruck eines heller werdenden Lichtes durch scheinbare Änderung der kontinuierlichen Stromspannung, was aber tatsächlich nur digital bewirkt wird (ELEKTRONIK-KOMPENDIUM, 2024a), (KAMPERT & SCHERBECK, 2017, S. 98). Die folgende Abbildung 6 zeigt die Bestandteile von PWM:

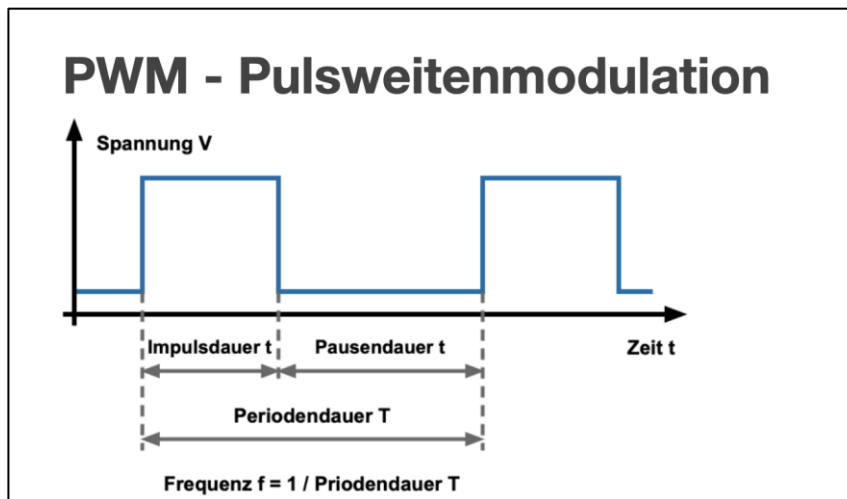


Abbildung 6: Darstellung PWM (ELEKTRONIK-KOMPENDIUM, 2024a)

3.5 Boolesche Logik, Logik-Gatter, Integrated Circuits (ICs)

Der englische Mathematiker George Boole stellte sich Mitte des 19. Jahrhunderts die Frage, ob man nur mit Zahlen oder auch mit Wahrheitswerten rechnen kann. Er wollte die Welt auf eine Reihe von wahren und falschen Aussagen reduzieren, deren Ergebnisse sich aber wiederum beliebig weiter verknüpfen lassen sollten. Die Grundidee laut Reichl (2024) sind Wahrheitswerte, Verknüpfungen und Operatoren. Die Wahrheitswerte beschäftigen sich mit Aussagen („es scheint die Sonne“), die entweder wahr oder falsch sein können. Wichtig ist jedoch, dass der tatsächliche Wahrheitsgehalt wirklich bekannt sein muss („es ist wahr, die Sonne scheint“). Jetzt kann diese Aussage mit einer oder mehreren anderen Aussagen verknüpft werden, deren einzelner Wahrheitsgehalt wiederum bekannt sein muss. Wenn also der Wahrheitsgehalt beider Ausgangsaussagen bekannt ist und beide Aussagen verknüpft sind, kann aufgrund logischer Rechengesetze tatsächlich das Endergebnis der Verknüpfung abgeleitet werden. Die Verknüpfung von Aussagen erfolgt mit logischen Operatoren wie AND, OR, NOT, XOR, NOR, NAND etc., die Ergebnisdarstellung visualisiert eine sogenannte „Wahrheitstabelle“. Die folgende

Abbildung 7 zeigt die gängigsten logischen Operatoren:

Tab. 2 Wahrheitstabelle für Boolesche Operatoren							
BOOLESCHE GRUNDOPERATOREN						XOR	NOR
A	B	$\neg A$	$\neg B$	$A \wedge B$	$A \vee B$	$A \oplus B$	$\neg(A \vee B)$
0	0	1	1	0	0	0	1
0	1	1	0	0	1	1	0
1	0	0	1	0	1	1	0
1	1	0	0	1	1	0	0

Abbildung 7: Wahrheitstabelle samt logischer Operatoren (REICHL, 2023, S. 12)

So entstand die nach George Boole benannte Boolesche Algebra mit genormten Rechenvorschriften, für die es aber dazumal keine passende Anwendung gab. Das änderte sich erst, als Claude Shannon die Arbeit von George Boole in den 1930er Jahren entdeckte und diese Erkenntnisse mit der Handhabung von ersten Relaisschaltungen für das entstehende Telefonnetz in Verbindung brachte. Später fand die Boolesche Algebra direkte Anwendung bei Vakuumröhren sowie Transistoren und schlussendlich natürlich als Grundlage der Digitaltechnik mittels Mikroschaltungen als *Integrated Circuits* (IC) (PLATT, 2022, S. 211). Speziell Transistoren sind hier wichtige Bausteine. Sie können als technische Umsetzung der Binärlogik gesehen werden, denn sie repräsentieren physisch gesehen die Zustände „1“ und „0“. Kombiniert bzw. verknüpft man Transistoren untereinander oder mit anderen elektronischen Bauteilen, können damit boolesche Verknüpfungen über logische Schaltungen realisiert werden. Tausende bzw. Milliarden an winzigen Transistoren befinden sich auf einem einzigen IC, der selbst zum Großteil aus Silizium besteht (MONK, 2016, S. 74). Ein einzelner, kleiner integrierter Schaltkreis wäre beispielsweise ein digitales Logik-Gatter, das mehrere logische Schaltungen in sich vereint. Ein großes IC ist die CPU, die aber natürlich mit anderen ICs wie RAM oder GPU (Graphics Processing Unit) zusammenarbeitet.

„Sobald wir mit Hilfe elektronischer Schaltungen Boolesche Algebra betreiben können, beherrschen wir auch die komplette Arithmetik“ (REICHL, 2023, S. 27). Setzt man sich die Umsetzung einen binären Halbaddierers zur Aufgabe, kann die folgende Herangehensweise als stellvertretend für die Herausforderung „vom analogen Problem zur binären Schaltung als Lösung“ gesehen werden. Im ersten Schritt wird die Wahrheitstabelle

eines einfachen Halbaddierers erstellt, die das Ergebnis der Addition von binären Werten darstellt (HOFFMANN, 2023, S. 218–219). Es ergeben sich zwei Eingaben mit „a“ und „b“ bzw. zwei Ausgaben mit „Übertragsbit“ und „Summenbit“, wie die folgende Abbildung 8 darstellt:

Tab. 1 Additionstafel für zwei einstellige Binärzahlen				
BINÄRE ADDITION: a + b				ZUM VERGLEICH
a	b	Übertragsbit	Summenbit	dezimal
0	0	0	0	$0 + 0 = 0$
0	1	0	1	$0 + 1 = 1$
1	0	0	1	$1 + 0 = 1$
1	1	1	0	$1 + 1 = 2$

Abbildung 8: Wahrheitswerttabelle der binären Addition (REICHL, 2023, S. 8)

Im zweiten Schritt werden die Ausgänge Übertragsbit und Summenbit als Gatterschaltung dargestellt. Das Übertragsbit kann durch eine AND-Schaltung realisiert werden, das Summenbit durch eine XOR-Schaltung. Das Ergebnis zeigt die Abbildung 9:

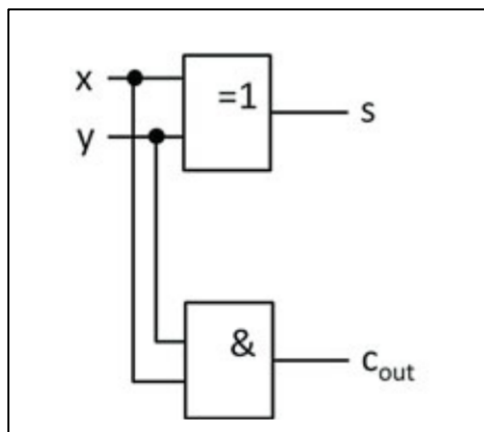


Abbildung 9: Ein Halbaddierer als logisches Gatter (REICHL, 2023)

Der tatsächliche Nachbau des Halbaddierers könnte mit Hilfe eines Micro:bits, eines Steckbretts, Vorwiderständen, LEDs und zwei physischen Gattern (ICs) als AND- und XOR-Gatter realisiert werden. Die folgende Abbildung 10 zeigt den virtuellen Schaltungsentwurf mit TinkerCAD:

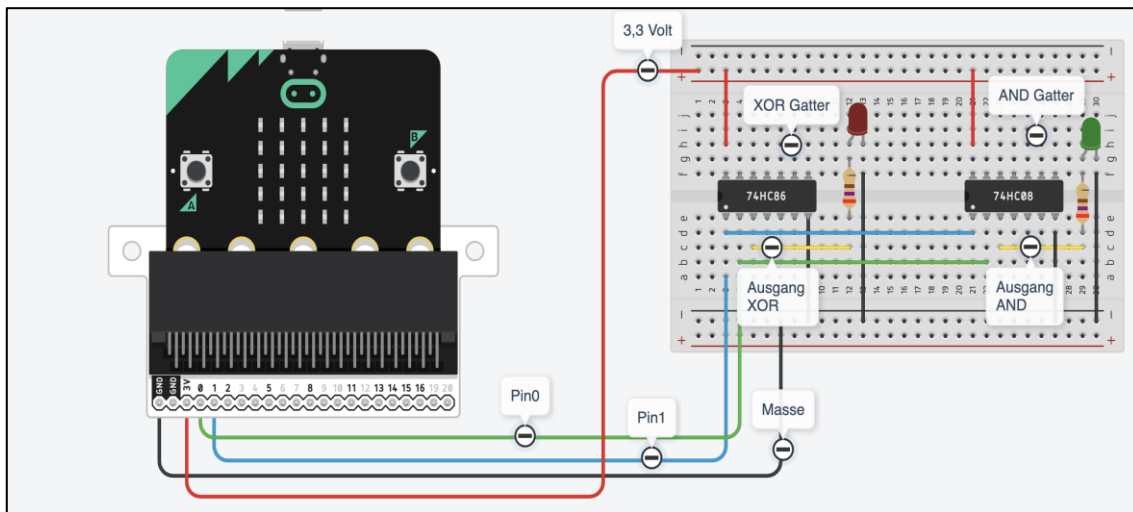


Abbildung 10: Schaltungsentwurf mit Pin 0, Pin 1, XOR- und AND-Gate (eigene Abbildung)

Eine mögliche Programmierung wäre etwa eine Funktion, mit den Buttons A und B des Micro:bits zwei gesonderte Eingänge der ICs (AND-Gatter und XOR-Gatter) auf „1“ bzw. „0“ zu schalten. Durch die integrierte, logische XOR- bzw. AND-Verknüpfung reagieren die Ausgänge der ICs gemäß der Wahrheitswertetabelle (siehe Abbildung 8). Entsteht im Ausgang des XOR-Gatters eine „1“, leuchtet die rote LED, entsteht im Ausgang des AND-Gatters eine „1“, leuchtet die grüne LED. In den anderen Fällen bleiben die LEDs dunkel. Die folgende Abbildung 11 zeigt die Programmierung mit MicroPython, die Abbildung 12 repräsentiert einen tatsächlichen Nachbau auf einem Steckbrett samt Micro:bit:


```
1  from microbit import *
2  # Variablen myPin1/myPin2 auf 0 setzen
3  myPin1 = 0
4  myPin2 = 0
5  # Anschlüsse p1 und p2 auf 0 setzen
6  pin1.write_digital(0)
7  pin2.write_digital(0)
8  display.show(Image.HAPPY)
9  while True:
10     # Aus-/Einschalten von p1
11     if(button_a.is_pressed()):
12         if(myPin1 == 0):
13             myPin1 = 1
14             # Hinweis für XOR-Gatter
15             display.show("P1:1")
16         else:
17             myPin1 = 0
18             # Hinweis für XOR-Gatter
19             display.show("P1:0")
20             pin1.write_digital(myPin1)
21     # Aus-/Einschalten von p2
22     if(button_b.is_pressed()):
23         if(myPin2 == 0):
24             myPin2 = 1
25             # Hinweis für AND-Gatter
26             display.show("P2:1")
27         else:
28             myPin2 = 0
29             # Hinweis für AND-Gatter
30             display.show("P2:0")
31             pin2.write_digital(myPin2)
```

Abbildung 11: Programmierung eines Halbadddierers mit MicroPython (eigene Abbildung)

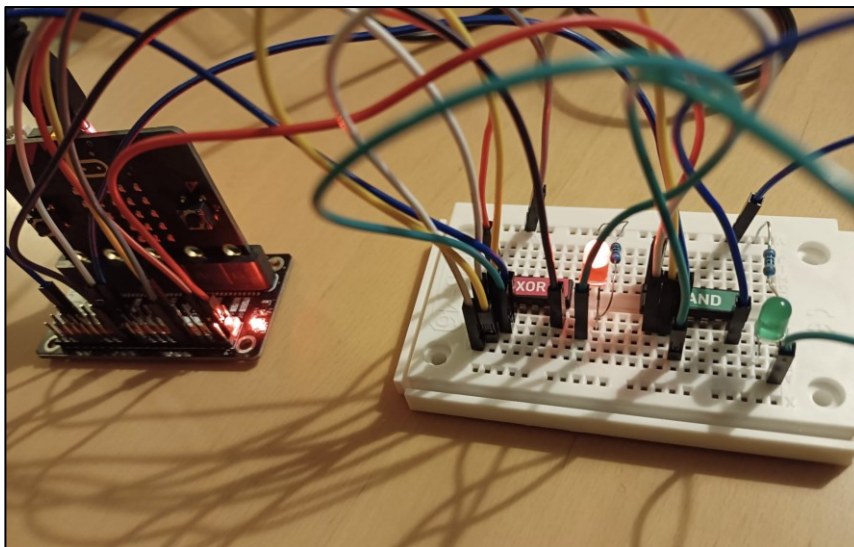


Abbildung 12: Mögliche Halbleiterschaltung (eigene Abbildung, eigener Aufbau)

3.6 Taktgeber und Timer

Zwei relevante Komponenten im Bereich Mikrocontroller sind die Systemtaktung (System Clock) und die CPU-Frequenz. Wichtig ist, dass, obwohl meist der CPU-Takt genannt

wird, der Systemtakt die wesentlich größere Rolle spielt. Abgeleitet vom Systemtakt werden beispielsweise der CPU-Takt, aber auch die Taktgeschwindigkeiten anderer Peripherie-Einheiten oder Daten-Übertragungsschnittstellen. Meist ist der CPU-Takt ident mit dem Systemtakt. Der CPU-Takt könnte aber nie höher sein als der generelle Systemtakt. Dies kann sehr gut am Mikrocontroller Micro:bit V2 eingesehen werden. Gemäß der Dokumentation des verbauten Controllers nRF52833 beträgt der Systemtakt 64 MHz (NORDIC SEMICONDUCTOR, 2024, S. 79), ebenso wie der CPU-Takt (NORDIC SEMICONDUCTOR, 2024, S. 19). Dies bedeutet, dass die CPU pro Sekunde 64 Millionen Takte zur Verfügung hat. Innerhalb eines Taktes könnte beispielsweise eine einfache Addieraufgabe gelöst werden, Aufgaben wie Speichern würden jedoch mehrere Takte benötigen. Die Dauer eines einzelnen Taktes der CPU des Micro:bit wäre 15,625 Nanosekunden.

$$\frac{1}{(64 \times 10^6)} = 15,625 \text{ Nanosekunden (ns)}$$

Ebenfalls interessant ist die Frage, wie Mikrocontroller Zeitmessungen machen bzw. Uhrzeit und Datum bereitstellen. Hier greift ein Mikrocontroller wie der nRF52833 auf eingebaute Timer zu, die wiederum den Systemtakt verwenden, um Zeitintervalle zu messen. Timer zählen Taktzyklen, die vom Systemtakt bereitgestellt werden, und können so präzise Zeitmessungen durchführen. Beim Micro:bit gibt es zusätzlich noch ein Modul namens *Real-Time-Counter* (RTC). Dieses Modul verwendet einen 32,768-KHz-Quarz, läuft unabhängig vom Hauptsystemtakt und ermöglicht gesondert genaue Zeit- und Datumsangaben. Die Abbildung 13 zeigt den Aufbau des RTCs:

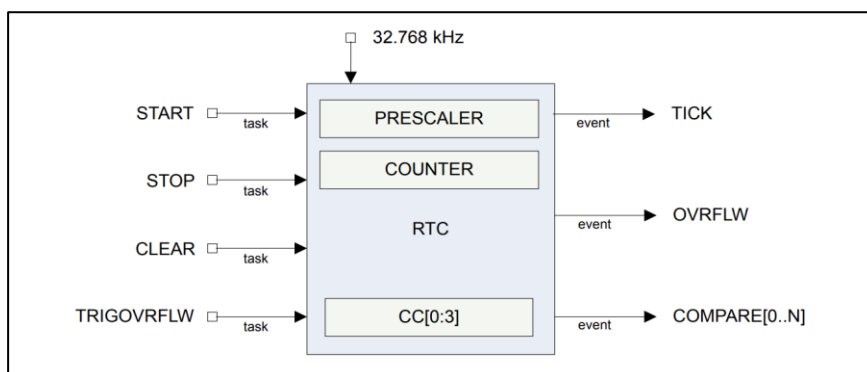


Abbildung 13: Aufbau eines RTCs (NORDIC SEMICONDUCTOR, 2024, S. 343)

3.7 Kommunikationsschnittstellen

„Kommunikationsprotokolle sind formale Beschreibungen von digitalen Nachrichten-formaten [sic] und -regeln[sic]. Sie sind erforderlich, um Nachrichten in oder zwischen Computersystemen auszutauschen.“ (ROUSE, 2023)

Vor allem bei Mikrocontrollern sind die kabelgebundenen Protokolle *Universal Asynchronous Receiver/Transmitter* (UART), *Serial Peripheral Interface* (SPI) und *Inter-Integrated-Circuit* (I2C) sehr wichtig und sollen näher vorgestellt werden. Anstatt der bekannten kabellosen Technologien wie WiFi und Bluetooth soll speziell das proprietäre Radio-Protokoll des Micro:bit erläutert werden.

3.7.1 Universal Asynchronous Receiver/Transmitter (UART)

UART ist ein asynchroner Kommunikationskanal, der ohne gesonderten Takt auskommt. Die Rolle des Takts übernehmen sogenannte Start- und Stopp-Bits. Die Übertragungsrate (Bitrate) ist nicht allzu hoch, da die Daten Bit für Bit seriell übertragen werden. Die UART-Schnittstelle besteht aus den erwähnten Start- und Stopp-Bits, den binären Daten selbst und eventuell noch aus Paritätsbits zur Fehlerüberprüfung. Das Start-Bit ist grundsätzlich auf „0“ (*low*) gestellt, das Stopp-Bit auf „1“ (*high*). Damit beginnt ein einziges Datenframe im IDLE-Zustand „1“ (*high*). Daten werden gebündelt in einzelnen Datenframes übertragen. Für Daten selbst ohne Start-, Stopp- oder Paritäts-Bit sind fünf bis maximal neun Bits vorgesehen. Die Übertragung erfolgt über eine umgekehrte Bitfolge mittels LSB, wie folgende Abbildung 14 zeigt:

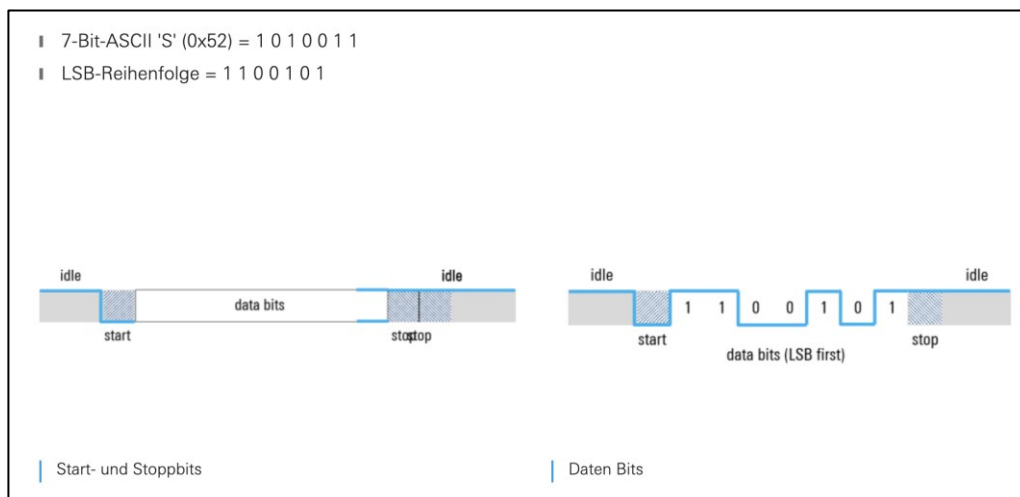


Abbildung 14: Übertragung des Buchstaben S (ASCII-Code dezimal 83) (ROHDE & SCHWARZ GMBH & CO KG, 2023)

Das Empfangsgerät (*RX – Receive*) und das Sendegerät (*TX – Transfer*) müssen sich auf eine gemeinsame Baudrate einigen. Üblich sind hier 4800, 9600, 19,2 Kilobaud (kBd), 57,6 kBd und 115,2 kBd. Die Baudrate ist die Symbolrate der übertragenen Daten. Die Bitrate ist in der Regel höher als die Baudrate (Beispiel: 4800 Baud mit vier Bits für ein Symbol wären 19.200 Bitrate) (KAMPERT & SCHERBECK, 2017, S. 159–160). *TX* und *RX* beider Kommunikationseinheiten müssen einen gesonderten Ground-Anschluss haben und werden zudem gegengleich verkabelt (*TX* zu *RX*, *RX* zu *TX*), siehe folgende Abbildung 15:

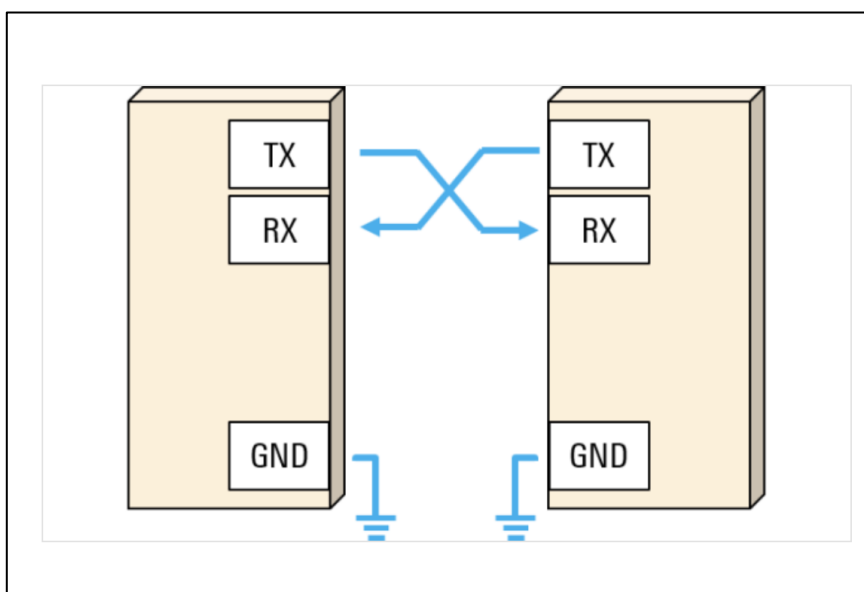


Abbildung 15: Sende- und Empfangskomponenten TX und RX (ROHDE & SCHWARZ GMBH & CO KG, 2023)

3.7.2 Serial Peripheral Interface (SPI)

Auch SPI beruht auf einer seriellen, synchronen Übertragung mit einem Taktsignal. Die Kommunikation besteht aus mindestens einer Master- und mindestens einer Slave-Komponente. Es können aber auch mehrere Slave-Geräte angeschlossen werden. Die Übertragung beginnt immer mit der Taktfrequenz des Masters. Die vier Datenleitungen heißen *Serial Clock (SCLK)*, *Master Out Slave In (MOSI)*, *Master In Slave Out (MISO)* und *Slave-Select (SS)* oder *Chip-Select (CS)*. Die Übertragung selbst ist im Duplexmodus möglich. Es existieren Start- und Stopp-Bits, allerdings fehlen die Paritätsbits zur Fehlerkontrolle. Die folgende Abbildung 16 zeigt ein Mastergerät und drei Slave-Geräte. Die Datenleitungen sind MOSI und MISO, wobei die MISO-Leitung des vorherigen Slaves in die MOSI-Leitung des nächsten Slaves führt. Erst das letzte MISO-Signal eines Slaves geht

zum MISO des Masters, was in Abbildung 16 ersichtlich ist:

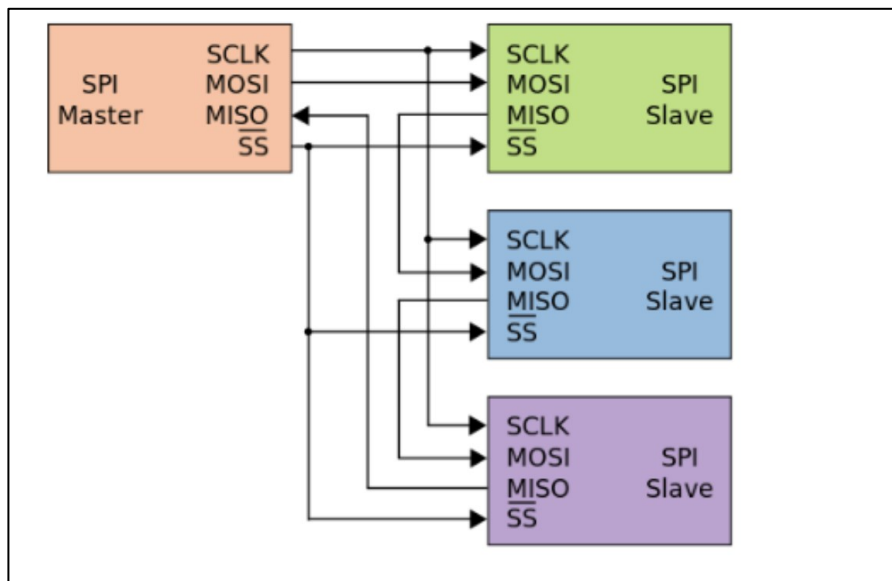


Abbildung 16: Aufbau einer SPI-Verbindung mit 3 Slaves (HQNEXTPCB, 2023)

Über mehrere $\overline{SS}$ s oder $\overline{CS}$ s könnten gesonderte Slaves angesteuert werden. Bei lediglich einem $\overline{SS}$ oder $\overline{CS}$ erkennen und lesen alle Geräte die gesamte Kommunikation (KOFLER, KÜHNAST & SCHERBECK, 2016, S. 443–444). Als Anschlussmöglichkeit für externe Geräte werden oft Komponenten wie SPI-Treiber oder Adapter eingesetzt und so mit dem Mikrocontroller verbunden.

3.7.3 Inter-Integrated-Circuit (IC2)

Im Gegensatz zu SPI benötigt IC2 nur zwei Leitungen, die *Serial Data Line* (SDA) für die Übertragung der Daten und eine Leitung für den Takt mit *Serial Clock* (SCL). Insgesamt können theoretisch 128 Geräte im gleichen Bussystem adressiert werden, praktisch jedoch nur 112 Geräte. Diese könnten sowohl Master als auch Slave sein. Die Abbildung 17 zeigt die beiden Komponenten SDA und SCL:

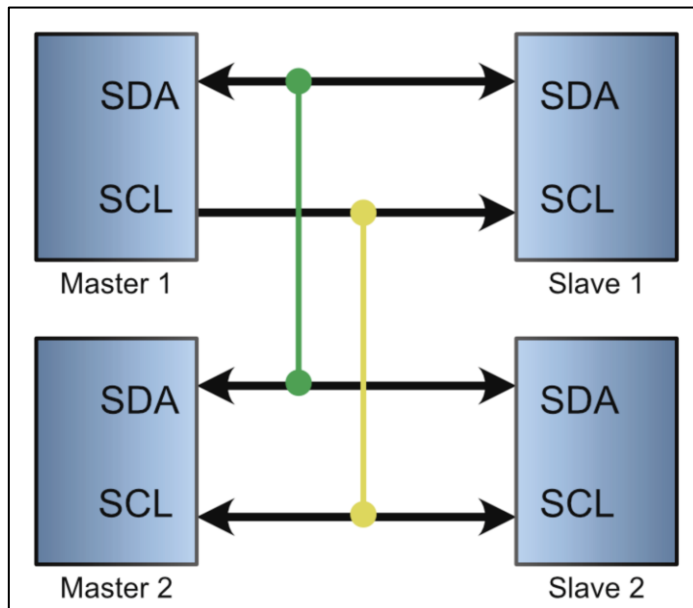


Abbildung 17: Grundaufbau einer I2C-Kommunikation mit vier Geräten (EDLINGER, 2020)

Die Kommunikation erfolgt durch Start- und Stopp-Bits. Die gesendeten Daten sind immer ein Byte groß und werden nach erfolgreicher Sendung mit einem *Acknowledgment* (ACK) bestätigt bzw. mit einem *Negative Acknowledgment* (NACK) nicht bestätigt. Slave, aber auch Master, können über gesonderte Register gelesen oder beschrieben werden. Interessant ist auch die Tatsache, dass die Daten genau in der Mitte eines jeden Clock-HIGH-Signals gelesen werden, damit das Datensignal genügend Zeit für einen Pegelwechsel zwischen LOW und HIGH hat (EDLINGER, 2020).

3.7.4 Radio-Protokoll

Das Radio-Protokoll wird zusammen mit dem Bluetooth-Protokoll durch ein integriertes 2,4-GHz-Funkmodul zur Verfügung gestellt und ist somit ein drahtloses Funkprotokoll. Laut Lancaster University (2024) ist das Radio-Protokoll ein sehr einfaches im Gegensatz zum aufwendigeren Bluetooth-Standard. Radio sendet an andere Geräte im Modus 1:1 oder 1:n. Allerdings können nicht beide Protokolle (Bluetooth und Radio) zur gleichen Zeit verwendet werden. Radio sendet standardmäßig über die Funkgruppe 0 unverschlüsselt, sodass sämtliche Daten von jedem vorhandenen Micro:bit abgefangen werden können. Allerdings ist nicht direkt ersichtlich, wer genau die Infos verschickt hat. Die folgende Abbildung 18 zeigt die offene Übertragung mit dem Radio-Protokoll:

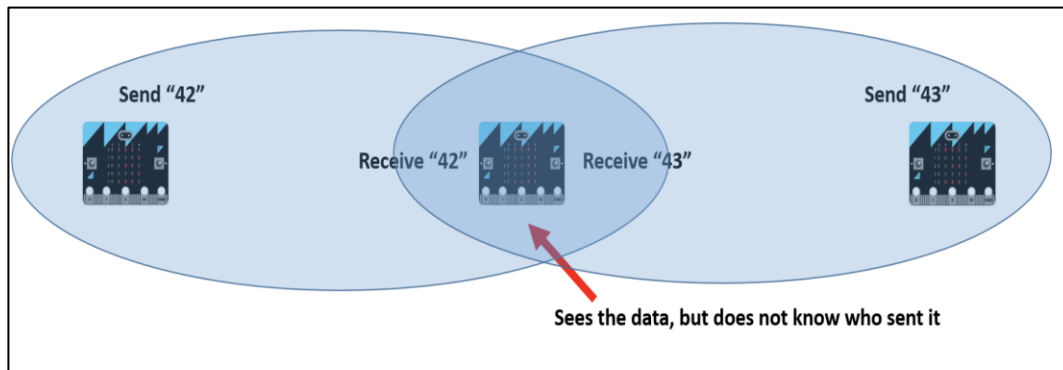


Abbildung 18: Offene Übertragung mit dem Radio-Protokoll (LANCASTER UNIVERSITY, 2024)

Damit die Übertragung auf einen Benutzerkreis eingeschränkt werden kann, werden Gruppen (0 bis 255) und Kanäle (0 bis 80) eingerichtet. So wird der Empfängerkreis im Funkverkehr eingeschränkt. Die Übertragung erfolgt standardmäßig unverschlüsselt, ist also nicht übertragungssicher. Üblicherweise ist eine Übertragungseinheit mit 32 Bytes festgelegt, dies kann aber auf bis zu 1024 Bytes erweitert werden. Der Funkabstand darf nur einige Meter betragen (LANCASTER UNIVERSITY, 2024).

3.8 Elektronische Speicher

Informationen werden über unterschiedliche Speichertechnologien gesichert. Diese Masterthesis fokussiert sich auf rein elektronische Speicher, da diese vorwiegend in Mikrocontrollern vorzufinden sind. Mechanische Speicher wie Festplatten als *Hard-Disk-Drive* (HDD), optische Datenträger (DVD, CD, Blue-ray) und andere werden somit nicht weiter thematisiert. Im Hinblick auf das Thema Digitaltechnik und binäre Schaltungen ist es natürlich auch interessant, binäre Technologien in Speichermedien allgemein zu beschreiben. Logische Schaltungen bzw. deren physikalische Realisierung finden sich in allen elektronischen Speicherarten. Speichertechnologien wie *Dynamic Random Access Memory* (DRAM), *Electrically Erasable Programmable Read Only Memory* (EEPROM) oder *Flash* nutzen logische Gatter hauptsächlich in der Steuer- und Adressierungslogik, nicht jedoch direkt zur Datenspeicherung. *Static Random-Access Memory* (SRAM) und *Register* hingegen nutzen logische Gatter direkt zur Speicherung von Daten. Beide Speicherarten basieren auf der Verwendung von *Flip-Flops*, die aus logischen Gattern wie NOT-, NAND- sowie NOR-Gattern bestehen.

Ein Grundbaustein der digitalen Speicherung ist das *RS-Flip-Flop* (Reset-Set-Flip-Flop). Es

besteht aus zwei NAND-Gattern (oder NOR-Gattern in einer anderen Variante) und hat zwei Eingänge (*Set* und *Reset*), die den Zustand des gespeicherten Bits steuern. Laut Elektronikkompendium (2024) besteht ein SR-Flip-Flop aus zwei NOR-Gattern mitsamt zwei gesonderten Ausgängen Q_1 und Q_2 . Diese beiden Ausgänge sind gegeneinander invertiert. Die Symbole E_1 repräsentieren das Setzen (*Set*) bzw. E_2 das Zurücksetzen (*Reset*) von Zuständen (ELEKTRONIK-KOMPENDIUM, 2024b).

Die folgende Abbildung 19 zeigt die beiden verknüpften NOR-Gatter, die entsprechende Wahrheitstabelle bzw. das Schaltzeichen eines RS-Flip-Flops:

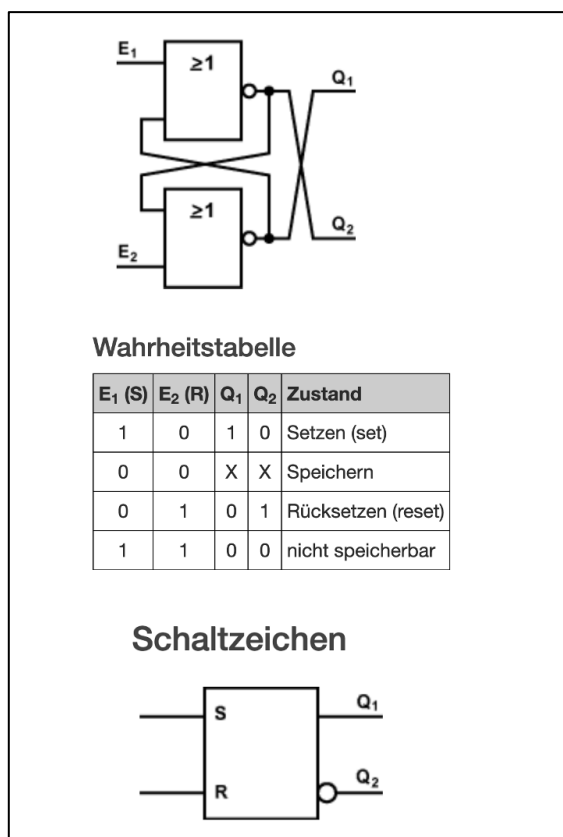


Abbildung 19: RS-Flip-Flop samt Wahrheitstabelle (ELEKTRONIK-KOMPENDIUM, 2024b)

Der nicht-gewünschter Zustand wäre $Reset = 1$ und $Set = 1$ gleichzeitig, ein sogenannter „verbotener Zustand“ (WÖSTENKÜHLER, 2024, S. 132–133). Vergleicht man diese Speicherlogik mit der Fernbedienung eines Fernsehapparates, so würde es hier keinen Sinn machen, einen Fernsehsender gleichzeitig neu zu speichern (setzen) und zurückzusetzen (reset). Sehr wohl wäre aber eine wiederholte Speicherung desselben Senders möglich.

3.8.1 Static Random-Access Memory (SRAM)

SRAM wird gemäß Mikrocontroller (2024) bei Mikrocontrollern (bzw. Mikroprozessoren) als RAM bzw. als Cache von CPUs eingesetzt. Dieser Speichertyp benutzt, wie bereits erwähnt, die Flip-Flop-Technologie, wobei vier bis sechs Transistoren hierfür verwendet werden. SRAM ist meist intern auf dem Mikrocontroller bzw. direkt in der CPU verbaut, kann jedoch auch problemlos extern angeschlossen werden (MIKROCONTROLLER, 2024). Wichtig ist es zu betonen, dass SRAM ein flüchtiger Speicher ist, der bei Stromausfall sämtliche Informationen verliert.

3.8.2 Dynamic Random-Access Memory (DRAM)

Bei der Speicherung mit DRAM kommt ein Transistor in Kombination mit einem Kondensator zum Einsatz. Dabei agiert der Transistor als eine Art Schalter. Das Bit selbst wird im Kondensator gespeichert. DRAM ist ebenfalls volatil und verliert mit dem Verlust der Stromzufuhr seine Speicherkapazität. Der Einsatz des Kondensators ist nicht unproblematisch, denn diese Konstellation benötigt einen geringen Stromfluss. Es besteht die Gefahr von Kriechströmen (unerwünschte Stromflüsse zwischen den Kondensatorplatten), sodass der Kondensator laufend aktualisiert (*refreshed*) werden muss (ELEKTRONIK-KOMPENDIUM, 2024c). Die Abbildung 20 zeigt den Aufbau eines DRAM:

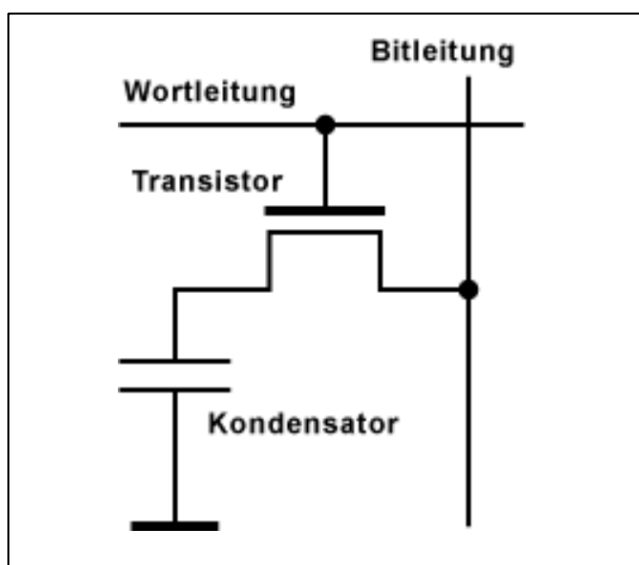


Abbildung 20: Aufbau eines DRAM (ELEKTRONIK-KOMPENDIUM, 2024c)

3.8.3 Register

Register in Mikrocontrollern übernehmen wesentliche Aufgaben bei der Speicherung und Verarbeitung von Daten. Sie sind ebenso wie SRAM flüchtige Speicher und befinden sich vor allem im Arbeitsbereich des Prozessors. Ausgenommen davon sind laut Mikrocontroller (2024) sogenannte *Special Function Register* (SFR). Das sind spezielle Register, die für die Steuerung von Schnittstellen, der Zeit und vielem mehr (IO-Pins, UART, SPI, Timer) am Mikrocontroller zuständig sind. Sie sind ebenfalls auf dem Mikrocontroller verbaut, aber nicht direkt bei der CPU (MIKROCONTROLLER, 2024).

Werden einzelne Speicherzellen (wie etwa Flip-Flops) zusammengefügt und in Reihe geschaltet, spricht man wiederum von Registern (REICHL, 2023, S. 35) bzw. Schieberegistern (WÖSTENKÜHLER, 2024, S. 187–190). Die folgende Abbildung 21 zeigt ein 4-Bit-Schieberegister:

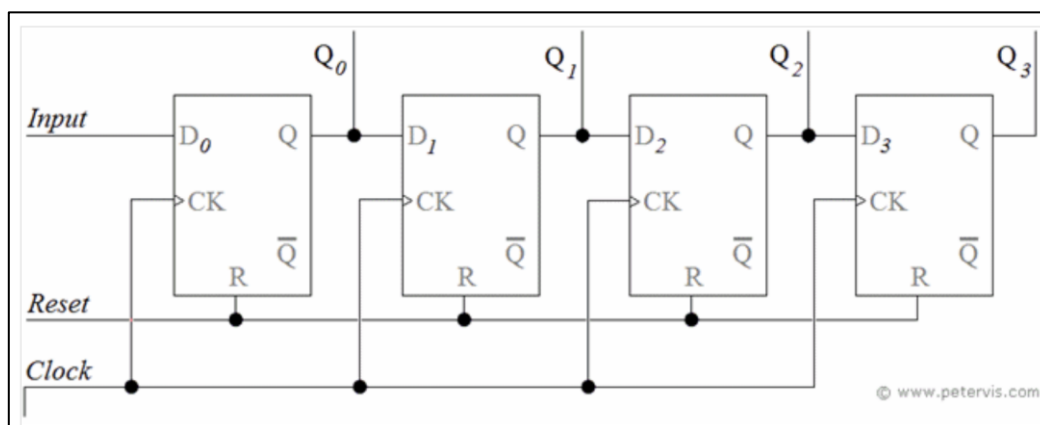


Abbildung 21: 4-Bit-Register (VIES, 2024)

Durch die reihenförmige Anordnung der einzelnen Schaltungen können Bits beliebig nach links oder rechts verschoben werden.

3.8.4 Solid-State-Disc (SSD) und Flash-Speicher

Solid-State-Disc (SSD) und Flash-Speicher sind eng miteinander verbunden. Als einsatzbereiter Speicher (USB-Stick, SSD-Festplatte) wird als SSD bezeichnet. Die Technik von SSD wiederum basiert auf der Flash-Technologie. Flash ist ein nichtflüchtiger, elektroni-

scher Speicher, der auch bei Unterbrechung der Stromzufuhr seine Informationen behält. Flash-Speicher bestehen aus Transistoren vom Typ *Metal Oxide Semiconductor Field-Effect Transistor* (MOSFET). Im Gegensatz zum herkömmlichen MOSFET (siehe Abbildung 22 links) besitzt ein sogenannter *Floating-Gate-Transistor* ein zusätzliches Gate (*Floating-Gate*). Das Control-Gate kann mit dem Standard-Gate eines MOSFETs verglichen werden. Hier wird eine Spannung angelegt, die variabel sein kann. Wird am Floating Gate eine positive Spannung angelegt, entsteht im P-Container zwischen Source und Drain ein Stromfluss. Zugleich werden Elektronen als Ladungen im Floating Gate erzeugt, die dann grundsätzlich auch im Floating Gate verbleiben. Dieses Floating Gate ist vollständig von einer isolierenden Oxidschicht umgeben. Je nach Anzahl bzw. Größe der Ladung wird der Zustand des Floating Gate als „1“ oder „0“ interpretiert. Ist im Floating Gate eine große Ladung an negativen Elektronen vorhanden, fließt kein Strom zwischen Drain und Source, dies wird als Zustand „0“ interpretiert. Gibt es im Floating Gate keine Elektronen bzw. keine negative Ladungen, fließt beim Anlegen einer Spannung am Control Gate ein Strom zwischen Drain und Source, das wäre der Zustand „1“. Beim Löschvorgang wird ein starkes, elektrisches Feld zwischen Control Gate, Floating Gate und Substrat P über eine umgekehrte Spannung erzeugt (das Substrat wird positiv). Dadurch werden Elektronen aus dem Floating-Gate durch die dünne Tunneloxidschicht in das positive Substrat gezogen („getunnelt“). Der Löschvorgang bewirkt wieder einen Stromfluss zwischen Drain und Source, es ist das Schreiben einer „1“. Das Programmieren einer „0“ geschieht, indem Elektronen in das Floating-Gate „injiziert“ werden. Diese Elektronen verhindern einen Stromfluss zwischen Drain und Source, somit wird der Transistor nichtleitend, was dem Zustand "0" entspricht. Die Injektion der Elektronen erfolgt durch eine stark erhöhte Spannung am Control-Gate oder durch gesonderte externe Spannung an der Drain. Die Elektronen „schwappen“ in beiden Fälle vom Substrat P in das Floating Gate (LUBER, 2024) (YALE UNIVERSITY, 2024). Dieser Vorgang ist durch eine Abnutzung der Oxidschicht nicht beliebig oft wiederholbar. Dadurch haben Flash-Speicher eine begrenzte Lebensdauer. Die Abbildung 22 zeigt die wichtigsten Komponenten eines herkömmlichen MOSFET bzw. eines Floating-Gate-Transistors als Flash-Speicher:

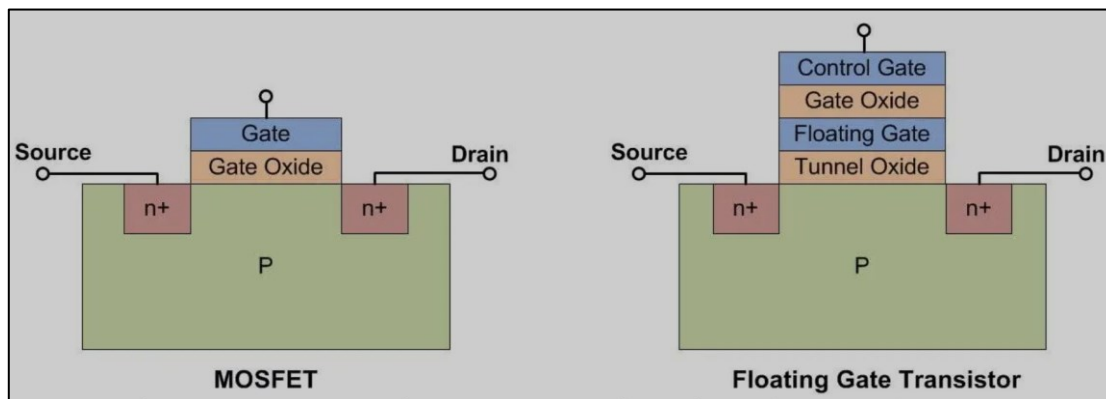


Abbildung 22: Gegenüberstellung eines herkömmlichen MOSFET-Halbleiters mit einem Floating-Gate-Transistor (ARAVINDAN, 2018)

Floating-Gate-Transistoren können in Kombination mit NOR- und AND-Gattern verwendet werden, um komplexere Flash-Speicher zu realisieren (ELEKTRONIK-KOMPENDIUM, 2024d). Daten werden blockweise gelöscht und programmiert. Typische Blockgrößen liegen zwischen 512 Bytes und mehreren Kilobytes. Dies macht Flash-Speicher weniger flexibel für häufige Updates, aber effizient für das Speichern großer, konstanter Datenmengen.

Electrically Erasable Programmable Read-Only Memory (EEPROM) ist eine weitere Ausprägung des Floating-Gate-Transistors. Es handelt sich hierbei ebenfalls um eine nichtflüchtige Speichertechnologie, die Daten ohne Stromversorgung speichern kann (KERSKEN, 2016). Daten können zudem byteweise gesondert gelöscht und programmiert werden. Dies ermöglicht eine sehr exakte Steuerung der Daten, was ideal für Anwendungen ist, bei denen kleine Datenmengen oft und präzise aktualisiert werden müssen.

3.8.5 Read-only Memory (ROM)

Read-only Memory (ROM) ist eine Art von nichtflüchtigem (*non-volatile*) Speicher, der in Computern und anderen elektronischen Geräten verwendet wird, um Daten dauerhaft zu speichern. Die Daten können nur ausgelesen werden. Ein ROM behält seine gespeicherten Daten auch dann, wenn die Stromversorgung unterbrochen wird (MEUTH, 2017, S. 180). Das prädestiniert es für die Speicherung von Firmware, Bootloadern und Konfigurationsdateien, die für den Start des Programms erforderlich sind. Siemers & Sikora (2007) beschreiben mehrere Arten von ROM-Speichern in unterschiedlichen Varianten:

- Masken-ROM (MROM)

Die Daten werden während des Herstellungsprozesses fest programmiert. Diese Art von ROM ist sehr kostengünstig in der Massenproduktion, aber nach der Programmierung nicht mehr veränderbar.

- Programmable ROM (PROM):

PROM kann von Nutzer*innen einmalig programmiert werden. Nach dem Programmieren sind die Daten gespeichert und können nicht mehr geändert werden.

- Erasable Programmable ROM (EPROM)

EPROM kann durch UV-Licht gelöscht und dann neu programmiert werden. Diese Art von ROM wird über einen speziellen EPROM-Programmierer beschrieben und gelöscht (SIEMERS & SIKORA, 2007, S. 326–331).

3.9 Mikrocontroller versus Mikroprozessor

Die beiden Begriffe Mikrocontroller bzw. Mikroprozessor werden oft verwechselt oder auch falsch verwendet. Tatsächlich ist eine klare Unterscheidung nicht so einfach. Gemäß Kersken (2016) ist der Mikroprozessor als Central Processing Unit (CPU) der wichtigste Teil eines Computers. Er ist auf dem Motherboard situiert, regelt die Ausführung der Programme und steuert die angeschlossene Hardware. Verwirrend mag die Tatsache sein, dass ein Mikroprozessor mehrere Kerne haben kann, die im Prinzip wiederum als gesonderte CPUs fungieren bzw. die Arbeit teilen. Mehrere Kerne (*Kernels*) haben aber ein einziges Gehäuse und sind auf einem einzelnen CPU-Sockel angebracht (KERSKEN, 2016, S. 119–121). RS (2023) beschreibt Mikroprozessoren zudem als sehr leistungstark mit Taktfrequenzen im Gigaherzbereich. Sie arbeiten primär mit externen Komponenten wie Arbeitsspeicher als *Random Access Memory* (RAM), *Read-only Memory* (ROM) und *Input-Output-Ports* (I/O-Ports) und sind für vielfältige, parallel ablaufende Aufgabenstellungen konzipiert (RS, 2023).

Mikrocontroller arbeiten ebenfalls mit einer CPU, verzichten aber auf ein Betriebssystem und arbeiten eher Firmware-basiert. Funktionsblöcke wie WLAN, Bluetooth, Sensoren, aber auch Arbeits- und Festspeicher sind auf dem Mini-Board zusammen mit der CPU fix aufgelötet (ALSABBAGH, 2019, S. 9). Typisch für einen Mikrocontroller sind eine

eingeschränkte Funktionsvielfalt und das Vorhandensein von nur wenigen I/O-Anschlüssen, die aber sehr hohe Übertragungsraten haben können (WINDECK, 2021).

Interessant ist auch die Feststellung, dass es nicht immer eindeutig ist, ob ein Gerät als Mikrocontroller oder Mikroprozessor eingestuft wird. So zeigt die folgende Abbildung 23, dass der bekannte Mini-Computer Raspberry Pi der gemeinnützigen Institution *Raspberry Pi Foundation* mit den ähnlichen Modellen *RP Pico* bzw. *RP 4* in beiden Segmenten vertreten sein kann. Raspberry-Pi-Modelle sind vorwiegend als Mikroprozessoren bekannt, jedoch werden auch Mikrocontroller offeriert.

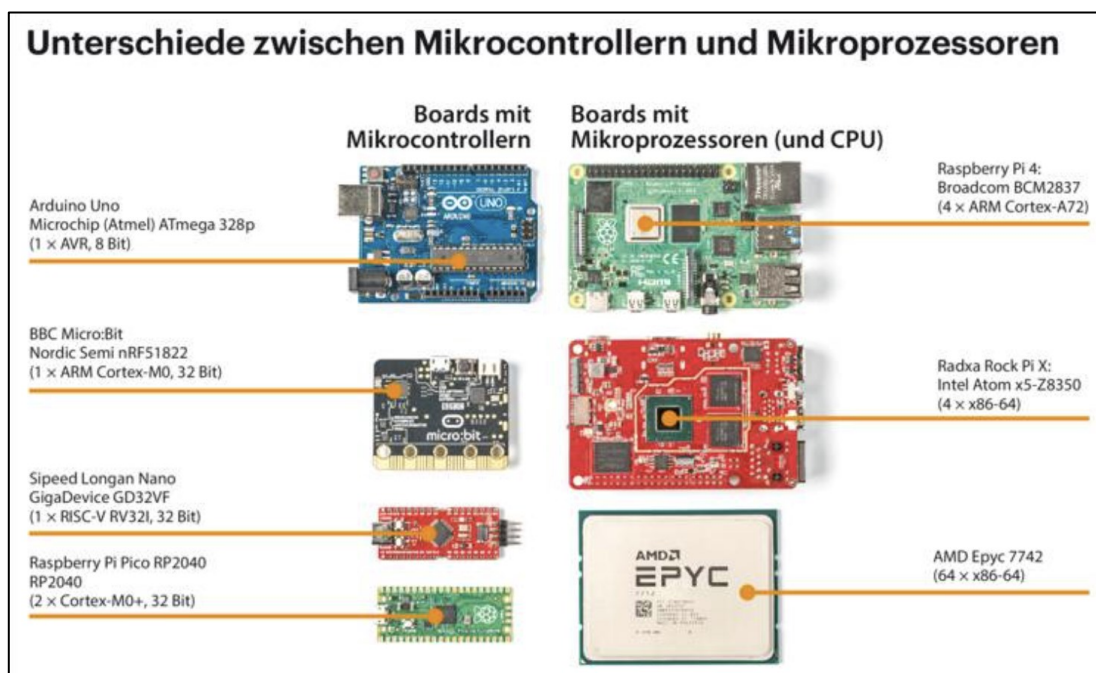


Abbildung 23: RP Pico und RP 4 als Mikrocontroller und Mikroprozessor (WINDECK, 2021)

Aufgrund der eingeschränkten Funktionalität, des fehlenden Betriebssystems und der fix vorgegebenen Schnittstellen ist die Programmierung von Mikrocontrollern einfacher. Bei Boards mit Mikroprozessoren ist der Lernaufwand bereits aufgrund des Betriebssystems erhöht. Das bereits erwähnte Modell RP 4 wird in der Regel mit dem Linux-Betriebssystem-Derivat *Raspian* betrieben, was logischerweise Know-how im Bereich Linux erfordert.

4. Digitale Grundlagen im AHS-Lehrplan und IT-Initiativen

Da sich diese Masterthesis auf Lehr- und Lerninhalte an Gymnasien fokussiert, ist es wichtig, den Lehrplan für die Unterrichtsfächer Informatik und „Digitale Grundbildung“ zu kennen. Seit dem Schuljahr 2022/2023 wurde an allen Schulen Österreichs im Bereich der Sekundarstufe 1 das neue, verpflichtende Unterrichtsfach „Digitale Grundbildung“ (DGB) implementiert. Die Sekundarstufe 1 betrifft in Österreich Schüler*innen im Alter von zehn bis vierzehn (Schulstufe fünf bis acht), verteilt auf die Schulformen Mittelschule, allgemein bildende höhere Schule Unterstufe (Gymnasium) und Sonderpädagogik (OEAD, 2023a). Ab dem Schuljahr 2023/2024 sind Lehrinhalte von DGB auch als verbindliche Übung an Sonderpädagogischen Schulen rechtlich verankert (BUNDESMINISTERIUM FÜR BILDUNG, WISSENSCHAFT UND FORSCHUNG, 2023b). Lehrpläne an öffentlichen Schulen Österreichs werden durch das Schulorganisationsgesetzes², BGBl. Nr. 242/1962, zuletzt geändert durch das Bundesgesetz BGBl. Nr. 365/1982, insbesondere dessen §§ 6 und 39, im Verordnungsweg³ veröffentlicht.

4.1 Lehrplan DGB für AHS-Unterstufe

Erstmalig im Schuljahr 2022/2023 wurde das Pflichtfach „Digitale Grundbildung“ eingeführt und ist seit dem Schuljahr 2023/2024 durchgehend in allen fünften bis achten Schulstufen mit mindestens einer fixen Stunde pro Woche verpflichtend zu unterrichten (im Einführungsjahr 2022 war die 8. Schulstufe noch ausgenommen). Ein interessantes Detail am Rande ist die durch die DGB-Einführung erfolgte Erhöhung der Jahreswochenstunden um vier Jahreswochenstunden gesamt in der Sekundarstufe I. Pro Jahr erhöhte sich somit die Anzahl an Wochenstunden einheitlich um eine DGB-Stunde, ohne dass ein anderes Unterrichtsfach gekürzt werden musste. Dafür wurden sogar 150 neue Planstellen für Lehrer*innen budgetiert (BUNDESMINISTERIUM FÜR BILDUNG, WISSENSCHAFT UND FORSCHUNG, 2023a).

² Bundesgesetz vom 25. Juli 1962 über die Schulorganisation (Schulorganisationsgesetz). StF: BGBl. Nr. 242/1962 idF BGBl. Nr. 267/1963

³ Verordnung des Bundesministers für Unterricht und Kunst vom 14. November 1984 über die Lehrpläne der allgemeinbildenden höheren Schulen; Bekanntmachung der Lehrpläne für den Religionsunterricht an diesen Schulen, StF: BGBl. Nr. 88/1985

4.1.1 Konzepte und Kompetenzen

Der Lehrplan von DGB wurde gemäß der Verordnungsbefähigung des Schulorganisationsgesetzes, BGBl. Nr. 242/1962, in der Verordnung 267 vom 6. Juli 2022 normiert. Dabei orientiert sich das Pflichtfach DGB an vorgegebenen Konzepten, um die angestrebten Kompetenzen der Schüler*innen zu erreichen. Die Konzepte wiederum betonen das Modell „Frankfurt Dreieck“, welches DGB aus drei Perspektiven betrachtet:

- **technisch-medial (T)**
- **gesellschaftlich-kulturell (G)**
- **interaktionsbezogenen (I)**

Mit diesen Untergliederungen gibt es somit keine einseitig technische Informatik-Komponente. Die Bestandteile Kommunikation/Interaktion bzw. Gesellschaft und Kultur zeigen auch deutlich die globalen Auswirkungen von Informationstechnik auf die Gesellschaft. Dadurch wird DGB auch als fächerübergreifendes, abstraktes Bindeglied zu anderen Unterrichtsfächern inhaltlich und praktisch relevant.

Die folgende Abbildung 24 fasst nochmals die Konzepte zu DGB zusammen und leitet davon fünf wichtige Kompetenzen für Schüler*innen in DGB ab, die sich über alle vier Jahre der SEK1 erstrecken:

Strukturen und Funktionen digitaler informatischer und medialer Systeme und Werkzeuge (T)

Diese beinhalten informatische Funktions- und Wirkprinzipien sowie die Reflexion ihrer nicht unmittelbar sichtbaren Einflüsse auf Kultur, Politik, Gesellschaft und Lebenswelt, Technik und Produktion.

Gesellschaftliche Wechselwirkungen durch den Einsatz digitaler Technologien (G)

Diese betreffen etwa soziale Umgangsformen, die politische Organisation, Machtstrukturen, ökonomische Interessen oder die Kommunikation. Die historische Perspektive erlaubt es, Kontinuitäten und Entstehungsprozesse sowie damit verbundene Traditionen zu verstehen.

Interaktion in Form von Nutzung, Handlung und Subjektivierung (I)

Sie erlaubt die Analyse, Reflexion und kreative Gestaltung persönlicher und kollektiver Handlungsrepertoires. Dies beinhaltet auch die Frage, wie und mit welchen Zielen Menschen Medien erstellen, gestalten und Systeme nutzen.

Kompetenzmodell und Kompetenzbereiche (1. bis 4. Klasse):

- **Orientierung:** gesellschaftliche Aspekte von Medienwandel und Digitalisierung analysieren und reflektieren
- **Information:** mit Daten, Informationen und Informationssystemen verantwortungsvoll umgehen
- **Kommunikation:** Kommunizieren und Kooperieren unter Nutzung informatischer, medialer Systeme
- **Produktion:** Inhalte digital erstellen und veröffentlichen, Algorithmen entwerfen und Programmieren: Zerlegen von Problemen, Muster erkennen, Verallgemeinern/Abstrahieren und Algorithmen entwerfen
- **Handeln:** Angebote und Handlungsmöglichkeiten in einer von Digitalisierung geprägten Welt einschätzen und verantwortungsvoll nutzen

Abbildung 24: Strategien und Kompetenzen von DGB (VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 1. Unterstufe, Digitale Grundbildung, Zentrale fachliche Kompetenz (1. bis 4. Klasse))

Da der Fokus der Masterarbeit auf Illustration und praktischem Einsatz von Digitaltechnik, Programmierung und Making liegt, sind insbesondere die Konzepte Technik (T) und Interaktion (I) im Unterrichtsfach DGB relevant. Die genannten Schwerpunkte betreffen besonders die Kompetenzen „Information“ und „Produktion“.

4.1.2 Digitaltechnik, Programmierung und Making

Über alle vier Jahre des Unterrichtsfachs DGB sind durchgehend die Wichtigkeit und Bedeutung von Programmierung erkennbar. Bereits im ersten Jahr DGB (5. Schulstufe) werden im Kompetenzbereich *Produktion* Begriffe wie Algorithmen oder Programmieren erwähnt. Ebenso findet sich ein Verweis auf die digitalen Komponenten eines Computers. Die folgende Abbildung 25 zeigt den aktuellen Auszug der angestrebten Kompetenzen der 5. Schulstufe in DGB:

Kompetenzbereich Produktion: Inhalte digital erstellen und veröffentlichen, Algorithmen entwerfen und Programmieren

Die Schülerinnen und Schüler können

- (T) eindeutige Handlungsanleitungen (Algorithmen) nachvollziehen, ausführen sowie selbstständig formulieren.
- (G) verschiedene Darstellungsformen von Inhalten und die Wirkung auf sich und andere beschreiben.
- (I) mit Daten einfache Berechnungen durchführen sowie in verschiedenen (visuellen) Formaten sammeln und präsentieren.
- (I) einzeln und gemeinsam Texte und Präsentationen (unter Einbeziehung von Bildern, Grafiken und anderen Objekten) strukturieren und formatieren.

Anwendungsbereiche

- Sequenzen und einfache Schleifen
- Planung, Gestaltung und Auswertung von Umfragen

Kompetenzbereich Handeln: Angebote und Handlungsmöglichkeiten in einer von Digitalisierung geprägten Welt einschätzen und verantwortungsvoll nutzen

Die Schülerinnen und Schüler können

- (T) unter Nutzung einer angemessenen Terminologie gängige physische Komponenten von Computersystemen (Hardware) identifizieren sowie beschreiben, wie interne und externe Teile von digitalen Geräten funktionieren und ein System bilden.
- (G) Mediennutzungsformen sowie deren historische Entwicklung und gesellschaftliche Etablierung im Zuge des Medienwandels beschreiben.
- (I) Hilfesysteme bei der Problemlösung nutzen.

Anwendungsbereiche

- Wichtigste Komponenten eines Computers
- Notwendige Funktionen eines Betriebssystems im Normalbetrieb

Abbildung 25: Digitaltechnik und Programmierung DGB 5. Schulstufe VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 1. Unterstufe, Digitale Grundbildung, zentrale fachliche Kompetenz (1. bis 4. Klasse))

Der Kompetenzbereich Produktion mit seinem Schwerpunkt auf Programmierung bleibt unverändert in der gesamten SEK1-Stufe erhalten und wird jedes Jahr in der Komplexität erhöht. In der 6. Schulstufe ist gemäß Abbildung 26 erstmals ein Vermerk auf eine „binäre Struktur“ im Kompetenzbereich *Information* erkennbar:

Kompetenzbereich Information: mit Daten, Informationen und Informationssystemen verantwortungsvoll umgehen Die Schülerinnen und Schüler können – (T) Daten erfassen, filtern, sortieren, interpretieren und darstellen. – (T) beschreiben, wie über das Internet Informationen bereitgestellt und abgerufen sowie Daten übertragen werden. – (I) Lizenzmodelle, insb. offene (Creative Commons, Open Educational Resources, Open Source), benennen, erklären und anwenden. Anwendungsbereiche – Organisation von Daten – (Visuelle) Darstellung von Daten – Beschreibung von Daten hinsichtlich ihrer Formate, Größe und binären Struktur
--

Abbildung 26: Kompetenzbereich Information mit Verweis auf binäre Struktur (VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 1. Unterstufe, Digitale Grundbildung, Zentrale fachliche Kompetenz (1. bis 4. Klasse))

Ein Verweis auf IoT findet sich erstmalig im Lehrplan der 7. Schulstufe wieder, allerdings fehlt hier eine Forderung nach praktischer Anwendungskompetenz. In der 8. Schulstufe wird der Kompetenzbereich *Produktion* sogar um die Organisation von Programmierung in Form von Kompetenzen wie Dokumentation und Projektmanagement ergänzt (siehe folgende Abbildung 27):

Kompetenzbereich Produktion: Inhalte digital erstellen und veröffentlichen, Algorithmen entwerfen und Programmieren Die Schülerinnen und Schüler können – (T) Programme entwerfen und iterativ entwickeln, die Kontrollstrukturen kombinieren, einschließlich verschachtelter Schleifen und zusammengesetzter Konditionale. – (G) Möglichkeiten verschiedener Darstellungsformen von Inhalten erproben und deren Einfluss auf die Wahrnehmung des Inhalts hinterfragen. – (I) mit bereitgestellten Medien und Software-Applikationen zielgerichtet und kreativ gestaltend kooperieren. – (I) einfache Programme oder Webanwendungen mit geeigneten Werkzeugen erstellen, um ein bestimmtes Problem zu lösen oder eine bestimmte Aufgabe zu erfüllen. Anwendungsbereiche – Dokumentation von Programmen – Projektplanung inklusive Aufgabenverteilung und Zeitplan – (Graphische) Notationen, Pseudocode – Ästhetische und technische Kompetenzen von Medienkulturen in Projekten

Abbildung 27: DGB 8. Schulstufe Kompetenz Produktion (VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 1. Unterstufe, Digitale Grundbildung, Kompetenzbeschreibungen und Anwendungsbereiche, Lehrstoff (1. bis 4. Klasse), 4. Klasse)

4.2 Lehrplan Informatik AHS Oberstufe

Der Lehrplan in der AHS-Oberstufe scheint derzeit überhaupt nicht an das Vorhandensein des neuen Pflichtfachs „Digitale Grundbildung DGB“ angepasst. Während DGB über vier Jahre mit mindestens einer Wochenstunde verpflichtend ist, sieht das Pflichtfach Informatik in der Oberstufe nur zwei Wochenstunden in der 9. Schulstufe vor. Von der zehnten bis zur zwölften Schulstufe gibt es kein vorgesehene Pflichtfach Informatik. In Zeiten der Digitalisierung von Studium und Beruf ist das mit Sicherheit ein großes Manko. Die Abbildung 28 zeigt die Stundentafel für AHS-Oberstufe mit Markierung des Pflichtfaches Informatik:

2. Soweit keine schulautonomen Lehrplanbestimmungen bestehen:						
aa) Pflichtgegenstände	Klassen und				Summe Oberstufe	Lehrverpflich- tungsgruppe
	5. Kl.	6. Kl.	7. Kl.	8. Kl.		
Religion/Ethik ^{1a)}	2	2	2	2	8	(III)/III
Deutsch	3	3	3	3	12	(I)
Erste lebende Fremdsprache	3	3	3	3	12	(I)
Zweite lebende Fremdsprache/Latein	3	3	3	3	12	(I)
Geschichte und Sozialkunde/ Politische Bildung	1	2	2	2	7	(III)
Geographie und Wirtschaftskunde *)	2	1	3	3	9	(III)
Mathematik	3	3	3	3	12	(II)
Biologie und Umweltkunde *)	2	3	-	2	7	III
Chemie	-	-	2	2	4	(III)
Physik	-	3	2	2	7	(III)
Haushaltsökonomie und Ernährung (Theorie) *)	2	2	-	-	4	III
Psychologie und Philosophie (einschließlich Praktikum) *)	-	1	2	2	5	III
Informatik	2	-	-	-	2	II
Musikerziehung	2	1	2	2	3	(IVa)
Bildnerische Erziehung	2	1	1)	1)	3	(IVa)
Bewegung und Sport	3	2	2	2	9	(IVa)
Summe der Pflichtgegenstände	30	30	29	31	120	

Abbildung 28: Pflichtfächer Oberstufe AHS (VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 2. Oberstufe, Pflichtgegenstände)

Zwar besteht noch die Möglichkeit, Informatik über ein Wahlpflichtfach ergänzend zum Pflichtfach anzubieten, aber das ist im Grund nur dann ein Pflichtfach, wenn es von Schüler*innen auch ausgesucht wird. Wahlpflichtfächer kommen nur zustande, wenn genügend Schüler*innen dies auch wählen bzw. wenn es von einem Gymnasium überhaupt angeboten wird (oft fehlen hierzu auch die Lehrpersonen). Ein Wahlpflichtfach Informatik sollte bei Zustandekommen mindestens vier Wochenstunden über zwei Jahre oder sechs Wochenstunden über drei Jahre beinhalten (siehe Abbildung 29):

Wahlpflichtgegenstände					
2)				10	10
Gesamtwochenstundenzahl aa) + bb)				130	
*) Typenbildende Pflichtgegenstände					
1) Alternative Pflichtgegenstände					
1a) Pflichtgegenstand für Schülerinnen und Schüler, die am Religionsunterricht nicht teilnehmen. Das Stundenausmaß des Pflichtgegenstandes Ethik ist nicht veränderbar.					
2) Siehe den Unterabschnitt Wahlpflichtgegenstände					
Oberstufe des Gymnasiums, Realgymnasiums und Wirtschaftskundlichen Realgymnasiums					
Wahlpflichtgegenstände 1)	Klassen und Wochenstunden				Lehrverpflichtungsgruppe 2)
	6.Kl	7.Kl	8.Kl.	Summe	
aa) Lebende Fremdsprache 2)	2	2	2	6	(I)
Darstellende Geometrie 3)	-	2	2	4	(II)
Informatik	2	2	2	6	II

Abbildung 29: Wahlpflichtfach Informatik über drei Jahre (VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 2. Oberstufe, Wahlpflichtgegenstände)

Interessant für diese Masterthesis ist vor allem das Pflichtfach Informatik in der 5. Klasse bzw. in der 9. Schulstufe. Digitale Grundlagentechnik ist im Lehrplan nur angedeutet mit „Funktionsweise von Informatiksystemen“. Wiederum stark vertreten ist aber eine programmiertechnische Komponente, bei der es primär um die Implementierung von Algorithmen und Programmiersprachen geht. Betrachtet man jedoch den aktuellen Lehrplan, der lediglich ein Schuljahr im Ausmaß von 2 Wochenstunden betrifft, scheinen die geforderten Bildungsziele sehr optimistisch zu sein. Die folgende Abbildung 30 zeigt den aktuellen Lehrplan Informatik der 9. Schulstufe:

5. Klasse (1. und 2. Semester)

Informatik, Mensch und Gesellschaft

- Die Bedeutung von Informatik in der Gesellschaft beschreiben, die Auswirkungen auf die Einzelnen und die Gesellschaft einschätzen und Vor- und Nachteile an konkreten Beispielen abwägen können
- Maßnahmen und rechtliche Grundlagen im Zusammenhang mit Datensicherheit, Datenschutz und Urheberrecht kennen und anwenden können
- Die Entwicklung der Informatik beschreiben und bewerten können
- Informatikberufe und Einsatzmöglichkeiten der Informatik in verschiedenen Berufsfeldern benennen und einschätzen können

Informatiksysteme

- Den Aufbau von digitalen Endgeräten beschreiben und erklären können
- Die Funktionsweise von Informatiksystemen erklären können
- Grundlagen von Betriebssystemen erklären, eine graphische Oberfläche und Dienstprogramme bedienen können
- Grundlagen der Vernetzung von Computern beschreiben und lokale und globale Computernetzwerke nutzen können

Angewandte Informatik

- Standardsoftware zur Kommunikation und Dokumentation sowie zur Erstellung, Publikation und multimedialen Präsentation eigener Arbeiten einsetzen können
- Standardsoftware für Kalkulationen und zum Visualisieren anwenden können
- Informationsquellen erschließen, Inhalte systematisieren, strukturieren, bewerten, verarbeiten und unterschiedliche Informationsdarstellungen verwenden können
- Digitale Systeme zum Informationsaustausch, zur Unterstützung der Unterrichtsorganisation und zum Lernen auch in kommunikativen und kooperativen Formen verwenden können

Praktische Informatik

- Begriffe und Konzepte der Informatik verstehen und Methoden und Arbeitsweisen anwenden können
- Algorithmen erklären, entwerfen, darstellen und in einer Programmiersprache implementieren können
- Grundprinzipien von Automaten, Algorithmen, Datenstrukturen und Programmen erklären können
- Datenbanken benutzen und einfache Datenmodelle entwerfen können

Abbildung 30: Lehrplaninhalte Pflichtfach Informatik 9. Schulstufe VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, fünfter Teil, FÜNFTER TEIL, 2. Oberstufe, Informatik)

Mit dem Schuljahr 2025/2026 wird aller Voraussicht nach der Lehrplan Informatik für die AHS-Oberstufe geändert. Geplant ist, dass von Schulstufe neun bis zwölf mindestens eine Wochenstunde Informatik im Lehrplan enthalten sein soll. Schulautonom kann diese Mindeststunde eigenständig beliebig erhöht werden. Gemäß einem bereits schulintern veröffentlichten Lehrplan-Vorentwurf in legistischer Formulierung (siehe Anhang 13.1) finden sich bei den Kompetenzen explizite Inhalte der Digitaltechnik samt Programmierung. Sogar Physical Computing samt Programmierung und Making ist enthalten. Die Abbildung 31 zeigt alle geforderten zehn neuen Kompetenzen:

Um diese Fähigkeiten zu beherrschen, werden eine Vielzahl an informatischen Konzepten, Prinzipien sowie Praktiken benötigt. Dem “informatics for all”-Framework folgend sind diese in zehn zentrale Themenbereiche gegliedert.

- Daten und Informationen (DUI). Verstehen, wie Daten gesammelt, organisiert, analysiert und verwendet werden, um Informationen über reale Artefakte und Szenarien zu modellieren, darzustellen und zu visualisieren.
Funktionsweisen und Einsatzmöglichkeiten von künstlicher Intelligenz nachvollziehen und bewerten.
- Algorithmen und Datenstrukturen, Programmierung (ADP). Bewerten, spezifizieren, entwickeln und verstehen von Algorithmen; Programmiersprachen nutzen, um sich durch die Entwicklung, das Testen und das Debuggen digitaler Artefakte rechnergestützt auszudrücken; und verstehen, was eine Programmiersprache ist.
- Computersysteme (CMP). Verstehen, was ein (virtuelles) Computersystem ist, wie seine Bestandteile gemeinsam als Ganzes funktionieren, und seine Grenzen kennen. Kenntnisse von Einplatinenrechnern inklusive Aktoren und Sensoren bis hin zu virtuellen Systemen.
- Netzwerke und Kommunikation (NWK). Verstehen, wie Netzwerke den Austausch digitaler Artefakte auf vielfältige Art und Weise ermöglichen und Kommunikationsverhalten der Gesellschaft beeinflussen.
- Mensch-Maschine Interaktion (MMI). Bewerten, spezifizieren, entwickeln und verstehen der Interaktion zwischen Menschen und digitalen Artefakten.
- Design und Entwicklung (DUE). Artefakte planen und erstellen, unter Berücksichtigung der Standpunkte von Stakeholdern und durch kritische Bewertung von Alternativen und deren Ergebnissen.
- Digitale Kreativität (DKR). Digitale Werkzeuge erkunden und verwenden, um diverse Artefakte zu entwickeln und zu pflegen, dabei auch eine Vielfalt von Medien nutzen.
- Modellierung und Simulation (MUS). Bewerten, modifizieren, entwerfen, entwickeln und verstehen von Modellen und Simulationen natürlicher und künstlicher Phänomene sowie deren Weiterentwicklung.
- Privatsphäre, Sicherheit und Schutz (PSS). Risiken beim Einsatz digitaler Technologien verstehen und wie man Individuen und Systeme schützt.
- Verantwortung und Selbstbestimmung (VUS). Kritisch und konstruktiv konkrete Computerartefakte sowie fortgeschrittene und potenziell kontroverse Techniken und Anwendungen der Informatik analysieren, insbesondere aus ethischer und sozialer Perspektive.

Abbildung 31: Informatische Konzepte des legislativen Vorentwurfs „Informatik Oberstufe NEU“ (siehe Anhang 13.1)

4.3 Initiative MINT

Die Abkürzung MINT stellt die Fachbereiche Mathematik (M), Informatik (I), Naturwissenschaften (N) und Technik (T) dar. Ziel von MINT ist es, Jugendliche für naturwissenschaftliche Themen zu begeistern, auch im Hinblick auf eine spätere Ausbildungs- und Berufswahl.

4.3.1 MINT im Hochschulbereich

Das Bundesministerium für Bildung, Wissenschaft und Forschung (BMBWF) als zuständiges Ministerium setzt einen besonderen Schwerpunkt auf MINT. Um auf stetige berufliche Herausforderungen der Digitalisierung wie etwa Automatisierung oder Künstliche

Intelligenz (KI) vorbereitet zu sein, fördert das BMBWF gezielt universitäre Einrichtungen in MINT-Bereichen, sowohl in der Ausbildung wie auch in der Forschung (BMBWF, 2023). Im universitären Bereich findet diese Förderung konkret über erhöhte Förderbudgets, zusätzliche MINT-Professuren und den Ausbau von Fachhochschulstudien im MINT-Bereich statt. Aktuell wird das MINT-Angebot im Fachhochschulbereich bis 2027 um 2.650 Studienplätze erweitert, die alle durch das Bundesbudget gedeckt werden (BMBWF, 2023).

4.3.2 MINT in der Pädagog*innen-Ausbildung

*„Auch in der Pädagog/inn/enbildung Neu [sic] spielt das Thema MINT eine große Rolle. Ziel ist bereits im Kindergarten (Stichwort Elementarpädagogik) bzw. in der Schule das Interesse an MINT zu wecken. Daher unterstützt das BMBWF Programme und Maßnahmen im Bildungsbe-
reich ebenso wie an der Nahtstelle zwischen der Schul- und der Hochschulausbildung und dar-
über hinaus.“(BMBWF, 2023)*

Das Interesse an MINT soll bereits sehr früh, am besten schon im Kindergarten, geweckt werden und sich dann über den ganzen Schulbereich erstrecken. Natürlich spielt MINT auch im konkreten Berufsfeld ohne akademische Ausbildung eine Rolle, beispielsweise in der Lehrlingsausbildung bei technischen Berufen nach einem Pflichtschulabschluss.

4.3.3 MINT-Gütesiegel

Eine besondere Form der MINT-Initiative ist das MINT-Gütesiegel. Damit werden Schulen ausgezeichnet, deren Fokus tatsächlich auf naturwissenschaftlichen Fächern liegt. Das MINT-Gütesiegel können elementarpädagogische Einrichtungen, Primarschulen, Schulen der Sekundarstufen, Berufsschulen und technische Schulen beantragen. Das Gütesiegel wird vom BMBWF verliehen und bescheinigt den Schulen unter anderem, dass MINT im Schulprofil verankert ist, einen starken Praxisbezug durch berufsorientierte Projekte hat und forschendes Lernen sowie genderneutrale Förderung von Mädchen und Burschen gewährleistet wird. Diese MINT-Zertifizierung bestätigt den Schulen zudem, dass sie *„mit verschiedenen Maßnahmen innovatives und begeisterndes Lernen in Mathematik, Informatik, Naturwissenschaft und Technik fördern und mit vielfältigen Zugängen für Mädchen und Burschen umsetzen“* (MINT SCHULE, 2023). Durch die starke Forschungsorientierung stellt dieses Gütesiegel speziell für Gymnasien einen großen

Mehrwert dar, da eine Berufsorientierung an Gymnasien nicht klar zu erkennen ist. Die genauen Kriterien für den Status „MINT-Gütesiegel“ am Beispiel der Sekundarstufe (SEK1 und SEK2) zeigt folgende Abbildung 32:

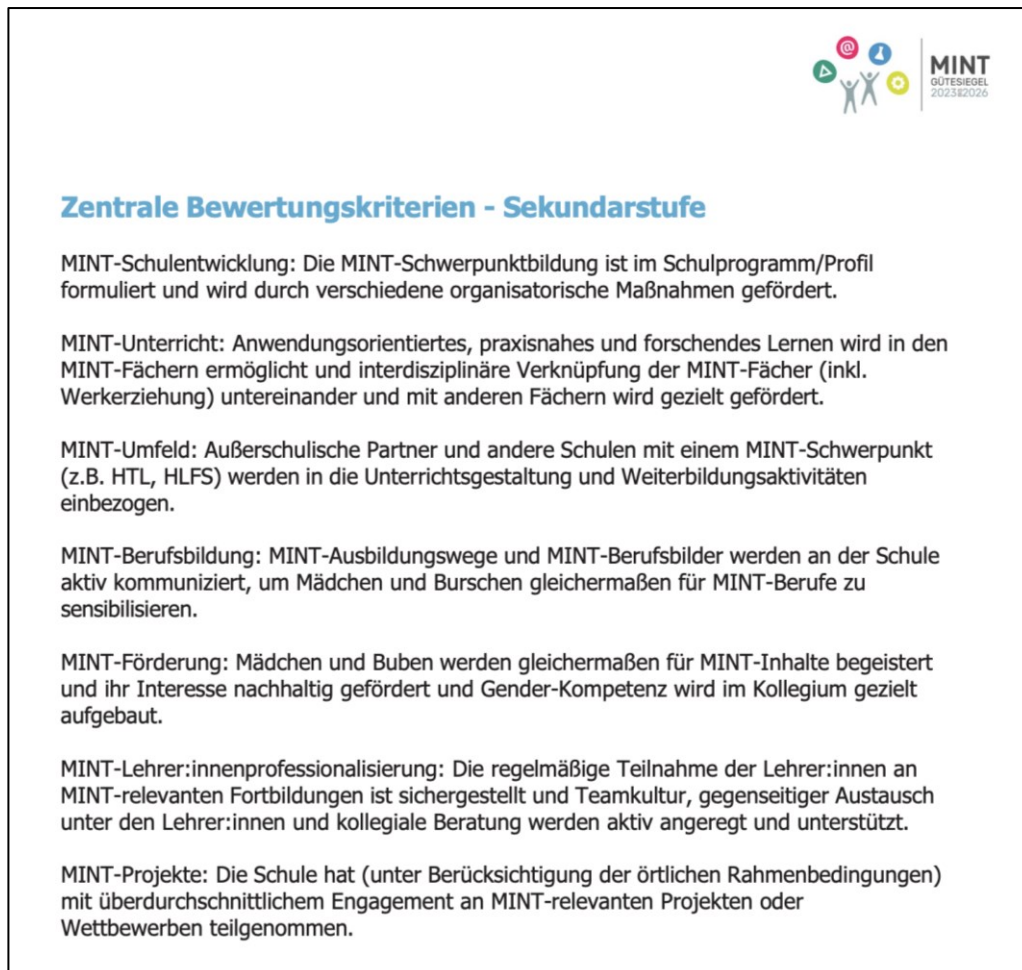


Abbildung 32: Kriterien Sekundarstufe für das MINT-Gütesiegel (MINTSCHULE.AT, 2024)

4.3.4 Citizen Science

Interessant ist auch das Projekt *Citizen Science*, das buchstäblich Forschen und Entdecken in das private Umfeld von Schüler*innen und Bürger*innen bringt. Damit bekommen gemäß OEAD (2023) alle Bürger*innen die Möglichkeit zum wissenschaftlichen Forschen, egal ob Amateur-Forscher*innen oder Wissenschaftler*innen selbst. Besonders forciert wird diese Initiative durch moderne Technologien wie *Global Positioning System* (GPS), Internet, Smartphones, weltweite Vernetzung und leistbare Mikrocontroller und

Sensoren. Entscheidend ist auch die angestrebte Kooperation mit universitären Bildungsträgern, deren Aufgabe in der Beratung, Auswertung, Einordnung und Analyse der Forschungsergebnisse liegt. Wichtig hierbei ist vor allem die Zielsetzung, dass im Endeffekt tatsächlich wissenschaftliche Ergebnisse erzielt und veröffentlicht werden. Es ist also eine gleichwertige Kooperation zwischen Bürger*innen (Schüler*innen) und der Wissenschaft, um das Potential der Amateurforschung optimal zu nutzen (OEAD, 2023b). Wichtig ist auch die Tatsache, dass die Idee von Citizen Science eine weltweite ist. Das Citizen-Science-Netzwerk *Citizen Science Global Partnership* (CSGP) fördert weltweit den Austausch zwischen Wissenschaft und Gesellschaft und hat seit 2022 seinen Sitz in Wien (CITIZEN SCIENCE, 2022).

4.3.5 Programmieren als schulische Kernkompetenz

Mit der Einführung der verbindlichen Übung „Digitale Grundbildung“ im Jahr 2018 und später mit der Implementierung von DGB als Pflichtfach (siehe Kapitel 4.1) trat ein Dilemma zutage, dessen Folgen sich auch im Lehrplan der Sekundarstufen 1 und 2 der Gymnasien wiederfinden. Soll der Informatik-Unterricht den Schwerpunkt auf Medienpädagogik (Umgang und Auswirkungen der Digitalisierung als wirtschaftliches und soziales Thema) setzen oder soll die reine informatische Bildung mit seinem populärsten Vertreter, dem Coding (Programmieren als Kernkompetenz), priorisiert werden? Natürlich hat diese Fragestellung auf den Unterricht in DGB bzw. Informatik (SEK2) der Gymnasien markante Auswirkungen aufgrund der ohnehin limitierten Wochenstunden. Sollte man hier also, vereinfacht ausgedrückt, primär Programmieren unterrichten? Oder müssen die wenigen, vorhandenen Unterrichtseinheiten halbiert werden, um auch die Auswirkungen der informatischen Welt auf die Gesellschaft zu reflektieren? Stadler & Grünberger (2021) stellen fest, dass informatische Bildung und Medienbildung tatsächlich zwei unterschiedliche Bereiche sind, die prinzipiell nicht miteinander vergleichbar sind. Einerseits die streng technologische, mathematische Logikwelt der Informatik, auf der anderen Seite die Auswirkung der Technik auf das soziale Gemeinwohl. Ein möglicher Kompromiss ist der didaktische Ansatz von *Computational Thinking*. Nicht nur die Fähigkeit zum Programmieren ist relevant, gefördert werden soll auch die Fähigkeit, Probleme zu lösen, Problemstellungen zu abstrahieren und Muster zu erkennen.

Dadurch wird der Schaffensprozess kreativer. Größere Zusammenhänge können erkannt werden. Somit ist Computational Thinking möglicherweise der Missing Link zwischen Medienpädagogik und reinem informatischen Denken (STADLER & GRÜNBERGER, 2021, S. 96–99).

„Computational Thinking eröffnet in der pädagogischen Praxis vielfältige Möglichkeiten, um beispielsweise Mediengestaltung aktiv zu erleben oder Phänomene der digital-vernetzten Welt zu verstehen. Somit ist es verständlich, dass Computational Thinking Einzug in Bildungspläne gefunden hat; auch in Österreich. Dabei sind Ansätze einer Verknüpfung von Medienbildung und Informatischer Bildung ersichtlich, die bei einem kritischen Blick allerdings weitere Bemühungen nahelegen.“ (STADLER & GRÜNBERGER, 2021, S. 100)

Medienbildung und informatische Bildung sind beide in den Lehrplänen der Sekundarstufe 1 und 2 hinterlegt und deren Wichtigkeit ist unbestritten. Medienbildung führt jedoch zu weniger Unterrichtsstunden mit rein informatischen Inhalten, wie etwa der Kernkompetenz des Programmierens.

Einen Schritt weiter geht der CEO von Apple, Tim Cook. *„Programmieren ist für Kinder wichtiger, als eine zweite Sprache zu lernen“* (MANSHOLT, 2021). Auch er beruft sich auf die Vorteile von Computational Thinking und sieht enorme Potentiale im Coding beim Problemlösen, kritischem Denken und einer unbegrenzten Kreativität. Tim Cook hat eine konkrete Vorstellung über die digitale Abschlusskompetenz von Schüler*innen: *„Wenn man mit der Schule fertig ist, sollte man in der Lage sein, eine App zu entwickeln und sie veröffentlichen zu können“* (MANSHOLT, 2021). Auch die politischen Verantwortlichen in Österreich betrachten die Software-Entwicklung als eine entscheidende berufliche Fähigkeit für zukünftige Anforderungen. Tatsächlich wurde im März 2023 im BMBWF eine eigene Arbeitsgruppe eingerichtet, die mit der Umsetzung eines eigenen Unterrichtsfachs „Programmieren“ ab der 5. Schulstufe beschäftigt war. Der damalige Staatssekretär Florian Tursky (ÖVP) sah darin eine große Chance, das notwendige IT-Wissen bereits in der Sekundarstufe 1 zu generieren und später ausbauen zu können. Im Endeffekt soll damit der bereits existierende bzw. in Zukunft zu erwartende Fachkräftemangel bei informatischen Berufen gemeistert werden können (ORF, 2023).

4.3.6 EU Code Week

Die Initiative EU Code Week ist ein von der EU unterstütztes Projekt zur besseren Umsetzung des digitalen Binnenmarktes. Ziel der Initiative ist es, „*mehr jungen Menschen die Grundlagen des Programmierens und des computerorientierten Denkens beizubringen*“ (CODE WEEK, 2023a). Dabei wendet sich Code Week primär an Schulen und deren Lehrpersonen. Kostenfreie Veranstaltungen und Lehrmaterialien werden zur Verfügung gestellt und auch per Website veröffentlicht. Auf einer interaktiven Europa-Karte können einzelne Coding-Projekte von Schulen eingesehen werden (CODE WEEK, 2023b). Die Erfassung der programmiertechnischen Artefakte der Schulen erfolgt über die Initiative eEducation Österreich. Die Veröffentlichung auf der EU-Code-Week-Webseite generiert sich automationsunterstützt (CODE WEEK, 2021).

4.3.7 Micro:bit - Das Schulbuch

Im Jahr 2018 wurde erstmalig in Österreich ein Schulbuch als *Open-Educational-Resources* (OER)-Version veröffentlicht und unter der Lizenz von *Creative Commons* (CC) mit Namensnennung lizenziert. 2022 folgte die 2. Auflage, wieder als CC- und OER-Version. Alle Inhalte sind kostenfrei nutzbar und können beliebig heruntergeladen bzw. weitergegeben werden. In beiden Werken sind programmiertechnische Projekte beschrieben. Die Lösungen können gesammelt online eingesehen werden. Das Buch ist kein reines Informatik-Fachbuch, es beschreibt 39 einzelne Maker-Projekte mit Bastelelementen und der visuellen Programmiersprache MakeCode (siehe Kapitel 5.6.3) der Micro:bit Educational Foundation (EEDUCATION, 2022). Beide Schulbücher waren Inhalt und Themen von zahlreichen Lehrer*innen-Workshops und erfreuen sich in Österreich großer Beliebtheit. In der Sekundarstufe 1 ist der Micro:bit auch der beliebteste und bekannteste Mikrocontroller und dient als wichtiges Lehrmittel für das Schulfach DGB. Zudem ist er der einzige Mikrocontroller, der als Schulbuchersatz als „*Unterrichtsmittel eigener Wahl*“ über das offizielle Schulbudget angekauft werden kann (SCHULBUCHAKTION, 2024).

4.3.8 EU-Initiative “Important Project of Common European Interest” (IPCEI)

Eines der bedeutendsten technologischen Projekte der letzten Jahrzehnte ist die EU-Initiative IPCEI. Unter starker Beteiligung Österreichs soll europaweit die Halbleiterproduktion innerhalb der EU vorangetrieben werden. Hintergrund des Projekts ist die derzeit starke Abhängigkeit Europas von asiatischen Zulieferanten und Produzenten, was vor allem bei den Lieferengpässen aufgrund der Corona-Krise in den Jahren 2021 und 2022 offenkundig wurde. Österreich wird beim EU-Projekt direkt von den Unternehmen AT&S, AVL, EEMCO, Infineon Austria und NXP Semiconductors Austria vertreten, alleamt starke Elektronik-Stakeholder des heimischen Marktes. Was diese Initiative besonders macht, ist aber nicht nur der Versuch, die Microchip-Technologie bzw. deren Produktion und das notwendige Know-how in Europa zu situieren, sondern auch die Chance, in Europa bzw. Österreich direkt grüne Technologie verbauen zu können.

„Das „Important Project of Common European Interest“ (IPCEI) im Bereich Mikroelektronik und Kommunikationstechnologien umfasst insgesamt 100 Unternehmensprojekte aus 20 europäischen Staaten, die sich zur Aufgabe gemacht haben, Europa grüner, digitaler und resilienter zu machen.“ (BMK INFOTHEK, 2023)

Auch der gesetzeskonforme Rahmen dieses kartellähnlichen Zusammenschlusses von EU-Ländern und Unternehmen steht mit dem „Europäischen Chip Gesetz“ (European Chip Act) kurz vor der Umsetzung (EUROPÄISCHE KOMMISSION, 2023) (BUNDESKANZLERAMT, 2023). In einem Radiointerview mit ORF-OE1 nannte der Geschäftsführer von AT & S, Andreas Gerstenmayer, die wichtigsten Komponenten für den Erfolg des EU-Projekts: *„Digitalisierung, Elektrifizierung und Durchgängigkeit von naturwissenschaftlichen Ausbildungen“* (OBERMAYR, 2023 ab 12:30).

Naturwissenschaftliche Kenntnisse sind also wichtige Teile der schulischen Bildung bzw. der späteren beruflichen oder universitären Ausbildung und Forschung.

4.3.9 STEM- bzw. STEAM-Initiative

Das Kunstwort *STEM* steht für die Fachbereiche Science, Technology, Engineering und Mathematics. Ergänzend und aktuell spricht man von STEAM, welches STEM um den Bereich Arts ergänzt. Generell kann STEM bzw. STEAM auch als internationaler Ansatz

der MINT-Thematik in Österreich gesehen werden und findet sich auch im österreichischen Hochschulplan 2030 wieder (BUNDESMINISTERIUM FÜR BILDUNG, WISSENSCHAFT UND FORSCHUNG, 2022).

5. Mikrocontroller und Programmiersprachen

Zur Umsetzung der praktischen Programmierarbeit dieser Masterarbeit wird nicht ein Mikroprozessor, sondern ein Mikrocontroller gewählt. Die Gründe für diese Wahl wurden im Kapitel 3.9 erläutert, hauptsächlich soll aber die Notwendigkeit eines Betriebssystems bei Mikroprozessoren vermieden werden. Daher werden auch die populären Mikroprozessor-Boards Raspberry Pi 3 bzw. 4 nicht berücksichtigt. Die folgenden Kapitel beschreiben mögliche Mikrocontroller, wobei bei der vorliegenden Vorauswahl die Kriterien Anschaffungspreis, Dokumentationen, didaktische Erfahrungen aus dem Schulbereich, Bekanntheitsgrad und Ausstattung berücksichtigt wurden. Als Zielgruppe der Anwender*innen werden Anfänger*innen sowohl bei den Lehrkräften als auch bei Schüler*innen definiert. Die endgültige Fixierung des einzelnen Mikrocontrollers bzw. der Programmiersprache zur Umsetzung der Praxisprojekte erfolgt im Kapitel 7.

5.1 Arduino Uno R3

Einer der populärsten Mikrocontroller ist der Arduino Uno R3. Die Programmierung erfolgt durch eine Mischung aus den Programmiersprachen C++ und C, damit ist auch objektorientiertes Programmieren möglich. Pädagogisch sehr gut aufbereitet ist die Website von Arduino, die sogar ein Programmier-Tool über die Cloud kostenlos zur Verfügung stellt. Auch eine lokale Entwicklungsumgebung als *Integrated Development Editor* (IDE) steht für die Betriebssysteme Windows, macOS und Linux kostenfrei zur Verfügung (ARDUINO, 2023a). Arduino ist ein Open-Source-Projekt und wird laufend weiterentwickelt. Für einige Arduino Boards wird mittlerweile auch die Programmiersprache MicroPython samt eigenem Editor *OpenMV* angeboten, nicht aber für den Arduino Uno R3 (ARDUINO, 2023b). Das Board selbst kostet ca. 30 Euro ohne Zubehör, eine Version 4 mit WiFi-Modul liegt bereits bei über 50,00 Euro. Eine umfangreiche Online-Dokumentation und zahlreiche Online Tutorials ermöglichen einen einfachen Einstieg in die Programmierung (ARDUINO, 2023c).

Der Mikrocontroller hat einen ATmega328P-Prozessor mit einem CPU-Takt von 16 MHz. Dazu kommen noch 20 Anschlüsse als *General Purpose-Input/Output* (GPIO), wovon

sechs davon analog sind. Es fehlen jedoch Module für Sound, Bluetooth und WiFi (ARDUINO, 2023d).

5.2 Arduino Nano RP2040 Connect

Das neue Produkt Arduino Nano RP2024 Connect bringt die beiden Welten von Raspberry Pi und Arduino zusammen. Das Board selbst ist ein Arduino, der aber durch den RP-Prozessor RP2040 Dual-Core ARM Cortex M0+ mit 133 MHz gesteuert wird. Das Board gibt es mit und ohne Headerpins. Die Ausstattung ist sehr umfangreich. Auf dem Board existieren 30 I/O-Pins, eingebettet sind ein Mikrofon, WiFi, Bluetooth und sogar ein Temperatursensor (ARDUINO.CC, 2023). Neben der Programmiersprache C/C++ für Arduino kann das Board auch mittels MicroPython programmiert werden. Der Preis liegt bei ca. 30 Euro ohne Zubehör. Wie auch beim Arduino Nano R3 (siehe Kapitel 5.1) bietet die Website von Arduino umfangreiche Installations- und Lernunterlagen (ARDUINO, 2023e).

5.3 ESP32-Mikrocontroller einzeln oder als CyberPi

Einer der günstigsten und zugleich leistungsstärksten Mikrocontroller ist der ESP32, der auf seinem Board gleich mehrere Built-in-Module mitliefert. Gemäß Datenblatt des Herstellers Espressif (2023) beinhaltet das Board eingebaute WiFi- und Bluetooth-Komponenten. Der Prozessor ist ein Xtensa Single- oder Dual-Core mit 32 Bit. Jeder der Kerne hat eine Taktrate von 240 MHz. Für einen LAN-Schluss steht ein *Media Access Controller* (MAC) für Ethernet zur Verfügung, der aber standardmäßig ohne physischen Anschluss ausgeliefert wird. 34 programmierbare PINs stehen zur Auswahl. Zehn davon sind sogenannte *Touch Pins*, die auch auf eine Fingerberührung reagieren (ESPRESSIF, 2023). Für den unmittelbaren Einsatz im Schulunterricht sollten auf alle Fälle PIN-Header (Stiftleisten) vorhanden sein. Sollten diese fehlen, müssten die Stiftleisten noch aufgelötet werden. Auch verbaute Sensoren fehlen am Board, darum auch der günstige Preis von ca. 10 Euro.

Der Mikrocontroller ESP32 wird oft auf ein weiteres Board mit mehr Anschlüssen und Sensoren verbaut. Ein derartiges Erweiterungsboard des ESP32 ist beispielsweise das *Makeblock CyberPi Go Kit*. Vertrieben und hergestellt wird das Board vom Unternehmen

Makeblock, einem chinesischen Technologieunternehmen mit Firmensitz in Shenzhen. Das Unternehmen fokussiert sich auf Robotik und Programmierung in der Schule, bekennt sich zur STEAM-Offensive (siehe Kapitel 4.3.9) und beliefert nach eigenen Aussagen 20 Millionen User*innen, 200.000 Schulen und hat weltweit über 2.000 Bildungspartner (MAKEBLOCK, 2023a).

Der oben angeführte CyberPi kann gesondert gekauft werden, ist aber beispielsweise auch in populären Mini-Robotern wie *mBot2* oder *mBot Neo* verbaut. Durch den Mikrocontroller ESP32 sind WiFi- und Bluetooth-Module implementiert, ergänzt wird der CyberPi durch verbaute Lautsprecher samt Mikrofon, fünf LED-Lichtstreifen, drei Button und sogar einem Joystick. Ebenso integriert ist ein Licht-, Beschleunigungs- und Gyroskop-Sensor. Die Ausgabe erfolgt über ein gesondertes, integriertes Farbdisplay (MAKEBLOCK, 2023b). Das Board ist mit knapp 80 Euro samt Erweiterungs-Shield für externe Komponenten nicht billig. Es gibt ausreichend technische und pädagogische Dokumentationen und Projektbeschreibungen, programmiert wird mit der VPL Makeblock (ähnlich wie Scratch oder MakeCode) oder mit Python oder MicroPython. Beide Lösungen sind online oder als kostenlose App verfügbar⁴.

5.4 Circuit Playground Express von Adafruit

Ein aktueller Mikrocontroller ist die Variante *Circuit Playground Express*, der sein rundes Board tatsächlich mit einem Maximum an Anschlüssen und Sensoren versehen hat. Bei Kosten von ca. 30 Euro bietet der Controller gemäß seiner Website (2023) zehn farblich beliebig steuerbare NeoPixel, Temperatur-, Licht- und Bewegungssensoren sowie ein eingebautes Mikrofon und Lautsprecher. Dazu kommen noch zwei Buttons und ein Schieberegler plus sieben Pads, die als Touch-Eingänge fungieren können. Die CPU ist ein ATSAMD21-ARM-Cortex-M0-Prozessor mit 48 MHz und ein SPI-Flashspeicher mit 2 MB. Vorhanden sind auch zwei USB-Anschlüsse: ein Micro-USB-Anschluss zum Programmieren und Debuggen und ein USB-Anschluss für Maus, Tastatur oder andere externe Geräte (ADAFRUIT INDUSTRIES, 2023).

Der Circuit Playground Express setzt wie auch andere Boards von Adafruit auf die neue

⁴ Online unter <https://www.makeblock.com/pages/software> (09.08.2023)

Programmiersprache CircuitPython (ein Derivat von Adafruit von MicroPython), mit deren Hilfe der Einstieg in die Programmierung weiter vereinfacht werden soll. Zusätzlich unterstützt wird auch MakeCode von Microsoft als VPL und es existiert eine Kompatibilität mit der Arduino-IDE (ADAFRUIT INDUSTRIES, 2023).

5.5 BBC-Micro:bit V2 Go

Wie alle anderen Micro:bit-Produkte ist auch das aktuelle Modell primär für den unmittelbaren Unterricht an Schulen konzipiert. Seit 2016 werden der Vertrieb und die Unterstützung der Lernmaterialien und Dokumentationen durch die Micro:bit Educational Foundation gewährleistet. Die ersten Micro:bits wurden sogar von der englischen Fernsehgesellschaft BBC landesweit verschenkt. Den Bildungsfokus formuliert die Organisation selbst wie folgt:

‘The Micro:bit Educational Foundation is a not-for-profit organisation that aims to inspire every child to create their best digital future.’ („Micro:bit“, 2023)

Der aktuelle Micro:bit V2 Go hat gemäß Datenblatt (2023) insgesamt 19 I/O-Pins, davon können 6 Pins analog genutzt werden. Unterstützt wird Bluetooth, aber kein WiFi. Neu sind das Radio-Protokoll, das eine Kommunikation mit mehreren Micro:bits unterstützt. Zudem sind sowohl Lautsprecher als auch Mikrofon intern verbaut. Die Ausgabe erfolgt über ein Display mit einer 5 x 5 LED-Matrix, was eine digitale Darstellung von einzelnen Zeichen, Zahlen oder Mustern ermöglicht. Ebenfalls auf dem Board implementiert sind Temperatur-, Bewegungs- und Magnetfeldsensoren. Interessant ist auch die Funktion von drei Touch-Pins, also drei Eingängen, die per Fingerdruck erkannt werden. Ebenfalls vorhanden sind zwei physische Buttons A und B bzw. ein virtueller Button in Form eines *Touch-Sensitiv-Logos*. Angetrieben wird der micro:bit V2 von einer Arm-Cortex-M4-32-Bit-CPU und einer Taktrate von 64 MHz (micro:bit, 2023).

Für die Programmierung des Micro:bit gibt es mehrere Möglichkeiten. Als visuelle Programmiersprache steht MakeCode von Microsoft zur Verfügung. Ähnlich wie bei der Visual Programming Language (VPL) Scratch können Befehle in Blockform mit Fehlermini-

mierung durchgeführt werden. Dies geschieht mit einem kostenfreien Online-Editor direkt über die Micro:bit Website⁵. Die VPL-Anweisungen können direkt per Click in die Programmiersprachen Python oder JavaScript transponiert werden, was den späteren Übergang zu einer textuellen Programmsprache erleichtert. Zudem kann per Online-Editor auch direkt in Python oder JavaScript programmiert werden. Davon zu trennen ist die gesonderte Website für einen reinen MicroPython-Editor⁶. Hier wird ausschließlich in MicroPython programmiert, eine optimierte Version von Python für Embedded Systems. Programmier-Schnittstellen des Micro:bit gibt es auch für externe Ressourcen wie Scratch, die mobile iOS/Android-Version für Micro:bit, eine Schnittstelle mittels Swift Playgrounds speziell für iOS-Geräte wie iPads und noch viele andere⁷.

Ein großer Mehrwert bei der Wahl des Micro:bit ist sicherlich der bereits vorhandene Bekanntheitsgrad des Gerätes an Österreichs Sekundarschulen über das frei verfügbare OER-Schulbuch⁸ (siehe Kapitel 4.3.7).

5.6 Programmiersprachen

Mikrocontroller können mit mehreren Programmiersprachen gesteuert werden. Die Auswahl reicht von reinen Assemblersprachen bis hin zu populären Hochsprachen wie C, C++, Python, JavaScript, MicroPython und neuerdings auch CircuitPython. Dazu kommen noch VPLs als grafische Programmiersprachen wie Scratch, MakeCode oder Mini-Bloc⁹. Im schulischen Kontext und hier vor allem beim Einstieg in die Mikrocontroller-Programmierung haben sich die Programmiersprachen C/C++ der Arduino Plattform und Python mit seinen Derivaten MicroPython und CircuitPython etabliert. Vor allem die Hochsprache Python hat sich in den vergangenen Jahren zur populärsten Programmiersprache der Welt entwickelt und führt auch aktuell das Tiobe Ranking für 2024 an¹⁰.

⁵ Siehe online unter <https://makecode.microbit.org/> (09.08.2023)

⁶ Siehe online unter <https://python.microbit.org/v/3> (09.08.2023)

⁷ Siehe online unter <https://microbit.org/code/#other-editors> (09.08.2023)

⁸ Downloadbar unter <https://microbit.eeducation.at/wiki/Hauptseite> (10.08.2023)

⁹ Downloadbar unter <http://blog.miniblog.org/> (10.08.2023)

¹⁰ Siehe online unter <https://www.tiobe.com/tiobe-index/> (12.07.2024)

5.6.1 Python

Python ist eine Hochsprache und plattformunabhängig. Die Sprache selbst gilt als einfach zu erlernen und ist deshalb als Einstiegsprogrammiersprache an Schulen sehr beliebt. Die Gründe für die Popularität sind einige Vereinfachungen, die Python im Vergleich zu anderen Hochsprachen bietet. Wie Kersken (2016) betont, schließen Sprachen wie Java oder C# Anweisungen mit einem Semikolon ab. Python verwendet hier nur einen Zeilenumbruch. Auch auf geschwungene Klammern (engl. *Parentheses*) verzichtet Python. Stattdessen werden *conditional Statements* oder Schleifen mit den entsprechenden Keywords begonnen und dann mit einem Doppelpunkt beendet. Die nachfolgenden Anweisungen zur Abfrage oder Schleife werden einfach eingerückt (*Intent*). Wird die Einrückung programmierbedingt beendet, folgen weitere Anweisungen ohne Einrückung. Ebenso einfach ist die Variablendeklaration in Python, ohne dass eine explizite Typendeklaration erforderlich ist. Python erkennt den Datentyp aufgrund des Wertes (KERSKEN, 2016, S. 529–534). Natürlich ist auch eine Objektorientierung möglich wie bei den anderen Hochsprachen Java oder C#. Eine besondere Bedeutung kommt den Funktionen in Python zu, da sie wiederkehrende Aufgaben bündeln und unter Verwendung einer Funktionsbezeichnung zusammen mit Übergabeparametern ausgeführt werden können. Funktionen treten selbstständig auf, können aber auch abhängig von bestimmten Datentypen und Instanzen (Objekten von Klassen) als Methoden genutzt werden (Joos, 2021). Bei Mikrocontrollern wird Python mittlerweile gänzlich durch die Python-Erweiterung MicroPython ersetzt. Die Gründe hierfür werden im Kapitel 5.6.1.1 erklärt. Aber der wahrscheinlich wichtigste Grund für die Popularität von Python liegt darin, dass es als die wichtigste Programmiersprache für Künstliche Intelligenz gilt.

5.6.1.1 MicroPython

MicroPython ist eine spezielle Implementierung von Python, optimiert für die Verwendung auf Mikrocontrollern und eingebetteten Systemen mit eingeschränkten Leistungsressourcen (ALSABBAGH, 2019, S. 8). Es kann die gesamte Syntax von Python verwendet werden, aus Performance-Optimierung-Gründen ist MicroPython ein „eingeschränktes“ Python. Es werden nicht alle Datentypen unterstützt und auch die Built-in-Module sind im Vergleich zu Python selbst beschränkt. Die folgende Abbildung 33 zeigt die Standard-

Datentypen und Module von MicroPython:

- Builtin types
 - Exception
 - bytearray
 - bytes
 - dict
 - float
 - int
 - list
 - str
 - tuple
- Modules
 - array
 - builtins
 - deque
 - json
 - os
 - random
 - struct
 - sys

Abbildung 33: Unterstützte Datentypen und Module von MicroPython (MICROPYTHON, 2023)

Auf einem typischen Mikrocontroller funktioniert MicroPython praktisch wie ein kleines Betriebssystem, das Benutzerprogramme ausführt und einen Befehlsinterpreter bereitstellt. *Read-Eval-Print-Loop* (REPL) ist eine interaktive Umgebung, bei der Code-Anweisungen direkt auf der Mikrocontroller-Unit (MCU) ausgeführt werden, entweder per Direktverbindung oder über das Internet als *WebREPL* (MIKROPYTHON, 2023). Es muss also nicht immer eine aktualisierte Version eines Programms auf das MCU hochgeladen werden. Es kann auch direkt in der MCU bearbeitet und verändert werden. Wichtig ist es nochmals zu erwähnen, dass nicht generell alle MCUs kompatibel mit MicroPython sind.

5.6.1.2 CircuitPython

CircuitPython ist eine Variante von MicroPython, die sich speziell auf die Entwicklung von elektronischen Schaltungen und Hardwareprojekten fokussiert. Entwickelt wird es vom Elektronikunternehmen Adafruit, um den Einstieg in die Programmierung von Mikrocontrollern und elektronischen Komponenten zu erleichtern. Dies wird über eine hö-

here Abstraktionsebene und eine einfachere Handhabung umgesetzt. CircuitPython enthält fix implementierte Bibliotheken und Funktionen, die den Umgang mit Sensoren, Aktoren und anderen elektronischen Bauteilen herstellerübergreifend erleichtern. Viele aktuelle Modelle von MCUs beinhalten bereits vorinstalliertes CircuitPython. Nach dem Anschließen der MCU am Endgerät der User*innen wird die MCU als USB-Gerät erkannt und es kann sofort mit einem Editor programmiert werden. In anderen Fällen muss oft ein Treiber vorab installiert werden (ADAFRUIT, 2023).

5.6.2 C/C++ bzw. Arduino-C

C und C++ sind zwei eigenständige, populäre Programmiersprachen. Da es um die Programmierung von MCUs für schulische Zwecke geht, soll sich dieses Kapitel auf Arduino-C konzentrieren. Tatsächlich wird auf der Arduino-Plattform vorwiegend in C++ programmiert. Allerdings könnten reine C-Entwickler*innen sich auch ausschließlich für C entscheiden. Die Arduino-Plattform baut wie erwähnt auf C++, was üblicherweise als *Arduino-C* bezeichnet wird. Die wichtigsten Funktionen, Variablen, Konstanten, Operatoren und Schleifen finden sich gut dokumentiert auf der Webseite von Arduino (ARDUINO, 2023f). Erweiterungen werden bei C/C++ über den Import von Bibliotheken durchgeführt (ARDUINO, 2023g).

5.6.3 Visuelle Programmiersprachen VPL

Keine eigentliche Programmiersprache selbst sind visuelle Programmiersprachen. Hier werden einzelne Komponenten bzw. Funktionen blockbasiert dargestellt und können per Drag and Drop aktiviert werden. Der große Vorteil einer VPL ist die Tatsache, dass sie eine enorme Erleichterung in den ProgrammierEinstieg bietet, da Befehle und Anweisungen mit visuellen Elementen durchgeführt werden können. Syntaxfehler wie bei textuellen Programmiersprachen sind nicht möglich. Ein Nachteil besteht darin, dass ab einer gewissen Komplexität des Programmcodes die visuellen Blockelemente schnell unübersichtlich werden. Populäre Plattformen für VPLs, speziell für Physical Computing, sind MakeCode¹¹ von Microsoft und mBlock¹² von MakeBlock (siehe Kapitel 5.3). Vor

¹¹ Online verfügbar unter <https://www.microsoft.com/en-us/makecode> (abgerufen am 02.07.2024)

¹² Online verfügbar unter <https://ide.makeblock.com/> (abgerufen am 02.07.2024)

allem an Schulen ist MakeCode sehr bekannt, da es beispielsweise direkt im Micro:bit-Online-Editor integriert ist, eine direkte Unterstützung für den populären Schulungsroboter EV3 von Lego bietet und auch eine visuelle Programmierung von Minecraft Education ermöglicht (MICROSOFT, 2024).

II. Praxisprojekte

Für die direkte Erprobung mit interessierten Lehrpersonen wurden dreizehn konkrete Projekte mit Physical Computing vorab erstellt. Tatsächlich wurden in den beiden Schulungen aus Zeitgründen nur zehn Projekte abgeschlossen. Die Praxisprojekte generieren ein aufbauendes Grundlagenwissen über Technische Informatik bzw. Digitaltechnik gemäß Kapitel 3. Das jeweilige Projekt ist durch eine genaue Projektbeschreibung dokumentiert, lässt aber immer auch Spielraum für eigene Adaptierungen im Sinne von Physical Computing (siehe Kapitel 2.4).

6. Beschreibung der Praxisprojekte

Insgesamt werden zehn Praxisprojekte erstellt, dokumentiert und mit einer ganztägigen Fortbildung (10 Unterrichtseinheiten zu je 50 Minuten) für Lehrpersonen durchgeführt. Während der Fortbildung selbst werden alle Projekte in Einzelschritten erklärt bzw. gemeinsam erstellt und durch die Teilnehmer*innen nachgebaut bzw. programmiert. Zusätzlich gibt es für alle zehn Praxisprojekte eine Detailbeschreibung samt Erklärung der technischen Grundlagen im Format *Portable Document Format* (PDF) bzw. als (vorab) gespeicherte Programmierlösungen im MakeCode- bzw. MicroPython-Format. Die Teilnehmer*innen verfügen am Ende sowohl über selbst erstellte, digitale Projektdateien als auch über offizielle Lösungsdateien.

Die Projekte haben in der Beschreibung und in der Schulung selbst einen identen Schulungsaufbau. Zuerst wird mithilfe des kostenfreien Online-Tools TinkerCAD der Schaltungsaufbau inklusive des virtuell integrierten Mikrocontrollers Micro:bit geplant, erstellt und simuliert. Es werden nur Praxisprojekte mit relevanten Themen der Digitaltechnik durchgeführt, bei denen die benötigten elektronischen Komponenten virtuell in

TinkerCAD zur Auswahl stehen. Im zweiten Schritt werden das Schaltungsdesign physisch nachgebaut (siehe Making im Kapitel 2.3) und die elektronischen Komponenten miteinander verbunden. Der letzte Schritt ist die visuelle und textuelle Programmierung der gewünschten digitalen Funktion.

6.1 Projekt 1 – Reihen- und Parallelschaltung

Das Projekt 1 (siehe Anhang Kapitel 13.2) bietet einen Einstieg in das Grundverständnis eines Stromkreises (siehe thematisch Kapitel 2.1). Dabei fließt der Strom mit einer konstanten Spannung von 3,3 Volt vom Adapter des Micro:bit zuerst lediglich durch einen Vorwiderstand und eine LED zurück zur Masse (Ground). Dies ist eine einfache Reihenschaltung. Die Parallelschaltung der folgenden Abbildung 34 erweitert die Reihenschaltung zu einer Parallelschaltung mit zwei LEDs und zwei Vorwiderständen. Auf die Verwendung eines digitalen PINS am Micro:bit wird verzichtet, da hier keine gesonderte Funktion programmiert wird. Abbildung 34 zeigt die Planung mit TinkerCAD:

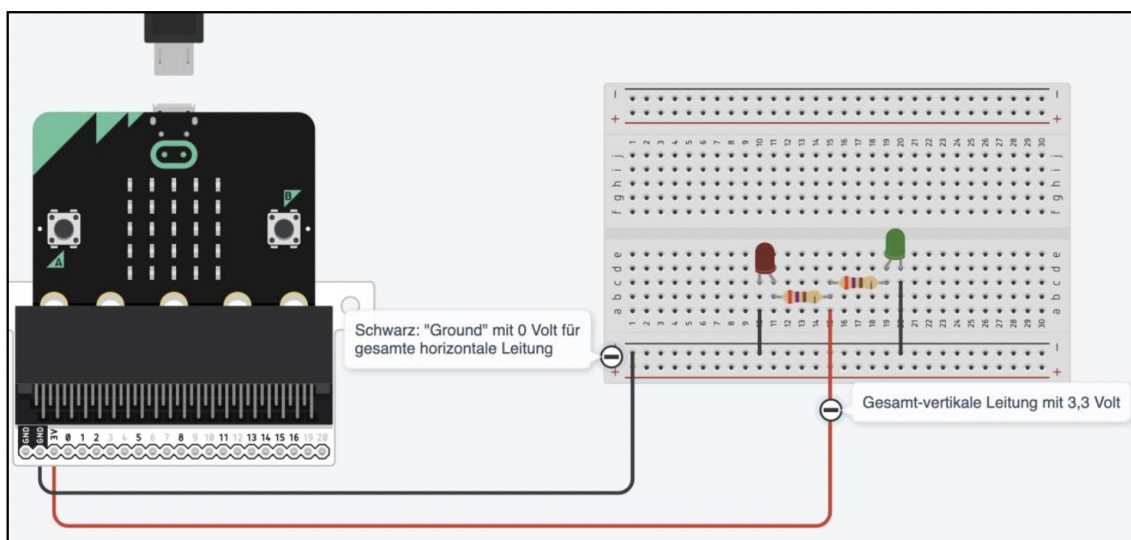


Abbildung 34: Parallelschaltung mit 2 LEDs (eigene Abbildung)

6.2 Projekt 2 – Stromschalten mit PIN

Das Projekt 2 (siehe Anhang, Kapitel 13.3) verwendet denselben Schaltungsaufbau wie

Projekt 1 (siehe thematisch Kapitel 2.1). Allerdings wechselt die Spannung zwischen 3,3 Volt und 0 Volt, ausgelöst durch den digitalen Pin 0, der visuell und textuell entsprechend programmiert wird. Die Abbildung 35 zeigt die Anbindung an Pin 0, der programmierbedingt die Spannung ändert und damit die LED ein- bzw. ausschaltet:

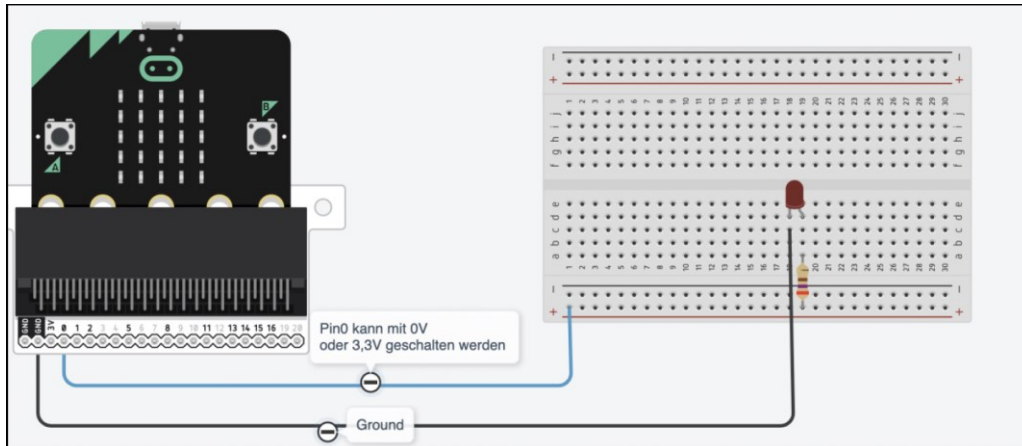


Abbildung 35: Spannungsänderung über Pin 0 (eigene Abbildung)

6.3 Projekt 3 – Ampelschaltung

Das Projekt 3 (siehe Anhang Kapitel 13.4) schließt nahtlos an das Projekt 2 an (siehe thematisch Kapitel 2.1). Neben dem Pin0 kommen auch die digitalen Pin 1 und Pin 2 zum Einsatz. Abwechselnd werden durch drei in Reihe geschaltete Stromkreise drei farbige LEDs zum Leuchten gebracht, vergleichbar mit einer Verkehrsampel. Die Abbildung 36 zeigt die programmiertechnische Umsetzung mit der visuellen Programmiersprache MakeCode:



Abbildung 36: Ampelschaltung mit 3 digitalen PINs (eigene Abbildung)

6.4 Projekt 4 – Potentiometer

Das Projekt 4 (siehe Anhang Kapitel 13.5) thematisiert das elektronische Bauteil *Potentiometer*, also die Möglichkeit, einen Stromwiderstand variabel zu steuern. Mit Hilfe des Potentiometers kann eine Eingangsspannung von 3,3 Volt auf bis zu 0 Volt gesenkt werden. Dies geschieht durch einen beweglichen Regler am Potentiometer mit gleichzeitiger Erhöhung oder Senkung des inneren Widerstands. Dabei wird erstmalig der analoge Eingang des Pin 0 verwendet, der durch einen im Micro:bit eingebauten Analog-Digital-Wandler (ADC) ganzzahlige Werte zwischen 0 und 1023 wiedergibt (siehe Thema ADC im Kapitel 3.4). Durch eine rechnerische Umwandlung kann schlussendlich der tatsächlich gemessene Spannungswert abgelesen werden. Die folgende Abbildung 37 zeigt die Programmierung der Ausgabe des Spannungswertes in der Programmiersprache Micro-Python:

```
1  from microbit import *
2  # Variable potWert
3  potWert = 0
4  # Variable Volt
5  volt = 0
6
7
8  while True:
9      # analoges Einlesen der Spannung
10     potWert = pin0.read_analog()
11     display.show("Analog:" + str(potWert))
12     sleep(500)
13     # Umrechnung der ADC-Werte 0-1023 in Volt
14     volt = (3.3 / 1023) * potWert
15     # laufende Anzeige beim Drehen am Poti
16     display.show("Volt:" + str(volt))
17     sleep(500)
18
```

Abbildung 37: Einlesen und Ausgabe der Spannung (eigene Abbildung)

6.5 Projekt 5 – Pulsweiten-Modulation (PWM)

Das Projekt 5 (siehe Anhang Kapitel 13.6) thematisiert die Pulsweiten-Modulation (siehe thematisch Kapitel 3.4). Im Ausgang hat der Micro:bit lediglich digitale, aber keine analogen PINs. Analoge Pins könnten ein digitales Ausgangssignal über einen Digital-Analog-Wandler in einen beliebigen, kontinuierlichen Spannungswert umwandeln, der jedoch bei Micro:bit-PINs nicht vorhanden ist. So werden tatsächlich nur „1“ und „0“ in einem bestimmten Tastverhältnis digital geschaltet. Dabei entspricht der Wert 1023 des Tastverhältnisses dem Verhältnis 100 %, der Wert 511 wäre ca. 50 % des Tastverhältnisses. Je näher also der Wert zu 1023 tendiert, desto „heller“ leuchtet die LED. Beim Projekt 5 wird somit die Helligkeit einer LED nicht über eine Spannungsänderung bewirkt, sondern mit dem Einsatz von PWM unter schneller Schaltung von „1“ und „0“ mit dem Effekt einer langsamen Zu und Abnahme der Helligkeit. Die folgende Abbildung 38 zeigt den Programmier-Code in MicroPython, wobei das Tastverhältnis durch Werte von 0 bis 1023 geregelt wird:

```

1  from microbit import *
2
3  # Schleife mit 5 Durchgängen
4  for x in range(5):
5      # brightness stellt das Tastverhältnis von PWM dar
6      # bei niedrigen Werten dominieren LOW, bei hohen Werten HIGH
7      # LED wird langsam heller
8
9      display.show(Image.HAPPY)
10     # Hochzählen, LED wird heller
11     for brightness in range(0, 1023):
12
13         pin1.write_analog(brightness)
14         # Achtung: bei sleep wird nur der Code unterbrochen, nicht die Ausführung
15         # somit geniert sleep(5): 5 Millisekunden/1 Mikrosekunde = 5000 Perioden
16         sleep(5)
17     # Pause vor dem Wechsel
18     sleep(2000)
19     # LED wird langsam dunkler
20     display.show(Image.SAD)
21     for brightness in range(1023, 0, -1):
22
23         pin1.write_analog(brightness)
24         # 5 millisekunden für Generierung der Perioden
25         sleep(5)
26     # Pause vor dem Wechsel
27     sleep(2000)
28 # Ende der Lichtshow
29 display.show(Image.ASLEEP)
30 sleep(2000)
31 display.clear()

```

Abbildung 38: Programmcode für eine PWM-Modulation in Python (eigene Abbildung)

6.6 Projekt 6 – Binärschaltung

Projekt 6 (siehe Anhang Kapitel 13.7) visualisiert die Darstellung von Binärzahlen (siehe Kapitel 3.3) durch 3 LEDs. Die LEDs werden von rechts nach links gelesen (LSB), wobei das Leuchten der LED „1“ und das Nichtleuchten der LED „0“ repräsentieren. Leuchten alle LEDs gleichzeitig, wäre das die Zahl 7, leuchtet nur die rechte LED, wäre es die Zahl 1. Wenn der Button A des Micro:bit gedrückt wird, wird von 0 bis 7 automatisch hochgezählt und der Wert am Display dargestellt. Die gleichzeitige, binäre Darstellung der aktuellen Zahl wird durch die Aktivierung der LEDs repräsentiert und kann von rechts nach links binär interpretiert werden. Die folgende Abbildung 39 zeigt den Programmiercode in MicroPython:

```
1 from microbit import *
2 # Anordnung der Pins von links nach rechts
3 pinRow = [pin2, pin1, pin0]
4
5 while True:
6     if button_a.is_pressed():
7         # zaehler von 0 bis 7
8         zaehler = 0
9         for i in range(8):# inklusive 0 bis 7
10            # Anzeigen des aktuellen numerischen Werts von zaehler
11            display.scroll(str(zaehler))
12            # Konvertieren von 'zaehler' in einen 4-stelligen String
13            # format-Funktion :03b erzwingt eine dreistellige Binärdarstellung
14            binaerWert = '{:03b}'.format(zaehler) # binaerWert ist ein String
15            # Schleife wird pro Zahl 3 mal durchlaufen von 0 bis 2
16            for i in range(3):
17                # Zugriff auf das i-te Bit des Binärstrings
18                element = binaerWert[i] # ist ein String
19                # Schreiben der Werte 1 und 0 in die 3 Pins
20                pinRow[i].write_digital(int(element))# Element wird int-Zahl
21            # nächste Zahl von 0 bis 7
22            zaehler += 1
23            # Warten für 1,5 Sekunde für nächste Zahlendarstellung
24            sleep(1500)
25            # OK Zeichen nach der Zahl 7
26            display.show(Image.YES)
```

Abbildung 39: Listing zur binären Darstellung durch Leuchten der LEDs (eigene Abbildung)

6.7 Projekt 7 – interner Lichtsensor

Beim Projekt 7 (siehe Anhang Kapitel 13.8) wird der interne Lichtsensor des Micro:bit genutzt. Dieser befindet sich direkt in den 25 LEDs auf der Vorderseite des Micro:bit. Die gemessenen Werte gehen von 0 (dunkel) bis 255 (ganz hell) und werden softwaremäßig bestimmt. Es gibt derzeit keine genaue Beschreibung in der Micro:bit-Dokumentation, aber vermutlich handelt es sich um einen Analog-Digital-Converter (ADC) mit 8 Bits, der softwaremäßig die Lichtgrenzen bestimmt (siehe thematisch Kapitel 3.4). Dabei wird die Lichtstärke nicht in Lumen, sondern als digitale Zahl zwischen 0 und 255 gemessen. Ist der gemessene Wert kleiner als 50, wird zusätzlich eine LED per digitalem Pin 0 geschaltet. Ist der Wert höher als 50, erfolgte kein Einschalten der LED. Das Symbolbeispiel wäre eine einfache Beleuchtung, die sich bei Dunkelheit automatisch einschaltet. Die folgende Abbildung 40 zeigt den visuellen Programmcode mit MakeCode:

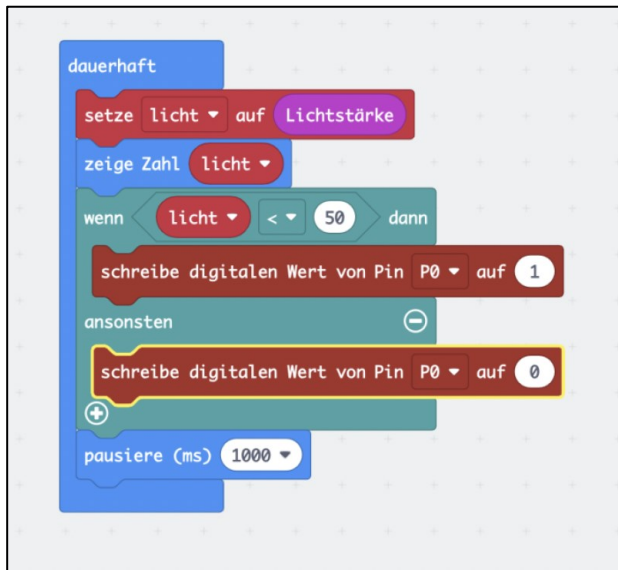


Abbildung 40: Messung der Lichtstärke mit Schalten einer LED (eigene Abbildung)

6.8 Projekt 8 – LED-Stripes (NeoPixel)

Das Projekt 8 (siehe Anhang, Kapitel 13.9) thematisiert den Aufbau und die Programmierung eines LED-Streifens (NeoPixel) mit insgesamt 60 LEDs. Es war das beliebteste Projekt aller Schulungsteilnehmer*innen, wahrscheinlich auch im Hinblick auf den unmittelbaren Einsatz im Unterricht mit Schüler*innen. 60 blinkende LEDs sind für Lehrpersonen und Schüler*innen gleichsam beeindruckend. Deren Adaptierungen sind spannende Beispiele für Making und Tinkering (siehe thematisch Kapitel 2.3).

Der LED-Streifen muss zunächst ordnungsgemäß mit dem Breakboard des Micro:bit verbunden werden. Eine wichtige Rolle kommt dabei dem Pin 0 zu, über den die Leuchtanimationen schlussendlich programmiert werden. Alle Teilnehmer*innen erweiterten eigenständig und physisch die Leuchtketten durch Verknüpfen mehrerer LED-Stripes. Im Endeffekt konnten bis zu 180 LEDs angesteuert werden. Die Abbildung 41 zeigt den korrekten Anschluss der Leuchtkette am Micro:bit als Entwurf in TinkerCAD:

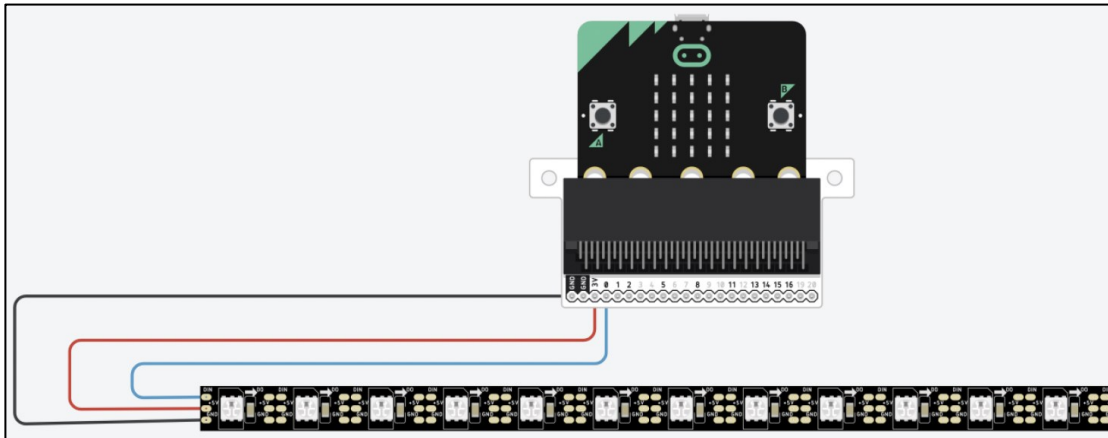


Abbildung 41: Entwurf des Anschlusses der Lichterkette am Micro:bit (eigene Abbildung)

6.9 Projekt 10 – Modul Radio

Das Projekt 10 (siehe Anhang Kapitel 13.10) thematisiert eine kabel- und USB-lose Übertragung von Informationen zwischen mindestens zwei Micro:bits. Die Übertragung erfolgt mit dem kabellosen Funkprotokoll *Radio* (siehe Kapitel 3.7.4). Damit die Übertragung auf einen Benutzerkreis eingeschränkt werden kann, werden Gruppen (0 bis 255) und Kanäle (0 bis 80) eingerichtet. Damit wird der Funkverkehr eingeschränkt. Die Übertragung erfolgt unverschlüsselt, sie ist also nicht sicher. Standardmäßig ist eine Übertragungseinheit mit 32 Bytes festgelegt, dies könnte aber auf bis zu 1024 Bytes erweitert werden. Der Funkabstand darf nur wenige Meter betragen.

Button A generiert bei einem Micro:bit dieses Projektes das Senden einer Botschaft, die dann vom Zweitgerät bzw. allen beteiligten Geräten empfangen werden kann. Die folgende Abbildung 42 zeigt die Funkübertragung vom linken, grün gekennzeichneten Micro:bit zum rechten, rot gekennzeichneten Gerät:

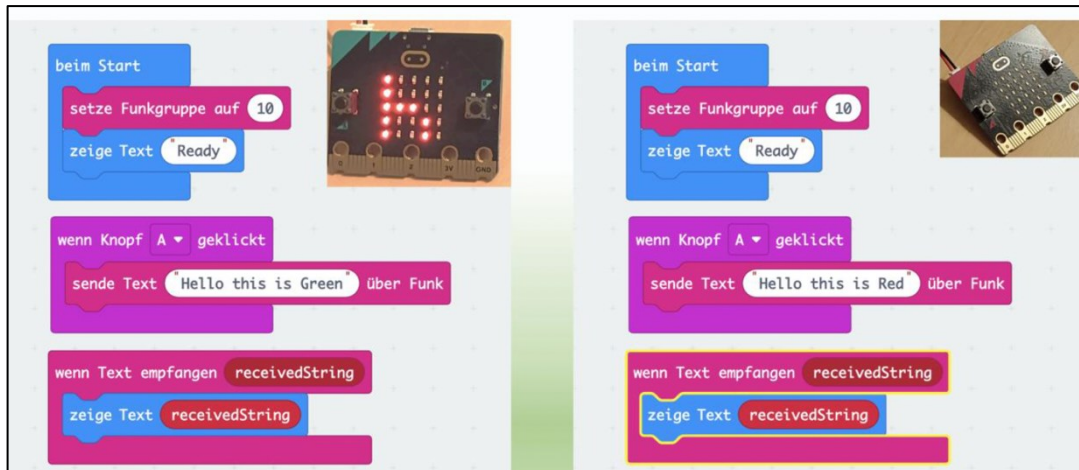


Abbildung 42: Übertragungsfunktion, programmiert in MakeCode (eigene Abbildung)

6.10 Projekt 11 - MOSFET

Das Projekt 11 (siehe Anhang Kapitel 13.11) fokussiert das Thema Halbleiter, dargestellt durch eine spezielle Art eines Transistors, den MOSFET (siehe Kapitel 3.5). Er besteht aus einem Gate (G), einer Source (S) und einem Drain (D). Wird am Gate eine positive Spannung angebracht, fließt zwischen Drain und Source ein Strom, da das positive Gate Elektronen anzieht und dadurch ein leitender „Kanal“ entsteht. Ohne Spannung verschwindet der Kanal. Dieser dient als Vermittlungsstelle zwischen seinen eigenen, äußeren Anschlüssen *Source* (links) und *Drain* (rechts). Das Gate im Projekt wird durch den Pin 0 des Micro:bit angesteuert. Sendet der Pin 0 eine „1“, wird das Gate unter Spannung gesetzt. Diese Spannung bewirkt, dass zwischen der MOSFET-Drain-Schaltung und der Source eine leitende Bahn entsteht. Es fließt ein Strom zwischen den äußeren Komponenten, der eine angeschlossene LED zum Leuchten bringt. Beim Pin-0-Zustand „0“ verschwindet diese „Bahn“ wieder und das Gate agiert dann als Isolator. Die LED leuchtet somit nicht. Die Abbildung 43 zeigt diese Schaltung durch die Verwendung eines MOSFET 2N7000:

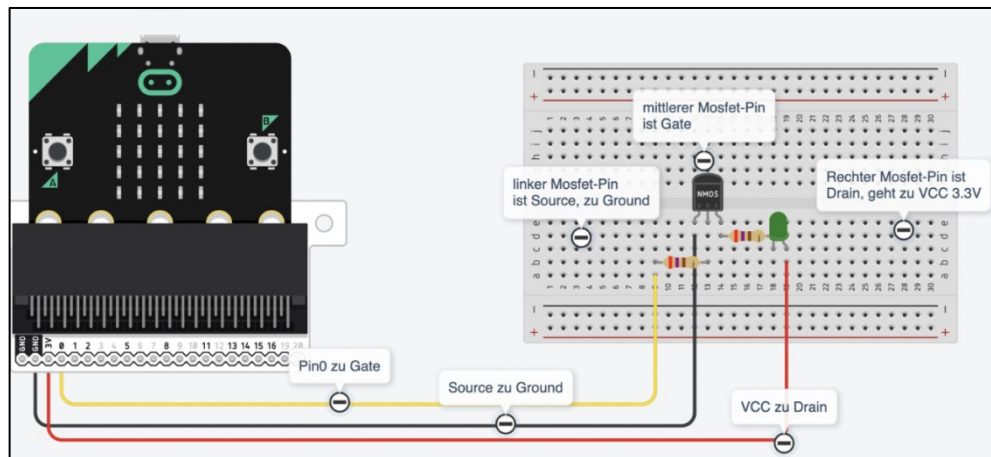


Abbildung 43: Schaltungsaufbau des MOSFET (eigene Abbildung)

III. Methodischer Teil

Der methodische Teil dieser Masterthesis thematisiert zuerst die Auswahl des Mikrocontrollers und der Programmiersprache. Der gewählte Mikrocontroller begründet in weiterer Folge die verwendete Schaltungssoftware zur visuellen Darstellung von elektronischen Schaltungen. Das Tool zur Schaltungssimulation muss auf alle Fälle mit dem gewählten Mikrocontroller kompatibel sein. Auch die Auswahl der Programmiersprache erfolgt in Abstimmung mit dem Mikrocontroller. In weiterer Folge werden vorab Schulungsunterlagen erstellt, die die didaktische Vermittlung von angewandter Digitaltechnik über Physical Computing ermöglichen sollen. Mit insgesamt zehn ausgearbeiteten Referenzbeispielen als Schulungsunterlagen werden zwei unterschiedliche Gruppen (Gruppe 1 und Gruppe 2) an Lehrpersonen geschult. Die Schulungstermine der beiden Gruppen sind verschieden. Beide Gruppen resultieren aus interessierten Lehrpersonen, die die Lehrbefähigung für „Digitale Grundbildung DGB“ berufsbegleitend über die Pädagogische Hochschule Vorarlberg als unterschiedliche Jahrgänge absolvieren.

Die erste Schulung mit Gruppe 1 fand mit 22 Teilnehmer*innen am 6.4.2024 beim Gymnasium Schoren in Dornbirn ganztätig statt. Gruppe 2 mit 15 Personen wurde am 27.04.2024, ebenfalls an der gleichen Örtlichkeit, geschult. Die Teilnehmenden der Gruppen waren nicht ident.

7. Auswahl Mikrocontroller, Programmiersprache, Schaltungssoftware

Für die Selektion der geeigneten Programmiersprache, der Schaltungssoftware und des Mikrocontrollers selbst wird ein bestimmter Ablauf gewählt. Eine wichtige Voraussetzung ist, dass einige Projekte auch in der SEK1 (AHS-Unterstufe bzw. Mittelschule) umsetzbar sein sollen. Im Zweifelsfall wird stets die einfachere Variante ausgewählt. Dies betrifft beispielsweise auch die Entscheidung zwischen Mikrocontroller oder Mikroprozessor. Wie in Kapitel 3.9 erwähnt, besteht der Hauptunterschied zwischen beiden Lösungen darin, dass ein Mikroprozessor ein eigenes Betriebssystem beinhaltet. Hier müsste man sich auch in das jeweilige Betriebssystem einarbeiten, bevor man zum ei-

gentlichen Physical Computing übergehen kann. Am Beispiel des in Kapitel 3.9 erwähnten Raspberry Pi wäre das üblicherweise eine Linux-Distribution, die sich von den in der Schule gewohnten Windows- bzw. iOS-Umgebungen (iPad) doch beträchtlich unterscheidet. Aus diesem Grund wird nur die Auswahl von Mikrocontrollern in Erwägung gezogen. Auch bei der Auswahl der Schaltungssoftware wird beachtet, dass diese auch in der SEK1 eingesetzt werden kann und nicht jedes elektronische Detail im Detail darstellbar sein muss.

Im ersten Schritt soll somit, aufgrund von selbst bestimmten Parametern, der am besten geeignete Mikrocontroller gewählt werden. Abhängig vom Mikrocontroller wird die verfügbare Auswahl an Programmiersprachen begrenzt sein, da nicht alle Mikrocontroller jede Programmiersprache unterstützen. Schlussendlich wird auch die Schaltungssoftware dahingehend beurteilt, ob der gewählte Mikrocontroller bzw. dessen GPIO direkt in der Software verfügbar und damit nutzbar sind.

7.1 Mikrocontroller

Die Auswahl des Mikrocontrollers erfolgte individuell nach den Kriterien Anschaffungspreis, Fokus auf den Schulbereich samt Dokumentation, spezielle Ausstattung, Unterstützung von VPLs und sonstige wichtige Mehrwerte. Die Tabelle 2 zeigt die Kriterien samt individueller Bewertung von „0“ bis „3“:

Tabelle 2: Kriterien für Mikrocontroller (eigene Tabelle)

Kriterium	Arduino Uno R3 (Kapitel 5.1)	Arduino Nano RP2040 Connect (Kapitel 5.2)	CyberPi als ESP32 (Kapitel 5.3)	BBC micro:bit V2 Go (Kapitel 5.5)	Adafruit Circuit Playground Ex- press (Kapitel 5.4)
Preis als Kom- plettset	30,00 – 50,00 Euro	30,00 Euro	80,00 Euro	30,00 Euro	30,00 Euro
Wertung	2*	3	1	3	3
Fokus auf Schule/Bil- dung/Lehr- materiali- en/Doku- mentationen	Open-Source-Pro- jekt, starke Schulorientierung, sehr gute Lehrun- terlagen über die Arduino Website	Open-Source-Pro- jekt, starke Schulori- entierung, sehr gute Lehrunterlagen über die Arduino Website und über die Rasp- berry Pi Foundation	Schwache Doku- mentation für Cy- berPi, sehr un- übersichtlich, teils lediglich chinesi- sche Dokumenta- tionen	Starke Schulorien- tierung, Vielzahl an Lernmaterialien (siehe auch Schul- buch als OER-Ver- sion für Micro:bit gemäß Kapitel 5.5)	Lehrmateriali- en/Dokumentati- onen vorhanden, kein ausschließli- cher Fokus auf Bil- dung und Schule
Wertung	3	3	0	3	1
Ausstattung	Es fehlen inte- griertes Display, Sound, Bluetooth und WiFi, CPU- Takt nur 16 MHz	Bluetooth und WiFi vorhanden, es feh- len integriertes Dis- play, Mikrophon und Lautsprecher; CPU- Takt mit 133 MHz	Geschlossenes Gehäuse, inte- griertes Display, Sound, Bluetooth und WiFi vorhan- den, CPU-Takt mit 240 MHz	Sehr kompaktes Bauteil, integriertes Display, Sound, Mik- rophon, Bluetooth und drahtloses Pro- tokoll <i>Radio</i> vorhan- den, CPU-Takt mit 64 MHz	Temperatur-, Licht- und Bewe- gungs-Sensor samt eingebau- tem Mikrophon und Lautsprecher sind vorhanden, CPU-Takte mit 48 MHz, Anschluss für Tastatur und Maus vorhanden, kein integriertes Display, 10 RGB- LEDs (Neopixels)
Wertung	1	1	3	3	2
Unterstüt- zung von Vi- sual Program- ming (VPL)	Keine direkte Un- terstützung des Herstellers, nur über Drittanbieter und teils kosten- pflichtig	Keine direkte Unter- stützung des Her- stellers, nur über Drittanbieter und teils kostenpflichtig	Gute Unterstüt- zung über Make- Block, sehr Scratch-orientiert	Gute Unterstützung über MakeCode von Microsoft, direkt in der Micro:bit-Web- site integriert	Gute Unterstüt- zung über Make- Code von Micro- soft
Wertung	1	1	2	3	2
Sonstiges	Gehört zu den am meist verkauften Mikrocontrollern weltweit	Kombiniert Arduino und Raspberry Pi, zwei sehr populäre Geräte	Vertrieben durch den weltweit größten Bildungs- anbieter (siehe Kapitel 5.3)	Hohe Popularität an österreichischen Schulen, kann statt einem Schulbuch als „ <i>Unterrichtsmittel eigener Wahl</i> “ über das Schulbudget ge- kauft werden (SCHULBUCHAKTION, 2024)	Noch wenig be- kannt
Bewertung	2	2	2	3	1
Gesamt	9	10	8	15	9
* 0 = keine Wertung, 1 = ausreichend, 2 = gut, 3 = sehr gut					

Als Zielgruppe der Mikrocontroller werden Anfänger*innen und fortgeschrittene User*innen sowohl bei den Lehrkräften als auch bei Schüler*innen definiert. Die Ergebnisse der eigenen Bewertung sind gemäß Tabelle 2 ersichtlich. Aufgrund der Evaluierung erscheint der Micro:bit als am besten geeignet, da er alle 15 von 15 Punkten erreicht. Die folgende Abbildung 44 zeigt wichtige Anschlüsse und PINs des gewählten Mikrocontrollers *Micro:bit*:

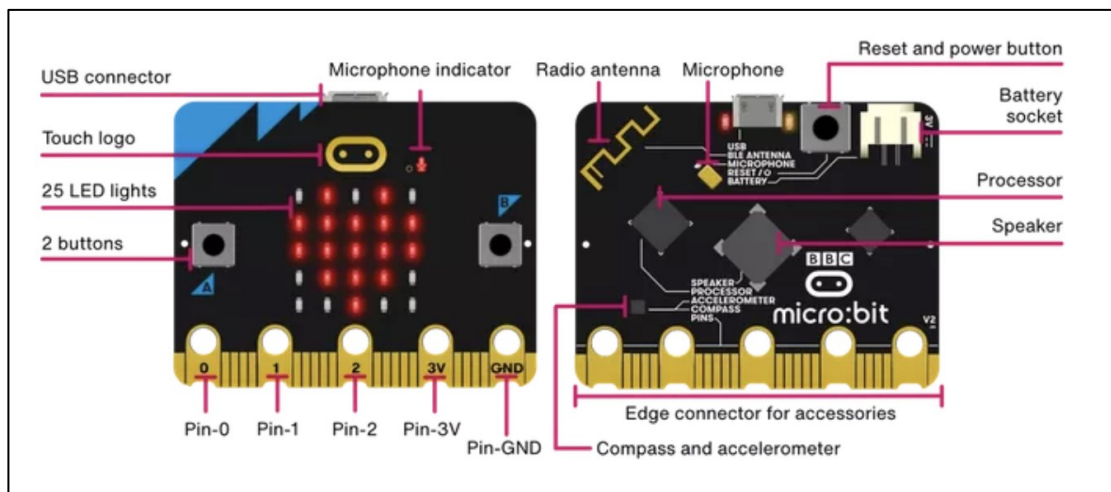


Abbildung 44: Micro:bit V2 mit wichtigen Komponenten (MICRO:BIT, 2024)

7.2 Programmiersprache

Die Wahl der Programmiersprache ergibt sich direkt nach der Auswahl des Micro:bit als geeignetem Mikrocontroller. Der Micro:bit bietet eine hervorragende Unterstützung für die visuelle Programmierung mit MakeCode¹³ (siehe Kapitel 5.6.3). Eine textuelle Programmierung ist mit den Programmiersprachen Python, JavaScript und MicroPython möglich. Da Python- und JavaScript-Code vor einem Upload auf das Gerät automatisch in MicroPython umgewandelt werden, erscheint hier die direkte Nutzung des nativen MicroPython logisch. MicroPython ist eine stark reduzierte Form von Python und wurde speziell für Mikrocontroller konzipiert (siehe Kapitel 5.6.1.1). Der Hersteller des Micro:bit bietet direkt auf seiner Website einen ausgezeichneten MicroPython-Editor¹⁴,

¹³ Siehe online unter <https://makecode.microbit.org/> (abgerufen am 04.07.2024)

¹⁴ Siehe online unter <https://python.microbit.org/v/3> (abgerufen am 04.07.2024)

der ohne Installation plattformübergreifend und kostenfrei genutzt werden kann. Weiters steht der ebenfalls kostenfreie, lokale Editor *Mu*¹⁵ zur Verfügung, wodurch keine Internetverbindung notwendig ist.

Somit fällt die Auswahl der Programmiersprache auf die visuelle Programmiersprache MakeCode (siehe Kapitel 5.6.3) und die textuelle Programmiersprache MicroPython (siehe Kapitel 5.6.1.1). Die folgende Abbildung 45 zeigt den MakeCode-Editor des Micro:bit als Weblösung:

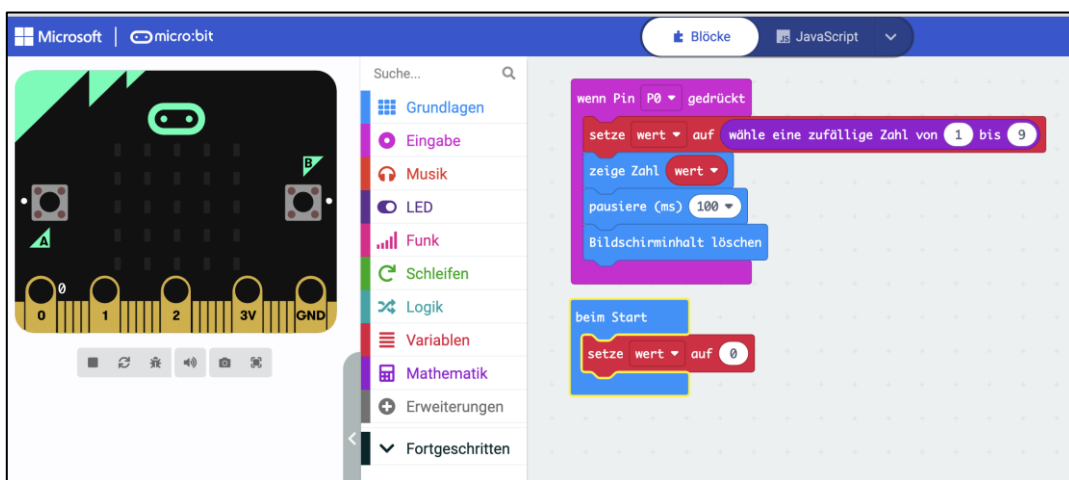


Abbildung 45: MakeCode-Editor für den Micro:bit (eigene Abbildung)

Abbildung 46 zeigt den Online-Editor für den Micro:bit in der Programmiersprache MicroPython:

¹⁵ Siehe online unter <https://codewith.mu/en/download> (abgerufen am 04.07.2024)

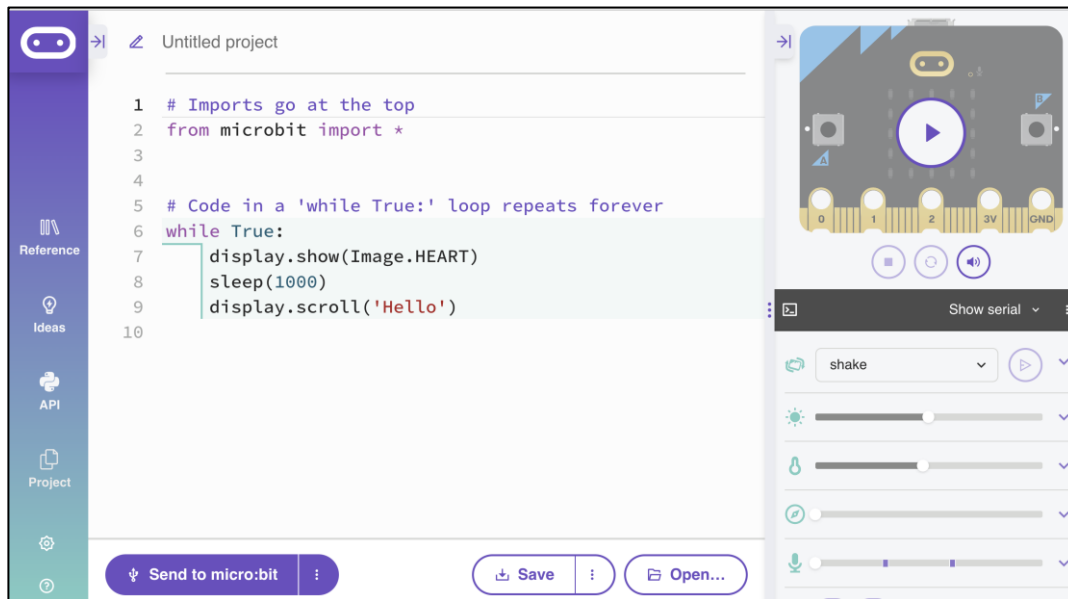


Abbildung 46: Online-Editor für MicroPython (eigene Abbildung)

7.3 Schaltungssoftware

Hier kommen die *Tools* TinkerCAD¹⁶, KiCad¹⁷ und Fritzing¹⁸ in die engere Auswahl. Fritzing und KiCad sind lokale Anwendungen, die zwar für Windows, macOS und Linux vorliegen, jedoch vorab installiert werden müssen und somit nicht online nutzbar sind. Zudem ist das Tool Fritzing kostenpflichtig. TinkerCAD ist im Schulbereich (auch in der SEK1) sehr populär und vor allem bei der 3D-Modellierung und im 3D-Druck häufig im pädagogischen Einsatz. Zudem ist es kostenlos und ohne Installation online nutzbar. Die eigene Evaluierung der Schaltungssoftware beurteilt das Tool TinkerCAD als höchst geeignet.

¹⁶ Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.07.2024)

¹⁷ Downloadbar unter <https://www.kicad.org/> (15.07.2024)

¹⁸ Downloadbar unter <https://fritzing.org/download> (15.07.2024)

8. Evaluation der Schulung für Gruppe 1 und Gruppe 2

Die übergreifende Forschungsfrage dieser Masterthesis beschäftigt sich konkret damit, ob ein Einstieg in die sehr komplexen Themen Technische Informatik bzw. Digitaltechnik mittels einer ganztägigen Fortbildung (für DGB- und Informatiklehrpersonen) mitsamt zehn dokumentierten Referenzprojekten und dem Ansatz von Physical Computing erfolgreich geschafft werden kann. Zur Beantwortung dieser Forschungsfrage werden sieben Thesen erstellt, deren Verifikation anhand einer Online-Umfrage mit sieben Evaluierungsfragen (Pflichtfragen) pro Projekt erfolgen soll. Drei weitere Fragen sind offene, optionale Fragen. Sämtliche teilnehmenden Lehrpersonen haben einfache Vorkenntnisse in einer visuellen Programmiersprache bzw. in Python. Ebenso absolvieren sämtliche Lehrpersonen die Schulung in Erwartung einer unterrichtsrelevanten Umsetzung ihrerseits im Pflichtfach DGB (SEK1) bzw. Informatik (SEK2). Die genaue Umsetzung an den Schulen ist jedoch offen. Sie kann inhaltlich zur Gänze oder nur auszugsweise erfolgen, da es hierfür im Lehrplan keine genauen Vorgaben gibt. Ebenso sind einige Beispiele für SEK1 und SEK2 geeignet, andere eher nur für die SEK1. Um Druck auf die Teilnehmenden der Schulung auszuschließen, wird in der Instruktion explizit mündlich festgehalten, dass bei keinem der zehn Projekte eine Verpflichtung zur Beantwortung der jeweils 10 Fragen pro Projekt vorliegt. Die Teilnahme an der Evaluierung ist anonym und freiwillig. Die Auswertung der Daten erfolgt methodisch anhand einfacher Häufigkeitszählungen samt Häufigkeitstabelle als deskriptive Statistik. Als Merkmale (Werte) werden fünf vorgegebene Werte definiert, die mit den Zahlen 1 bis 5 mittels Online-Umfrage qualitativ-ordinal vorliegen (SACHS, 2007, S. 11). Es liegt also eine Ordinalskala (Rangskala) vor, da die Merkmalsausprägungen in eine natürliche Rangordnung (Reihenfolge) gesetzt werden können (BOSCH, 1999, S. 5). Die grafische Umsetzung der Häufigkeitstabelle (SACHS, 2007, S. 15) als absolute Häufigkeit erfolgt als Stabdiagramm (Balkendiagramm) oder als Häufigkeitspolygon (BOSCH, 1999, S. 9).

8.1 Datenerhebung, Fragen und Thesen

Die Evaluationsdaten wurden am 06.05.2024 bzw. am 27.04.2024 direkt nach den Schulungen mit Lehrpersonen für die beiden berufsbegleitenden Befähigungslehrgänge erhoben ($n_1 = 22$, $n_2 = 15$). Die Fragen werden via Microsoft Forms (Office 365) organisiert.

Die folgende Abbildung 47 zeigt Teile der Umfrage aus Sicht der teilnehmenden Lehrpersonen:

The screenshot shows a Microsoft Forms survey titled "Physical Computing - Bsp01". The survey is titled "Physical Computing - Bsp01" and is saved. It contains three questions:

- 1. Die Kombination** von Design-Entwurf mit TinkerCAD, praktischem Aufbau am Steckbrett und die visuelle und textuelle Programmierung erleichtert den Einstieg bzw. das Verständnis für das Thema Computertechnik (Digital- und Analogtechnik).
[5 Sterne "sehr zutreffend", 4 Sterne "zutreffend", 3 Sterne "eher nicht zutreffend", 2 Sterne "nicht zutreffend", 1 Stern "weiß ich nicht"] *
- 2. Die visuelle Programmierung** mit MakeCode erleichtert den Einstieg in die Programmierung.
[5 Sterne "sehr zutreffend", 4 Sterne "zutreffend", 3 Sterne "eher nicht zutreffend", 2 Sterne "nicht zutreffend", 1 Stern "weiß ich nicht"] *
- 3. Die visuelle Programmierung** mit MakeCode erleichtert **das Verständnis** für die textuelle Programmierung mit MicroPython.
[5 Sterne "sehr zutreffend", 4 Sterne "zutreffend", 3 Sterne "eher nicht zutreffend", 2 Sterne "nicht zutreffend", 1 Stern "weiß ich nicht"] *

Abbildung 47: Teilauszug der Forms-Umfrage (eigene Abbildung)

Die Auswertung der Antworten erfolgt mit Microsoft Excel (Office 365), das direkt eine Schnittstelle zu Microsoft Forms bietet. Im Durchschnitt benötigen die Teilnehmenden 30 Minuten zur Beantwortung der Fragen aller zehn Projekte. Es werden folgende zehn Feststellungen zur Beantwortung gemäß Tabelle 3 vorgeschlagen:

Tabelle 3: Fragen der Evaluation

1	Die Kombination von Design-Entwurf mit TinkerCAD, praktischem Aufbau am Steckbrett und die visuelle und textuelle Programmierung erleichtert den Einstieg bzw. das Verständnis für das Thema Computertechnik (Digital- und Analogtechnik).
---	---

2	Die visuelle Programmierung mit MakeCode erleichtert den Einstieg in die Programmierung.
3	Die visuelle Programmierung mit MakeCode erleichtert das Verständnis für die textuelle Programmierung mit MicroPython.
4	Die Kombination aus Unterricht und angeleitetem Nachbau und Erklärung der einzelnen Projekte ist eine Hilfe für das Verständnis der Thematik "Computertechnik".
5	Die fertig programmierten und dokumentierten Projektbeispiele als gespeicherte Sicherung (Backup) stellen eine Erleichterung für den eigenen Unterricht dar.
6	Der inhaltliche Zusammenhang von „Technischer Informatik“ mit der Kombination von Digital- bzw. Analogtechnik und Programmierung ist aus den Projekten erkennbar.
7	Das Übungsbeispiel X gibt Gelegenheit für Adaptierungen und/oder neue Ideen.
*Antwortmöglichkeiten: 5= sehr zutreffend, 4= zutreffend, 3= eher nicht zutreffend, 2= nicht zutreffend, 1= weiß nicht	

Die Fragen orientieren sich am strukturellen Aufbau der beiden Schulungen. Inhaltlich steht bei Frage 1 die Betrachtung des Kombinars (Planung/Design mit Planungssoftware, praktische Umsetzung mit elektronischen Bauteilen, Programmierung der Funktionen) im Zentrum. Für die Fragen 2 bis 3 wird der Umgang mit visueller und textueller Programmierung erhoben. Die Fragen 4 und 5 gewichten die Handhabung des Schulungsmaterials bzw. der ausprogrammierten Lösungen. Frage 6 nimmt die notwendige Kombination aus Technischer Informatik, elektronischer Analogtechnik sowie deren Programmierung in den Fokus. Frage 7 betont die individuellen Ressourcen der Teilnehmer*innen am Training. Es wird erwartet, dass alle Schulungsteilnehmenden den Einstieg in die Technische Informatik und Digitaltechnik, mit der sich wiederholenden Schulungsdidaktik samt Projektbeschreibung, erfolgreich absolvieren. Zudem wird erwartet,

dass die Schulungsteilnehmenden geringe Adaptierungen der Schulungsbeispiele eigenständig vornehmen können. Die Erwartungen werden in den sieben Thesen der Tabelle 4 formuliert:

Tabelle 4: Formulierung von sieben Thesen zur Beantwortung der Forschungsfrage (eigene Tabelle)

These 1	Die Kombination von Design-Entwurf mit TinkerCAD, praktischem Aufbau am Steckbrett und die visuelle und textuelle Programmierung wird von mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 als erleichternd wahrgenommen.
These 2	Die visuelle Programmierung mit MakeCode als Einstieg in die Programmierung wird von mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 als erleichternd wahrgenommen.
These 3	Die visuelle Programmierung mit MakeCode erleichtert für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 das Verständnis für die textuelle Programmierung mit MicroPython.
These 4	Die Kombination aus Unterricht, angeleitetem Nachbau und Erklärung der einzelnen Projekte ist für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 eine Hilfe für das Verständnis der Thematik "Computertechnik".
These 5	Die fertig programmierten und dokumentierten Projektbeispiele als gespeicherte Sicherung (Backup) stellen für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 eine Erleichterung für den eigenen Unterricht dar.
These 6	Der Gesamtzusammenhang von Technischer Informatik mit der Kombination von Digital- und Analogtechnik sowie Programmierung ist für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 aus den Projekten erkennbar.
These 7	Das Übungsbeispiel X gibt mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 Gelegenheit für Adaptierungen und neue Ideen.

8.2 Ergebnisse

In einem ersten Schritt werden die Häufigkeiten zur Antwort „*weiß nicht*“ in Gruppe 1 und 2 dargestellt. Die Antwort „*weiß nicht*“ streut über alle 10 Projekte. So wird zum Beispiel Frage 1 über alle 10 Projekte insgesamt ein Mal (Gruppe 1) bzw. zwei Mal (Gruppe 2) mit „*weiß nicht*“ beantwortet. Oder so wird z.B. Frage 5 über alle 10 Projekte insgesamt fünf Mal (Gruppe 1) bzw. fünf Mal (Gruppe 2) mit „*weiß nicht*“ beantwortet. Die Häufigkeiten schwanken nur wenig. Die folgende Abbildung 48 zeigt die Streuung von „*weiß nicht*“:

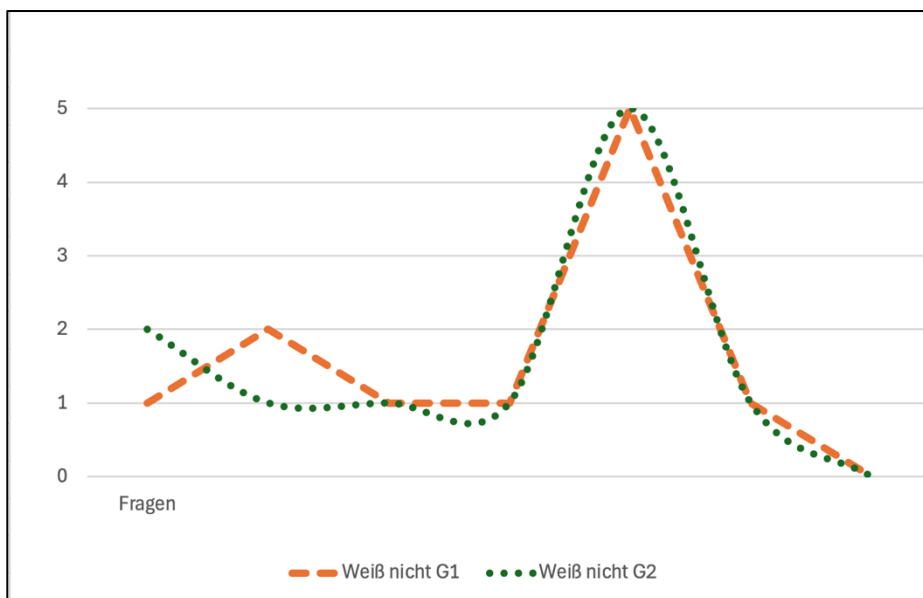


Abbildung 48: Anzahl "weiß nicht" über alle 10 Projekte

Total wird die Antwort „*weiß nicht*“ über alle 10 Projekte 22-mal gesamt aus Gruppe 1 und Gruppe 2 angegeben.

Die Streuung zwischen 0 und 5 Antworten kann pro Projekt als normal interpretiert werden. Dass bei Frage 5 (fertig programmierte/dokumentierte Projektbeispiele, Erleichterung) in beiden Gruppen je fünf Antworten mit „*weiß nicht*“ angegeben sind, ist überraschend. Denn es ist nicht nachvollziehbar, aus welchen Überlegungen die fertig dokumentierten Projekte samt Erklärung keine Erleichterung darstellen sollten. Womöglich fehlen erstens den Antwortenden die beruflichen Vorkenntnisse und Erfahrungen, um

Frage 5 beantworten zu können. Oder es ist zweitens die Frage 5 zu wenig klar formuliert, da aus der Frageformulierung möglicherweise nicht deutlich hervorgeht, auf welche Unterlagen genau Bezug genommen wird.

Da der Wert 1 („weiß nicht“) inhaltlich nicht auswertbar ist, wird er für die Bewertung der Fragen nicht herangezogen. Nach Bereinigung der Daten um den Wert 1 („weiß nicht“) zeigen sich für die Fragen 1 bis 7 über alle 10 Projekte in den Gruppen 1 und 2 folgende Ergebnisse:

Frage 1

Die Antworten zu Frage 1 („Kombination von Design-Entwurf, praktischem Aufbau am Steckbrett und visueller und textueller Programmierung als Erleichterung beim Einstieg“) ergeben bei den Gruppen 1 und 2 über alle zehn Projekte folgendes Ergebnis:

Von 77 Antworten in Gruppe 1 sowie 89 Antworten in Gruppe 2 werden 139 Antworten mit dem Wert 5 („sehr zutreffend“) sowie 21 Antworten mit dem Wert 4 („zutreffend“) gegeben. 4 Antworten erfolgen mit dem Wert 3 („eher nicht zutreffend“) und dem Wert 2 („nicht zutreffend“).

Dies bedeutet, dass 12,65 % der Schulungsteilnehmenden die Frage 1 über alle 10 Projekte (total 166 Antworten) mit „zutreffend“ sowie 83,73 % mit „sehr zutreffend“ bewerten. Weiters bewerten 1,20 % der Schulungsteilnehmenden die Frage 1 als „nicht zutreffend“ und 2,41 % der Schulungsteilnehmenden als „eher nicht zutreffend“. Die folgende Abbildung 49 zeigt das Ergebnis der Häufigkeitszählung zu Frage 1 im Überblick:

Frage 1 (Gruppe 1 + Gruppe 2)																						
	G1P1	G1P2	G1P3	G1P4	G1P5	G1P6	G1P7	G1P8	G1P10	G1P11	G2P1	G2P2	G2P3	G2P4	G2P5	G2P6	G2P7	G2P8	G2P10	G2P11	Σ	
Anzahl "2"	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	1,00	0,00	2,00	
Anzahl "3"	0,00	1,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	4,00	
Anzahl "4"	4,00	0,00	1,00	1,00	0,00	0,00	0,00	1,00	1,00	1,00	2,00	1,00	1,00	4,00	0,00	1,00	0,00	1,00	1,00	1,00	21,00	
Anzahl "5"	13,00	11,00	8,00	5,00	5,00	4,00	5,00	4,00	4,00	6,00	10,00	7,00	5,00	8,00	7,00	7,00	8,00	7,00	7,00	8,00	139,00	
Σ Antworten	17,00	12,00	9,00	7,00	5,00	4,00	5,00	5,00	5,00	8,00	12,00	8,00	6,00	12,00	7,00	8,00	8,00	9,00	10,00	9,00	166,00	
Antworten G1	77,00																					
Antworten G2	89,00																					

Abbildung 49: Häufigkeiten Frage 1 (G1 und G2)

Aufgrund der Häufigkeitsauszählung ergibt sich für Frage 1 die folgende Häufigkeitstabelle mit absoluten und relativen Häufigkeiten als Tabelle 5:

Tabelle 5: Häufigkeitstabelle für Frage 1

	Häufigkeit	
Bewertung	absolut	relativ
"2" - nicht zutreffend	2,00	1,20%
"3" - eher nicht zutreffend	4,00	2,41%
"4" - zutreffend	21,00	12,65%
"5" - sehr zutreffend	139,00	83,73%
Σ	166,00	100,00%

Die visuelle Darstellung der absoluten Häufigkeiten erfolgt gemäß folgender Abbildung 50:

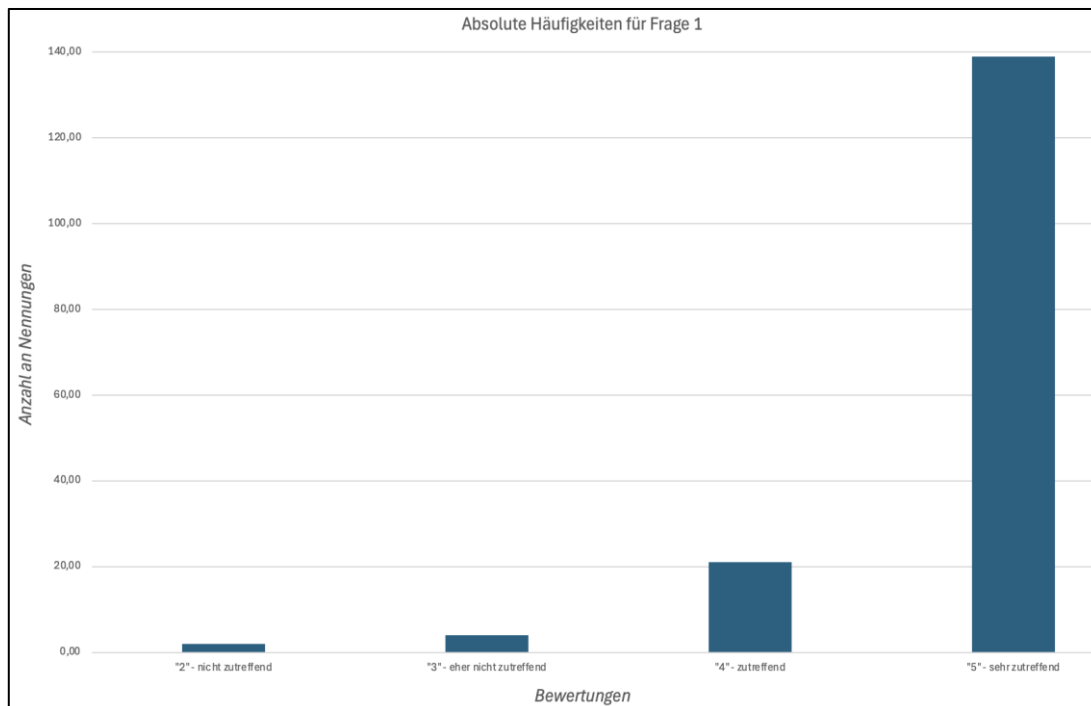


Abbildung 50: Grafische Darstellung der absoluten Häufigkeiten für Frage 1 (eigene Abbildung)

Mit rund 96 % Zustimmung der Schulungsteilnehmenden wird also die „Kombination

von Design-Entwurf, praktischem Aufbau am Steckbrett und visueller und textueller Programmierung“ durch die Projekte 1 bis 10 als Erleichterung beim Einstieg wahrgenommen. Die Zustimmung liegt damit weit über 50 %. These 1 kann damit angenommen werden. Da in Schulungen nie alle Teilnehmenden zu 100 % erreicht werden können, sind 4 % Ablehnungen ohne weiteres erklärbar.

Frage 2

Die Antworten zu Frage 2 („Visuelle Programmierung mit MakeCode als Erleichterung für den Programmier-Einstieg“) ergeben bei den Gruppen 1 und 2 über alle zehn Projekte folgendes Ergebnis:

Von 76 Antworten in Gruppe 1 sowie 90 Antworten in Gruppe 2 werden 132 Antworten mit dem Wert 5 („sehr zutreffend“) sowie 23 Antworten mit dem Wert 4 („zutreffend“) gegeben. 11 Antworten erfolgen mit dem Wert 3 („eher nicht zutreffend“). Keine Antwort ergibt den Wert 2 („nicht zutreffend“).

Dies bedeutet, dass 13,86 % der Schulungsteilnehmenden die Frage 2 über alle 10 Projekte (total 166 Antworten) mit „zutreffend“ sowie 79,52% mit „sehr zutreffend“ bewerten. Weiters bewerten 0,00 % der Schulungsteilnehmenden die Frage 2 als „nicht zutreffend“ und 6,63 % der Schulungsteilnehmenden als „eher nicht zutreffend“. Die folgende Abbildung 51 zeigt das Ergebnis der Häufigkeitszählung zur Frage 2 im Überblick:

Frage 2 (Gruppe 1 + Gruppe 2)																					
	G1P1	G1P2	G1P3	G1P4	G1P5	G1P6	G1P7	G1P8	G1P10	G1P11	G2P1	G2P2	G2P3	G2P4	G2P5	G2P6	G2P7	G2P8	G2P10	G2P11	Σ
Anzahl "2"	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Anzahl "3"	1,00	0,00	1,00	1,00	0,00	1,00	0,00	0,00	0,00	3,00	1,00	0,00	0,00	0,00	0,00	2,00	0,00	0,00	1,00	0,00	11,00
Anzahl "4"	2,00	1,00	0,00	1,00	0,00	0,00	0,00	1,00	1,00	0,00	2,00	1,00	1,00	5,00	0,00	2,00	1,00	2,00	0,00	3,00	23,00
Anzahl "5"	13,00	11,00	8,00	5,00	5,00	3,00	5,00	4,00	4,00	5,00	9,00	7,00	5,00	7,00	7,00	4,00	7,00	8,00	9,00	6,00	132,00
Σ Antworten	16,00	12,00	9,00	7,00	5,00	4,00	5,00	5,00	5,00	8,00	12,00	8,00	6,00	12,00	7,00	8,00	8,00	10,00	10,00	9,00	166,00
Antworten G1	76,00																				
Antworten G2	90,00																				

Abbildung 51: Häufigkeiten Frage 2 (G1 und G2)

Aufgrund der Häufigkeitsauszählung ergibt sich für Frage 2 die folgende Häufigkeitstabelle mit absoluten und relativen Häufigkeiten als Tabelle 6:

Tabelle 6: Häufigkeitstabelle für Frage 2

Bewertung	Häufigkeit	
	absolut	relativ
"2" - nicht zutreffend	0,00	0,00%
"3" - eher nicht zutreffend	11,00	6,63%
"4" - zutreffend	23,00	13,86%
"5" - sehr zutreffend	132,00	79,52%
Σ	166,00	100,00%

Die visuelle Darstellung der absoluten Häufigkeiten erfolgt gemäß folgender Abbildung 52:

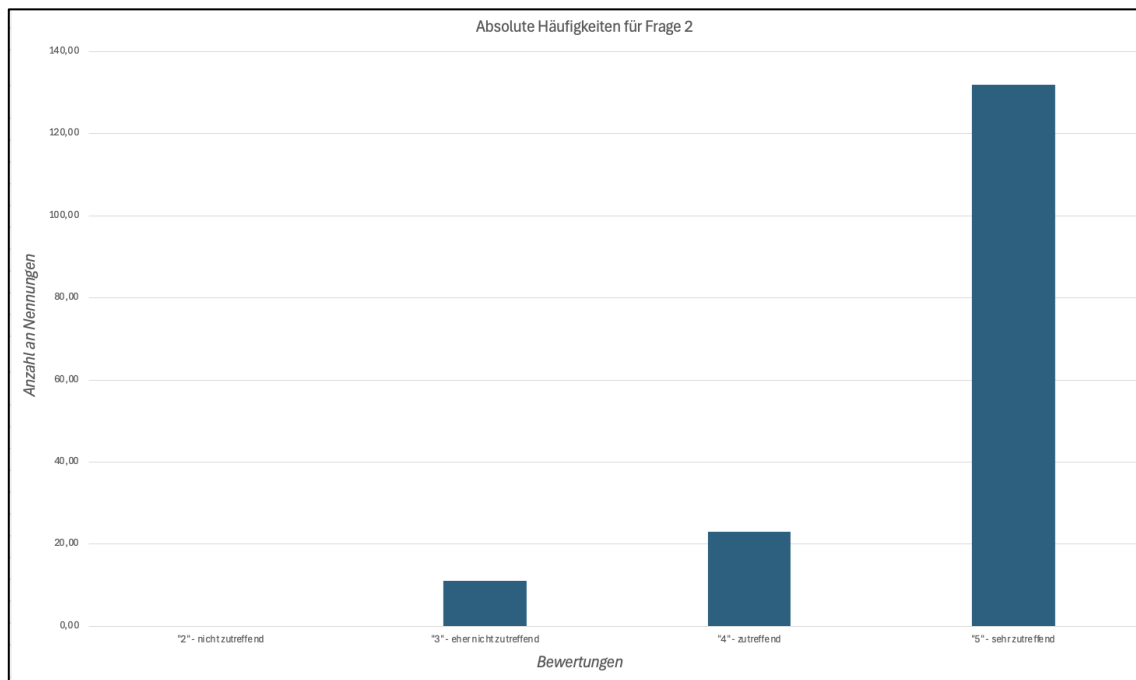


Abbildung 52: Grafische Darstellung der absoluten Häufigkeiten für Frage 2 (eigene Abbildung)

Mit rund 93 % Zustimmung der Schulungsteilnehmenden wird also die „visuelle Programmierung mit MakeCode“ durch die Projekte 1 bis 10 als Erleichterung beim ProgrammierEinstieg wahrgenommen. Die Zustimmung liegt damit weit über 50 %. These 2 kann damit angenommen werden. Da in Schulungen nie alle Teilnehmenden zu 100 %

erreicht werden können, sind die 7 % Ablehnungen ohne weiteres erklärbar.

Frage 3

Die Antworten zu Frage 3 („Visuelle Programmierung mit MakeCode als Erleichterung für den textuellen ProgrammierEinstieg mit MicroPython“) ergeben bei den Gruppen 1 und 2 über alle zehn Projekte folgendes Ergebnis:

Von 77 Antworten in Gruppe 1 sowie 88 Antworten in Gruppe 2 werden 118 Antworten mit dem Wert 5 („sehr zutreffend“) sowie 33 Antworten mit dem Wert 4 („zutreffend“) gegeben. 13 Antworten erfolgen mit dem Wert 3 („eher nicht zutreffend“). Eine Antwort ergibt den Wert 2 („nicht zutreffend“).

Dies bedeutet, dass 20,00 % der Schulungsteilnehmenden die Frage 3 über alle zehn Projekte (total 166 Antworten) mit „zutreffend“ sowie 71,52 % mit „sehr zutreffend“ bewerten. Weiters bewerten 0,61 % der Schulungsteilnehmenden die Frage 3 als „nicht zutreffend“ und 7,88 % der Schulungsteilnehmenden als „eher nicht zutreffend“. Die folgende Abbildung 53 zeigt das Ergebnis der Häufigkeitszählung zur Frage 3 im Überblick:

Frage 3 (Gruppe 1 + Gruppe 2)																						
	G1P1	G1P2	G1P3	G1P4	G1P5	G1P6	G1P7	G1P8	G1P10	G1P11	G2P1	G2P2	G2P3	G2P4	G2P5	G2P6	G2P7	G2P8	G2P10	G2P11	Σ	
Anzahl "2"	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	
Anzahl "3"	3,00	1,00	1,00	0,00	0,00	1,00	0,00	0,00	0,00	2,00	1,00	0,00	0,00	2,00	0,00	0,00	0,00	1,00	1,00	0,00	13,00	
Anzahl "4"	2,00	1,00	0,00	2,00	1,00	0,00	0,00	1,00	1,00	0,00	4,00	3,00	1,00	5,00	0,00	3,00	1,00	3,00	2,00	3,00	33,00	
Anzahl "5"	12,00	10,00	8,00	4,00	4,00	3,00	5,00	4,00	4,00	6,00	6,00	5,00	5,00	5,00	7,00	4,00	7,00	6,00	7,00	6,00	118,00	
Σ Antworten	17,00	12,00	9,00	7,00	5,00	4,00	5,00	5,00	5,00	8,00	11,00	8,00	6,00	12,00	7,00	7,00	8,00	10,00	10,00	9,00	165,00	
Antworten G1	77,00																					
Antworten G2	88,00																					

Abbildung 53: Häufigkeiten Frage 3 (G1 und G2)

Aufgrund der Häufigkeitsauszählung ergibt sich für Frage 3 die folgende Häufigkeitstabelle mit absoluten und relativen Häufigkeiten als Tabelle 7:

Tabelle 7: Häufigkeitstabelle für Frage 3

Bewertung	Häufigkeit	
	absolut	relativ
"2" - nicht zutreffend	1,00	0,61%
"3" - eher nicht zutreffend	13,00	7,88%
"4" - zutreffend	33,00	20,00%
"5" - sehr zutreffend	118,00	71,52%
Σ	165,00	100,00%

Die visuelle Darstellung der absoluten Häufigkeiten erfolgt gemäß folgender Abbildung 54:

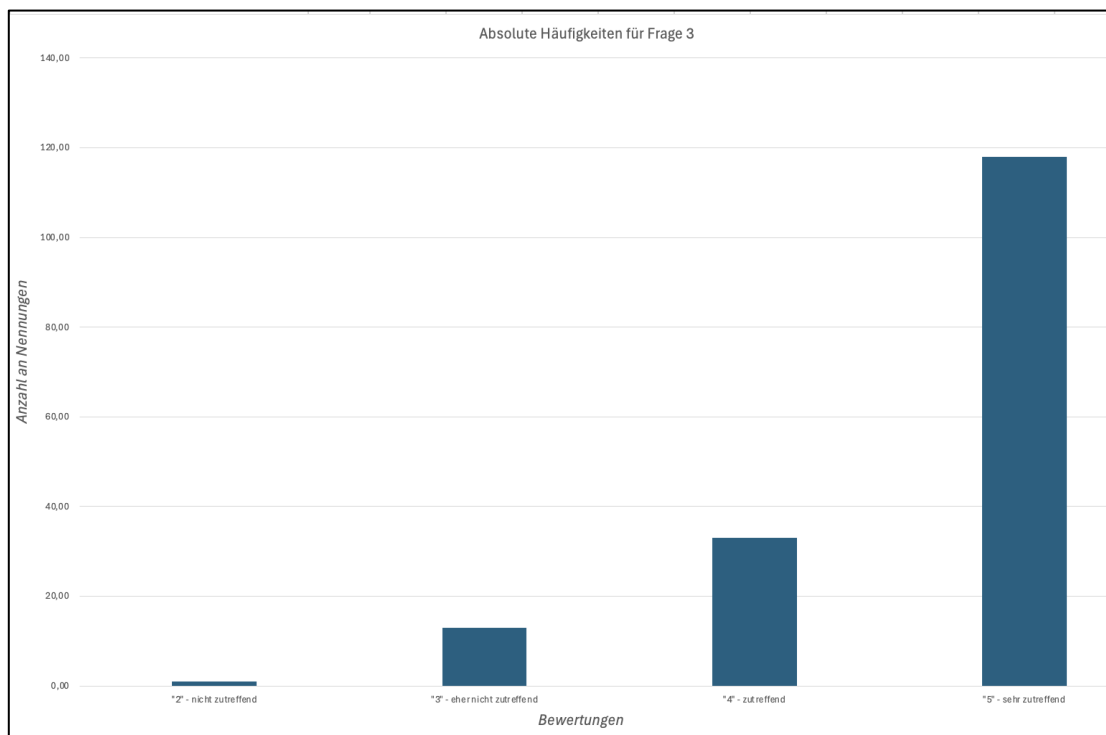


Abbildung 54: Grafische Darstellung der absoluten Häufigkeiten für Frage 3 (eigene Abbildung)

Mit rund 92 % Zustimmung der Schulungsteilnehmenden wird also die „visuelle Programmierung mit MakeCode“ durch die Projekte 1 bis 10 als Erleichterung beim textu-

ellen ProgrammierEinstieg mit MicroPython wahrgenommen. Die Zustimmung liegt damit weit über 50 %. These 3 kann damit angenommen werden. Da in Schulungen nie alle Teilnehmenden zu 100 % erreicht werden können, sind rund 8 % Ablehnungen ohne weiteres erklärbar.

Frage 4

Die Antworten zu Frage 4 („Kombination aus Unterricht und angeleitetem Nachbau und Erklärung der einzelnen Projekte als Hilfe für das Verständnis der Thematik ,Computertechnik“) ergeben bei den Gruppen 1 und 2 über alle zehn Projekte folgendes Ergebnis:

Von 77 Antworten in Gruppe 1 sowie 90 Antworten in Gruppe 2 werden 135 Antworten mit dem Wert 5 („sehr zutreffend“) sowie 23 Antworten mit dem Wert 4 („zutreffend“) gegeben. 9 Antworten erfolgen mit dem Wert 3 („eher nicht zutreffend“). Keine Antwort ergibt den Wert 2 („nicht zutreffend“).

Dies bedeutet, dass 13,77 % der Schulungsteilnehmenden die Frage 4 über alle 10 Projekte (total 167 Antworten) mit „zutreffend“ sowie 80,84 % mit „sehr zutreffend“ bewerten. Weiters bewerten 0,00 % der Schulungsteilnehmenden die Frage 4 als „nicht zutreffend“ und 5,39 % der Schulungsteilnehmenden als „eher nicht zutreffend“. Die folgende Abbildung 55 zeigt das Ergebnis der Häufigkeitszählung zur Frage 4 im Überblick:

Frage 4 (Gruppe 1 + Gruppe 2)																					
	G1P1	G1P2	G1P3	G1P4	G1P5	G1P6	G1P7	G1P8	G1P10	G1P11	G2P1	G2P2	G2P3	G2P4	G2P5	G2P6	G2P7	G2P8	G2P10	G2P11	Σ
Anzahl "2"	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Anzahl "3"	0,00	0,00	0,00	2,00	0,00	1,00	0,00	0,00	0,00	2,00	1,00	0,00	0,00	2,00	0,00	0,00	0,00	0,00	1,00	0,00	9,00
Anzahl "4"	2,00	2,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00	0,00	2,00	1,00	1,00	2,00	0,00	3,00	1,00	3,00	0,00	2,00	23,00
Anzahl "5"	15,00	10,00	8,00	5,00	4,00	3,00	5,00	4,00	4,00	6,00	9,00	7,00	5,00	8,00	7,00	5,00	7,00	7,00	9,00	7,00	135,00
Σ Antworten	17,00	12,00	9,00	7,00	5,00	4,00	5,00	5,00	5,00	8,00	12,00	8,00	6,00	12,00	7,00	8,00	8,00	10,00	10,00	9,00	167,00
Antworten G1	77,00																				
Antworten G2	90,00																				

Abbildung 55: Häufigkeiten Frage 4 (G1 und G2)

Aufgrund der Häufigkeitsauszählung ergibt sich für Frage 4 die folgende Häufigkeitstabelle mit absoluten und relativen Häufigkeiten als Tabelle 8:

Tabelle 8: Häufigkeitstabelle für Frage 4

	Häufigkeit	
Bewertung	absolut	relativ
"2" - nicht zutreffend	0,00	0,00%
"3" - eher nicht zutreffend	9,00	5,39%
"4" - zutreffend	23,00	13,77%
"5" - sehr zutreffend	135,00	80,84%
Σ	167,00	100,00%

Die visuelle Darstellung der absoluten Häufigkeiten erfolgt gemäß folgender Abbildung 56:

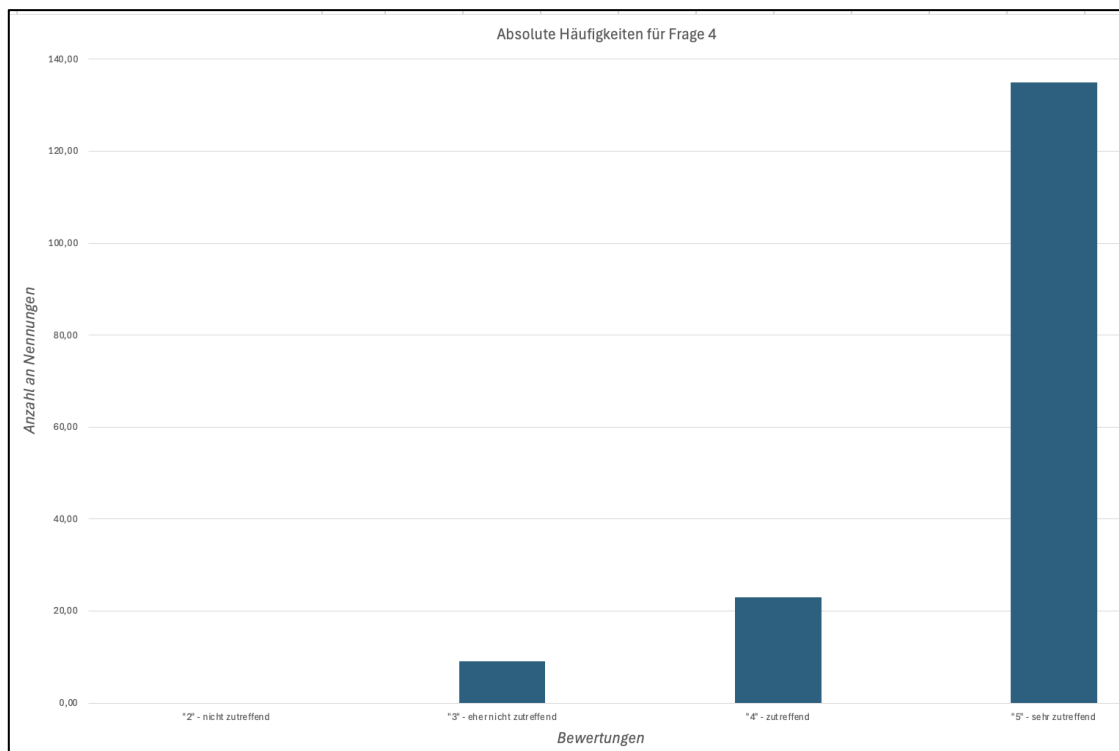


Abbildung 56: Grafische Darstellung der absoluten Häufigkeiten für Frage 4 (eigene Abbildung)

Mit rund 94 % Zustimmung der Schulungsteilnehmenden wird also die „Kombination aus Unterricht und angeleitetem Nachbau und Erklärung der einzelnen Projekte als Hilfe für das Verständnis der Thematik „Computertechnik“ durch die Projekte 1 bis 10 als Erleichterung beim textuellen ProgrammierEinstieg mit MicroPython wahrgenommen. Die

Zustimmung liegt damit weit über 50 %. These 4 kann damit angenommen werden. Da in Schulungen nie alle Teilnehmenden zu 100 % erreicht werden können, sind rund 5 % Ablehnungen ohne weiteres erklärbar.

Frage 5

Die Antworten zu Frage 5 („Die fertig programmierten und dokumentierten Projektbeispiele als gespeicherte Sicherung (Backup) als Erleichterung für den eigenen Unterricht“) ergeben bei den Gruppen 1 + 2 über alle 10 Projekte folgendes Ergebnis:

Von 73 Antworten in Gruppe 1 sowie 87 Antworten in Gruppe 2 werden 134 Antworten mit dem Wert 5 („sehr zutreffend“) sowie 16 Antworten mit dem Wert 4 („zutreffend“) gegeben. 9 Antworten erfolgen mit dem Wert 3 („eher nicht zutreffend“). Eine Antwort ergibt den Wert 2 („nicht zutreffend“).

Dies bedeutet, dass 10,00 % der Schulungsteilnehmenden die Frage 5 über alle 10 Projekte (total 160 Antworten) mit „zutreffend“ sowie 83,75% mit „sehr zutreffend“ bewerten. Weiters bewerten 0,63 % der Schulungsteilnehmenden die Frage 5 als „nicht zutreffend“ und 5,39 % der Schulungsteilnehmenden als „eher nicht zutreffend“. Die folgende Abbildung 57 zeigt das Ergebnis der Häufigkeitszählung zur Frage 5 im Überblick:

Frage 5 (Gruppe 1 + Gruppe 2)																					
	G1P1	G1P2	G1P3	G1P4	G1P5	G1P6	G1P7	G1P8	G1P10	G1P11	G2P1	G2P2	G2P3	G2P4	G2P5	G2P6	G2P7	G2P8	G2P10	G2P11	Σ
Anzahl "2"	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00
Anzahl "3"	0,00	1,00	1,00	1,00	0,00	1,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	2,00	0,00	0,00	0,00	1,00	1,00	0,00	9,00
Anzahl "4"	1,00	1,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	3,00	1,00	1,00	2,00	1,00	2,00	1,00	0,00	0,00	2,00	16,00
Anzahl "5"	14,00	10,00	8,00	5,00	5,00	3,00	5,00	4,00	4,00	6,00	9,00	7,00	5,00	7,00	6,00	6,00	7,00	8,00	9,00	6,00	134,00
Σ Antworten	15,00	12,00	9,00	7,00	5,00	4,00	5,00	4,00	4,00	8,00	12,00	8,00	6,00	11,00	7,00	8,00	8,00	9,00	10,00	8,00	160,00
Antworten G1	73,00																				
Antworten G2	87,00																				

Abbildung 57: Häufigkeiten Frage 5 (G1 und G2)

Aufgrund der Häufigkeitsauszählung ergibt sich für Frage 5 die folgende Häufigkeitstabelle mit absoluten und relativen Häufigkeiten als Tabelle 9:

Tabelle 9: Häufigkeitstabelle für Frage 5

Bewertung	Häufigkeit	
	absolut	relativ
"2" - nicht zutreffend	1,00	0,63%
"3" - eher nicht zutreffend	9,00	5,63%
"4" - zutreffend	16,00	10,00%
"5" - sehr zutreffend	134,00	83,75%
Σ	160,00	100,00%

Die visuelle Darstellung der absoluten Häufigkeiten erfolgt gemäß folgender Abbildung 58:

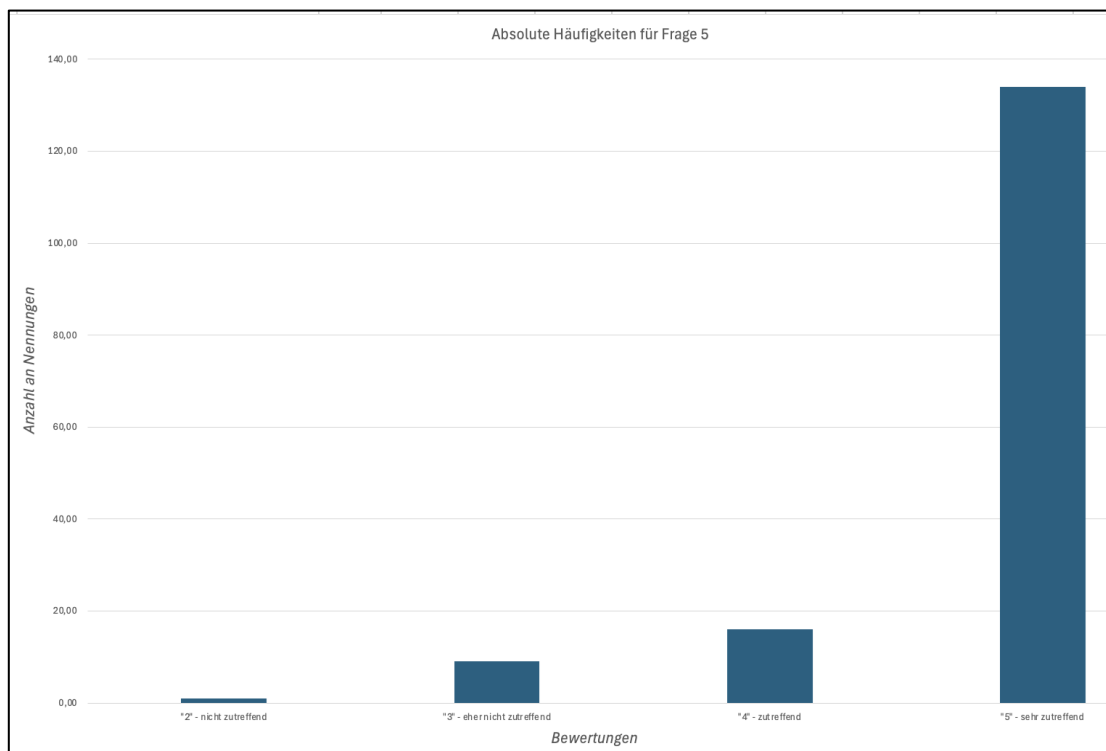


Abbildung 58: Grafische Darstellung der absoluten Häufigkeiten für Frage 5 (eigene Abbildung)

Mit rund 94 % Zustimmung der Schulungsteilnehmenden werden also die „die fertig programmierten und dokumentierten Projektbeispiele als gespeicherte Sicherung (Backup)“

durch die Projekte 1 bis 10 als Erleichterung für den eigenen Unterricht wahrgenommen. Die Zustimmung liegt damit weit über 50 %. These 5 kann damit angenommen werden. Da in Schulungen nie alle Teilnehmenden zu 100 % erreicht werden können, sind die rund 6 % Ablehnungen ohne weiteres erklärbar

Frage 6

Die Antworten zu Frage 6 („Der inhaltliche Zusammenhang von ‚Technischer Informatik‘ mit der Kombination von Digital- bzw. Analogtechnik und Programmierung ist aus den Projekten erkennbar“) ergeben bei den Gruppen 1 und 2 über alle zehn Projekte folgendes Ergebnis:

Von 77 Antworten in Gruppe 1 sowie 88 Antworten in Gruppe 2 werden 127 Antworten mit dem Wert 5 („sehr zutreffend“) sowie 26 Antworten mit dem Wert 4 („zutreffend“) gegeben. 10 Antworten erfolgen mit dem Wert 3 („eher nicht zutreffend“). 2 Antworten ergibt den Wert 2 („nicht zutreffend“).

Dies bedeutet, dass 15,76 % der Schulungsteilnehmenden die Frage 6 über alle 10 Projekte (total 165 Antworten) mit „zutreffend“ sowie 76,97 % mit „sehr zutreffend“ bewerten. Weiters bewerten 1,21 % der Schulungsteilnehmenden die Frage 6 als „nicht zutreffend“ und 6,06 % der Schulungsteilnehmenden als „eher nicht zutreffend“. Die folgende Abbildung 59 zeigt das Ergebnis der Häufigkeitszählung zur Frage 4 im Überblick:

Frage 6 (Gruppe 1 + Gruppe 2)																					
	G1P1	G1P2	G1P3	G1P4	G1P5	G1P6	G1P7	G1P8	G1P10	G1P11	G2P1	G2P2	G2P3	G2P4	G2P5	G2P6	G2P7	G2P8	G2P10	G2P11	Σ
Anzahl "2"	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	2,00
Anzahl "3"	0,00	0,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00	0,00	0,00	0,00	2,00	0,00	0,00	0,00	2,00	1,00	0,00	10,00
Anzahl "4"	4,00	2,00	0,00	1,00	0,00	0,00	0,00	1,00	0,00	0,00	5,00	2,00	2,00	1,00	0,00	2,00	1,00	3,00	0,00	2,00	26,00
Anzahl "5"	13,00	10,00	8,00	5,00	5,00	3,00	5,00	4,00	4,00	6,00	7,00	6,00	4,00	8,00	7,00	6,00	7,00	4,00	8,00	7,00	127,00
Σ Antworten	17,00	12,00	9,00	7,00	5,00	4,00	5,00	5,00	5,00	8,00	12,00	8,00	6,00	11,00	7,00	8,00	8,00	9,00	10,00	9,00	165,00
Antworten G1	77,00																				
Antworten G2	88,00																				

Abbildung 59: Häufigkeiten Frage 6 (G1 und G2)

Aufgrund der Häufigkeitsauszählung ergibt sich für Frage 6 die folgende Häufigkeitstabelle mit absoluten und relativen Häufigkeiten als Tabelle 10:

Tabelle 10: Häufigkeitstabelle für Frage 6

	Häufigkeit	
Bewertung	absolut	relativ
"2" - nicht zutreffend	2,00	1,21%
"3" - eher nicht zutreffend	10,00	6,06%
"4" - zutreffend	26,00	15,76%
"5" - sehr zutreffend	127,00	76,97%
Σ	165,00	100,00%

Die visuelle Darstellung der absoluten Häufigkeiten erfolgt gemäß folgender Abbildung 60:

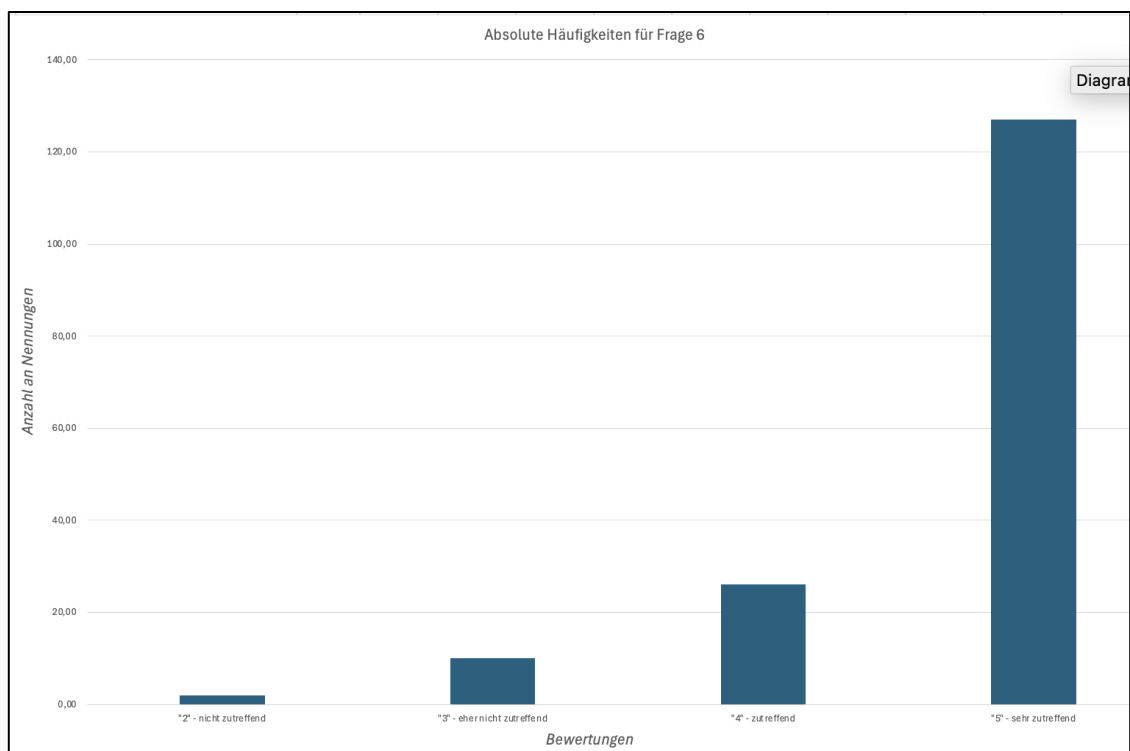


Abbildung 60: Grafische Darstellung der absoluten Häufigkeiten für Frage 6 (eigene Abbildung)

Mit rund 93 % Zustimmung der Schulungsteilnehmenden wird also „der inhaltliche Zu-

sammenhang von ‚Technischer Informatik‘ mit der Kombination von Digital- bzw. Analogtechnik und Programmierung ist aus den Projekten“ durch die Projekte 1 bis 10 von den Teilnehmenden erkannt. Die Zustimmung liegt damit weit über 50 %. These 6 kann damit angenommen werden. Da in Schulungen nie alle Teilnehmenden zu 100 % erreicht werden können, sind rund 7 % Ablehnungen ohne weiteres erklärbar.

Frage 7

Die Antworten zu Frage 7 („Das Übungsbeispiel X gibt Gelegenheit für Adaptierungen und/oder neue Ideen“) ergeben bei den Gruppen 1 und 2 über alle zehn Projekte folgendes Ergebnis:

Von 78 Antworten in Gruppe 1 sowie 87 Antworten in Gruppe 2 werden 114 Antworten mit dem Wert 5 („sehr zutreffend“) sowie 30 Antworten mit dem Wert 4 („zutreffend“) gegeben. 17 Antworten erfolgen mit dem Wert 3 („eher nicht zutreffend“). 4 Antworten ergeben den Wert 2 („nicht zutreffend“).

Dies bedeutet, dass 18,18 % der Schulungsteilnehmenden die Frage 7 über alle 10 Projekte (total 165 Antworten) mit „zutreffend“ sowie 69,09 % mit „sehr zutreffend“ bewerten. Weiters bewerten 2,42 % der Schulungsteilnehmenden die Frage 7 als „nicht zutreffend“ und 10,30 % der Schulungsteilnehmenden als „eher nicht zutreffend“. Die folgende Abbildung 61 zeigt das Ergebnis der Häufigkeitszählung zur Frage 7 im Überblick:

Frage 7 (Gruppe 1 + Gruppe 2)																					
	G1P1	G1P2	G1P3	G1P4	G1P5	G1P6	G1P7	G1P8	G1P10	G1P11	G2P1	G2P2	G2P3	G2P4	G2P5	G2P6	G2P7	G2P8	G2P10	G2P11	Σ
Anzahl "2"	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	2,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	1,00	4,00
Anzahl "3"	1,00	0,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00	3,00	1,00	0,00	2,00	0,00	0,00	1,00	1,00	2,00	1,00	17,00
Anzahl "4"	3,00	2,00	0,00	2,00	1,00	0,00	0,00	1,00	0,00	0,00	4,00	2,00	1,00	2,00	2,00	2,00	1,00	3,00	1,00	3,00	30,00
Anzahl "5"	13,00	10,00	8,00	4,00	4,00	3,00	5,00	4,00	4,00	6,00	5,00	5,00	5,00	6,00	5,00	5,00	6,00	5,00	7,00	4,00	114,00
Σ Antworten	17,00	12,00	9,00	7,00	5,00	4,00	5,00	5,00	5,00	9,00	12,00	8,00	6,00	10,00	7,00	8,00	8,00	9,00	10,00	9,00	165,00
Antworten G1	78,00																				
Antworten G2	87,00																				

Abbildung 61: Häufigkeiten Frage 7 (G1 und G2)

Aufgrund der Häufigkeitsauszählung ergibt sich für Frage 7 die folgende Häufigkeitstabelle mit absoluten und relativen Häufigkeiten als Tabelle 11:

Tabelle 11: Häufigkeitstabelle für Frage 7

	Häufigkeit	
Bewertung	absolut	relativ
"2" - nicht zutreffend	4,00	2,42%
"3" - eher nicht zutreffend	17,00	10,30%
"4" - zutreffend	30,00	18,18%
"5" - sehr zutreffend	114,00	69,09%
Σ	165,00	100,00%

Die visuelle Darstellung der absoluten Häufigkeiten erfolgt mit einem Balkendiagramm (Stabdiagramm) gemäß folgender Abbildung 62:

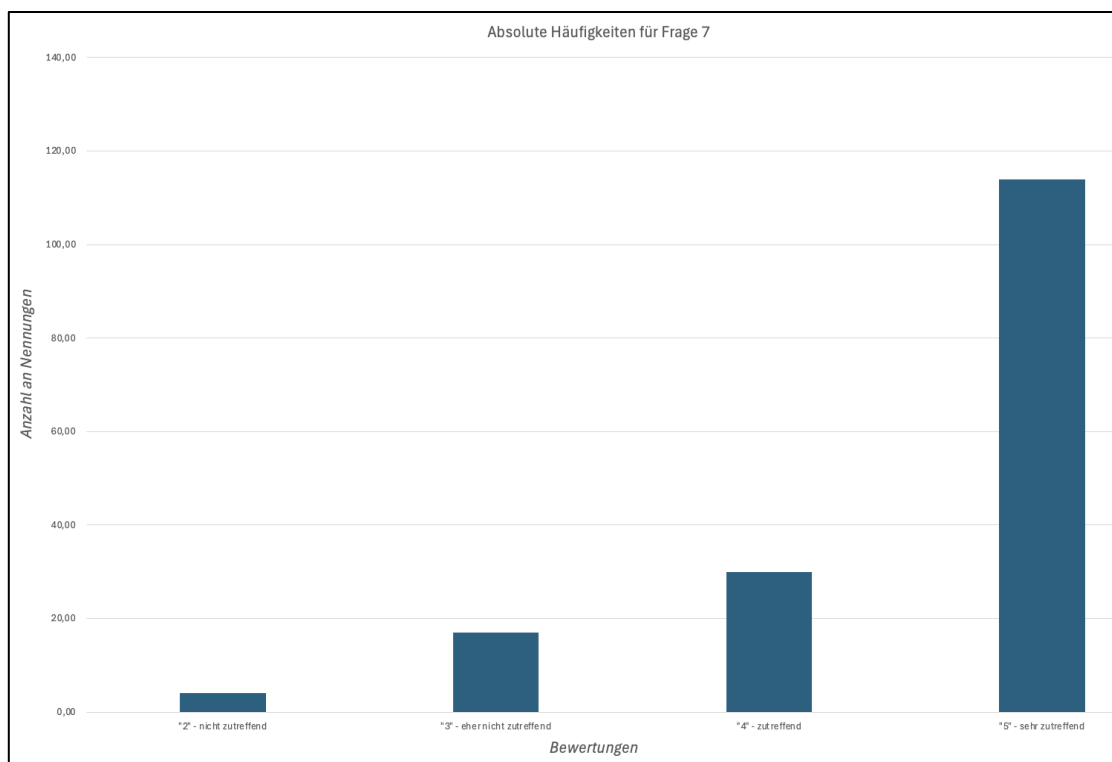


Abbildung 62: Grafische Darstellung der absoluten Häufigkeiten für Frage 7 (eigene Abbildung)

Mit rund 87 % Zustimmung der Schulungsteilnehmenden wird erkannt, dass „das

Übungsbeispiel X Gelegenheit für Adaptierungen und/oder neue Ideen“ durch die Projekte 1 bis 10 gibt. Die Zustimmung liegt damit weit über 50 %. Diese 7 kann damit angenommen werden. Da in Schulungen nie alle Teilnehmenden zu 100 % erreicht werden können, sind die rund 13 % Ablehnungen auch hier noch ohne weiteres erklärbar.

Frage 8

Die Frage 8 (*„Welche eigene Adaptierung(Anpassung des bestehenden Beispiels) fällt Ihnen zum Referenzbeispiel X ein?“*) ist im Gegensatz zu den Fragen 1 bis 7 eine optionale Frage und stellt Adaptierungen des betreffenden Projekts aus Sicht der Teilnehmenden ins Zentrum. Dabei interessieren vor allem Adaptierungen im Design, beim Einsatz der elektronischen Komponenten und schlussendlich bei der Programmierung. Passende Anregungen könnten direkt in das jeweilige Projekt eingebaut werden bzw. in der Projektbeschreibung als Adaptierung erwähnt werden (sofern nicht schon angeführt).

Beim Projekt 1 (Aufbau einer einfachen Reihen- und Parallelschaltung ohne Programmierung) gibt es lediglich zwei Antworten bei G1 und G2, die beide nicht wirklich verwendet werden können. Ein Hinweis urgiert das Abspielen von Musik, was aber für dieses Projekt nicht relevant ist, da keine Programmierung erfolgt. Der zweite Hinweis ist ein Vorschlag, mehrere LEDs und Widerstände zu testen, was wenig Sinn ergibt, da die Durchlassspannung der farbigen LEDs fast ident ist und sich der Vorwiderstandswert in Ohm davon ableitet. Beide Antworten werden damit nicht als verwertbar eingestuft. Aufgrund der Einfachheit der Schaltung und der eingeschränkten Adaptierungsmöglichkeiten erscheint es logisch, dass hier keine weiteren Adaptierungsvorschläge vorhanden sind.

Projekt 2 (*Entwicklung einer programmiertechnischen Schaltung zur Visualisierung einer LED über einen digitalen PIN*) generiert nur zwei Rückmeldungen von G1 und G2. Ein sinnvoller Hinweis ist die Empfehlung, Evaluierungen nicht am Schluss des Workshops, sondern unmittelbar nach dem jeweiligen Projekt zu machen. Der zweite Hinweis ist, die Sperrfunktion einer LED durch absichtliches Falschstecken am Steckbrett zu visualisieren. Der erste Hinweis mit der zeitnahen Evaluierung ist sinnvoll, kann aber zur inhaltlichen Auswertung nicht berücksichtigt werden, da er keine konkrete Projektadaptierung darstellt. Die zweite Antwort ist ebenfalls nachvollziehbar, jedoch wird sie bereits in der

Projektbeschreibung als Anpassung erwähnt. Aufgrund der Einfachheit der Schaltung und der eingeschränkten Adaptierungsmöglichkeiten erscheint es logisch, dass hier keine weiteren Adaptierungsvorschläge vorhanden sind.

Beim Projekt 3 (*programmiertechnische Umsetzung einer Ampelschaltung mit 3 LEDs*) gibt es keine Antworten bei G1 und G2. Aufgrund des doch komplexeren Schaltungsaufbaus und eines Mehr an Programmiercode erscheint es logisch, dass sich die Teilnehmenden mehr auf das Projekt selbst statt auf Adaptierungen fokussierten.

Beim Projekt 4 (*programmiertechnische Umsetzung eines Potentiometers mit einer LED*) gibt es lediglich zwei Antworten bei G1 und G2. Der Vorschlag einer tatsächlichen Visualisierung einer Verkehrsampel mit blinkenden Lichtern wird bereits in der Projektbeschreibung als Adaptierung genannt und wird nicht berücksichtigt. Der zweite Hinweis auf „Vergleich von Analog und Digital“ kann nicht klar interpretiert und gewertet werden. Aufgrund des doch komplexeren Schaltungsaufbaus und eines Mehr an Programmiercode erscheint es logisch, dass sich die Teilnehmenden mehr auf das Projekt selbst statt auf Adaptierungen fokussierten.

Beim Projekt 5 (*Einsatz von Pulsweitenmodulation - PWM*) gibt es lediglich zwei Antworten bei G1 und G2. Hier sind die Adaptierungsvorschläge dahingehend, dass beim Schulungsbeispiel das Heller- und Dunklerwerden der LED nicht kontinuierlich, sondern per Button erfolgen soll. Diese Vorschläge sind sinnvoll und werden bei den kommenden Workshops und in der Projektbeschreibung direkt eingebaut oder als Adaptierungsmöglichkeit angeführt. Aufgrund des doch komplexen Themas PWM und eines Mehr an Programmiercode erscheint es logisch, dass sich die Teilnehmenden mehr auf das Projekt selbst anstatt auf Adaptierungen fokussierten

Beim Projekt 6 (*Binäre Zahlendarstellung durch LEDs*) gibt es lediglich drei Antworten bei G1 und G2. Eine Feststellung ist, dass sich der Programmiercode in MakeCode von MicroPython unterscheidet. Dies stimmt insofern, als eine Eins-zu-eins-Umsetzung hier vor allem bei MakeCode so gut wie unmöglich ist (in MakeCode wurde vorwiegend mit Arrays gearbeitet, um den Code übersichtlich zu halten). Diese Adaptierung kann somit nicht übernommen werden, ist aber berechtigt. Ein weiterer Vorschlag ist die Erweiterung auf mehr LEDs (es wurden aus Komplexitätsgründen nur 3 LEDs verwendet). Dieser Vorschlag ist berechtigt, da in der Digitaltechnik üblicherweise mit 4 Bits (Nibble) oder 8

Bits (Byte) gearbeitet wird. Trotzdem wird das Referenzbeispiel nicht geändert, da vor allem der Mehraufwand am Steckbrett beträchtlich wäre. Als Projekt-Adaptierungs-Idee wird aber der Vorschlag übernommen und im Projekt vermerkt. Der letzte Vorschlag ist eine Adaptierung (Ausbau) zu einem Binäraddierer. Hierzu wären ein binärer Halb- bzw. Volladdierer mitsamt Logikgattern zu implementieren, was einen wesentlich höheren Komplexitätsgrad aufweist. Dies wäre optimal für einen Aufbaukurs, kann aber für den Grundkurs nicht berücksichtigt werden. Aufgrund des doch komplexeren Schaltungsaufbaus und eines Mehr an Programmiercode erscheint es logisch, dass sich die Teilnehmenden mehr auf das Projekt selbst statt auf Adaptierungen fokussierten.

Beim Projekt 7 (*Messen und Auslesen des internen Lichtsensors*) werden keine Rückmeldungen von G1 und G2 generiert. Dies überrascht, denn hier wäre eine logische Kopplung mit einer LED erwartet worden. Aufgrund der etwas komplexeren Themen DAC und ADC erscheint es logisch, dass sich die Teilnehmenden mehr auf das Projekt selbst statt auf Adaptierungen fokussierten.

Beim Projekt 8 (*Programmierung und Inbetriebnahme eines LED-Stripes*) gibt es vier Antworten bei G1 und G2. Eine Bewertung findet das LED-Stripe motivierend, ein Hinweis auf den Begriff „Pixel“, der mit einem LED-Stripe auch gut erklärbar sei und der praktische Tipp für eine Weihnachtsbeleuchtung sind zwar alle wertvoll, allerdings sind es keine unmittelbaren Adaptierungen. Der vierte Vorschlag mit Farbbänderungen wurde sowohl bei G1 wie auch bei G2 bereits in der Schulung vor Ort umgesetzt. Aufgrund des etwas komplexeren Steckvorgangs am LED-Stripe und der etwas umfangreicheren Programmierung erscheint es logisch, dass sich die Teilnehmenden mehr auf das Projekt selbst statt auf Adaptierungen fokussierten.

Beim Projekt 10 (*Programmierung der drahtlosen Radio-Schnittstelle des Micro:bit*) gibt es keine direkten Antworten bei G1 und G2. Dieses überrascht, denn hier wären doch weitere Vorschläge erwartet worden. Aufgrund des doch proprietären Radio-Protokolls (wenig bekannt) erscheint es logisch, dass sich die Teilnehmenden mehr auf das Projekt selbst statt auf Adaptierungen fokussierten.

Beim Projekt 12 (*Programmierung einer Schaltung mit einem MOSFET*) gibt es zwei Antworten bei G1 und G2. Ein Vorschlag ist, den MOSFET falsch zu verbinden, wahrscheinlich um das resultierende Nichtfunktionieren zu demonstrieren. Ein anderer Vorschlag

ist, vor dem Gate ein Potentiometer zu schalten. Dieser Vorschlag ist sehr interessant, denn auf diese Weise könnte genau festgestellt bzw. geregelt werden, ab welcher Mindestspannung ein MOSFET leitend wird. Allerdings sind beide Hinweise schon als Adaptierung in der Projektbeschreibung angeführt. Aufgrund des etwas komplexeren Themas MOSFET erscheint es logisch, dass sich die Teilnehmenden mehr auf das Projekt selbst statt auf Adaptierungen fokussierten.

Fragen 9 und 10

Bei den Fragen 9 und 10 geht es nicht um direkte Änderungen des Projektes beim Aufbau, beim Design oder bei der Programmierung, sondern einfach um Ideen und sonstige Hinweise. Die Antworten werden gesammelt im Originalwortlaut wiedergegeben und dann gewertet:

- *„Vielleicht noch deutlicher hinweisen, dass die weiße Zeile die programmierbaren Pins sind. Und auch kurz physikalische Grundlagen, also dass egal wo Widerstände in der Reihe gesteckt werden“ (Projekt 1)*
- *„Immer genügend Widerstände und LEDs dabei haben, die Kontakte brechen in der Unterstufe sicher leicht ab“ (Projekt 1)*
- *„Klar erwähnen WARUM rot auf der + Linie ist“ (Projekt 1)*
- *„1. Erklärung warum man die roten Jumperkabel auf das Steckbrett umstellen muss auf die weiße Reihe. 2. Deutlicher erläutern, was den Unterschied ist zwischen den Anschluss der Jumperkabel auf die rote + Linie und in das Brett. (Physikalische Bedeutung)“ (Projekt 2)*
- *„Noch deutlicher auf Unterschied zwischen lesenden und schreibenden Pins bzw. wie das genau funktioniert“ (Projekt 4)*
- *„Für die Kids in OS erwähnen: for brightness in range(0, 1023): wenn nach 1023 kein Argument mehr kommt heißt es per default, dass die Variable +1 gerechnet wird, bei Sonderwünschen hier dann reinschreiben!“ (Projekt 5)*
- *„Zufällige LEDs ansteuern“ (Projekt 8)*
- *„Mit Mikrofon koppeln - Lautstärkeampel in der Klasse“ (Projekt 8)*
- *„cooles Beispiel für die Schüler:innen“ (Projekt 8)*

- „Wo sind überall LED-Ketten verwendet?“ (Projekt 8)
- „Evtl. für Klassenparty mit verschiedenen Beats der Musik koppeln“ (Projekt 8)
- „Cooles Beispiel, weil LED-Ketten halt oft verwendet werden --> könnte Lebenswelt der Kids betreffen“ (Projekt 8)
- „Erklärung Verschlüsselung evtl.“ (Projekt 10)
- „Die physikalischen Hintergründe sind unverzichtbar. Sehr gut, dass Tobias diese immer auch in der Fachsprache mitgeliefert hat. Danke euch beiden!“ (Projekt 11)
- „Ist schon tief in der Technik drin, für Fortgeschrittene“ (Projekt 11)

Viele der Anregungen resultieren aus inhaltlichen Unklarheiten der Teilnehmenden, wobei jedoch tatsächlich alle urgierten Klarstellungen in der Fortbildung selbst zumindest einmal erwähnt wurden. Wirklich sehr gut scheint den Teilnehmenden das Projekt 8 mit den LED-Stripes zu gefallen. Hier gibt es extra positive Hinweise und Ideen. Besonders wertvoll ist der Hinweis, nur einzelne LEDs anzusteuern. Dieser Hinweis wird entweder direkt in das Projekt 8 eingebaut oder als mögliche Projektadaptierung in der Projektbeschreibung vermerkt. Ein weiterer, wertvoller Hinweis ist die Idee für das Projekt 10, auch eine verschlüsselte Übertragung zu realisieren. Dies wäre mit MicroPython einfacher zu realisieren als mit MakeCode, da hier benötigte Blöcke fehlen. Es müsste eine gesonderte Funktion in Python oder JavaScript kodiert werden, um dies mit MakeCode zu kombinieren. Diese Idee wird tatsächlich beim Projekt 10 als Adaptierung in der Projektbeschreibung ergänzt bzw. wird ein gänzlich neues Projekt mit Verschlüsselung erstellt (dies aber außerhalb der Masterthesis).

9. Zusammenfassung und Ausblick

In der vorliegenden Arbeit geht es um die Frage, ob ein Einstieg in die sehr komplexen Themen Technische Informatik bzw. Digitaltechnik über eine ganztägige Fortbildung für DGB- und Informatiklehrpersonen erfolgreich geschafft werden kann. Technische Informatik und Digitaltechnik sind bereits im aktuellen Lehrplan für Digitale Grundbildung und Informatik teilweise verankert. Diese Themen sollen gemäß eines bereits vorliegenden Gesetzesvorentwurfs (nicht veröffentlicht) als überarbeiteter Informatik-Lehrplan eine noch bedeutendere Rolle einnehmen. Initiativen wie MINT, STEAM oder EU Code Week forcieren im schulischen Bereich die beschriebenen, technischen Richtungen. Das aktuelle EU-Projekt IPCEI fordert sogar eine eigene Halbleiterproduktion in der EU samt gut ausgebildeter Student*innen.

Die Fortbildung mit 10 Unterrichtseinheiten hat einen vorgegebenen Aufbau und Ablauf unter Zuhilfenahme des Mikrocontrollers Micro:bit. Dieser Mikrocontroller wird vorab im Hinblick auf Preis, Ausstattung und Unterstützung einer visuellen Programmiersprache evaluiert und für geeignet empfunden. Weiterhin werden durch die Wahl des Micro:bit auch die visuelle Programmiersprache MakeCode von Microsoft und die native Programmiersprache MicroPython priorisiert. Als Schaltungsdesign-Software ist das Tool TinkerCAD aufgrund seiner Popularität und Online-Funktionalität am besten geeignet.

Zehn vorgefertigte Projektbeschreibungen fordern konkret ein virtuelles Schaltungsdesign, Making durch den Bau der Schaltungen und visuelle und textuelle Programmierung der gewünschten Funktion. Diese Kombination illustriert, dass Technische Informatik bzw. Digitaltechnik mit Mathematik, Physik, Elektronik sowie Schaltungstechnik gebündelt und interdisziplinär unterrichtet werden sollte. Mit der Ergänzung von visueller und textueller Programmierung wird auch der pädagogische Ansatz von Physical Computing verständlich und wichtig. Um die Neuartigkeit dieses Ansatzes zu überprüfen, wurden folgende sieben Thesen formuliert:

Die Kombination von Design-Entwurf mit TinkerCAD, praktischem Aufbau am Steckbrett und die visuelle und textuelle Programmierung wird von mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 als erleichternd wahrgenommen. (These 1)

Die visuelle Programmierung mit MakeCode als Einstieg in die Programmierung wird von mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 als erleichternd wahrgenommen. (These 2)

Die visuelle Programmierung mit MakeCode erleichtert für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 das Verständnis für die textuelle Programmierung mit MicroPython. (These 3)

Die Kombination aus Unterricht und angeleitetem Nachbau und Erklärung der einzelnen Projekte ist für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 eine Hilfe für das Verständnis der Thematik "Computertechnik". (These 4)

Die fertig programmierten und dokumentierten Projektbeispiele als gespeicherte Sicherung (Backup) stellen für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 eine Erleichterung für den eigenen Unterricht dar. (These 5)

Der Gesamtzusammenhang von „Technischer Informatik“ mit der Kombination von Digital- und Analogtechnik und Programmierung ist für mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 aus den Projekten erkennbar. (These 6)

Das Übungsbeispiel X gibt mindestens 50 Prozent der Teilnehmenden von Schulung 1 und 2 Gelegenheit für Adaptierungen und neue Ideen. (These 7)

Diese sieben Thesen werden anhand der Evaluation von zwei ganztägigen Fortbildungen für Lehrpersonen durchgeführt. Die Evaluation zeigt, dass beide Schulungen von den Teilnehmenden sehr gut angenommen werden. Knapp 93 % der Teilnehmenden bewerten in Summe die Fragestellungen zu den sieben Thesen als „*sehr zutreffend*“ bzw. „*zutreffend*“. Alle sieben Thesen können mit weit mehr als den geforderten 50 Prozent bestätigt werden und beantworten damit die Forschungsfrage.

Auch die drei offenen und optionalen Fragen nach möglichen eigenen Adaptierungen bzw. nach Erweiterungsideen bringen neue Erkenntnisse. Vor allem das Projekt 8 (Ansteuern einer LED-Kette mit NeoPixel) wird von den Teilnehmenden als besonders motivierend und ausbaufähig eingestuft. Konkret wird hier die Programmierung der Einzel-

LEDs vorgeschlagen. Beim Projekt 10 (Radio-Protokoll) gibt es einen sehr wichtigen Hinweis auf die Möglichkeit einer verschlüsselten Übertragung der Daten.

Kritisch kann festgestellt werden, dass der Inhalt der Fortbildung die Teilnehmenden insgesamt stark in Anspruch nimmt. Herausfordernd sind für die Teilnehmenden die Themen Elektronik, Schaltkreise und vor allem auch die geforderte Kompetenz im praktischen Tun als Making (manueller Aufbau der Schaltungen).

Ausblickend kann festgehalten werden, dass die Evaluation der ersten zehn Projekte einen vielversprechenden Beginn eines gesonderten und geplanten Lehrbuchs darstellt. Die geschulten Lehrpersonen werden die evaluierten Inhalte in ihren SEK1- und SEK2-Schulen bereits ab dem Schuljahr 2024/2025 unterrichten. Es gibt auch berechtigten Anlass dazu, die Fortbildung generell allen interessierten SEK1- und SEK2-Schulen in Vorarlberg anzubieten.

10. Literatur- und Quellenverzeichnis

ADAFRUIT (2023): *Welcome to CircuitPython!* Online verfügbar unter: URL: <https://learn.adafruit.com/welcome-to-circuitpython/installing-circuitpython> [09.08.2023]

ADAFRUIT INDUSTRIES (2023): *Circuit Playground Express*. Online verfügbar unter: URL: <https://www.adafruit.com/product/3333> [09.08.2023]

ALSABBAGH, Marwan (2019): *MicroPython cookbook: over 110 practical recipes for programming embedded systems and microcontrollers with Python*. Birmingham, UK: Packt

ARAVINDAN, Avinash (2018): Flash 101: Types of NAND Flash. *Embedded.com* Online verfügbar unter: URL: <https://www.embedded.com/flash-101-types-of-nand-flash/> [09.07.2024]

ARDUINO (2023a): *Software*. Online verfügbar unter: URL: <https://www.arduino.cc/en/software> [07.08.2023]

ARDUINO (2023b): *MicroPython with Arduino | Arduino Documentation*. Online verfügbar unter: URL: <https://docs.arduino.cc/micropython/> [07.08.2023]

ARDUINO (2023c): *Tutorials | Arduino Documentation*. Online verfügbar unter: URL: <https://docs.arduino.cc/tutorials/> [07.08.2023]

ARDUINO (2023d): *Arduino R3 Uno*. Online verfügbar unter: URL: <https://docs.arduino.cc/static/182b8c4f8af5d1a952d82bd6486dd9a7/A000066-datasheet.pdf> [07.08.2023]

ARDUINO (2023e): *Nano RP2040 Connect | Arduino Documentation*. Online verfügbar unter: URL: <https://docs.arduino.cc/hardware/nano-rp2040-connect> [08.08.2023]

ARDUINO (2023f): *Arduino Reference - Arduino Reference*. Online verfügbar unter: URL: https://www.arduino.cc/reference/en/?_gl=1*_zq2yit*_ga*MjA4MDE-wODQzOC4xNjkwNTI1OTI2*_ga_NEXN8H46L5*MTY5MTYxNDg0Ni4xMC4xLjE2OTE2MTQ5NDkuMC4wLjA [09.08.2023]

ARDUINO (2023g): *Libraries - Arduino Reference-Libraries*. Online verfügbar unter: URL: <https://www.arduino.cc/reference/en/libraries/> [09.08.2023]

ARDUINO.CC (2023): *ABX00053-datasheet. Arduino® Nano RP2040 Connect*. Online verfügbar unter: URL: <https://docs.arduino.cc/static/9cc8c1d53ad2f9fd10a6b155801b7d59/ABX00053-datasheet.pdf> [08.08.2023]

BERNSTEIN, Herbert (2019): *1. Grundlagen der Digitaltechnik*. Berlin, München, Boston: De Gruyter [06.07.2024]

BMBWF (2023): *MINT_HS*. Online verfügbar unter: URL: <https://www.bmbwf.gv.at/Themen/HS-Uni/Hochschulgovernance/Leitthemen/MINT.html> [04.08.2023]

BMK INFOTHEK (2023): Startschuss für größte europäische Mikroelektronik-Initiative unter österreichischer Beteiligung. *bmvit INFOTHEK* Online verfügbar unter: URL:

<https://infothek.bmk.gv.at/ipcei-mikroelektronik-initiative-oesterreich-beteiligung/>
[06.08.2023]

BOSCH, Karl (1999): *Grundzüge der Statistik: Einführung mit Übungen*. (2., erg. Aufl.). München Wien: Oldenbourg

BUNDESKANZLERAMT (2023): „*European Chips Act*“: *Parlament nimmt Gesetz zur Stärkung der Chip-Industrie in der EU an - Bundeskanzleramt Österreich*. Online verfügbar unter: URL: <https://www.bundestkanzleramt.gv.at/themen/europa-aktuell/2023/07/parlament-nimmt-gesetz-zur-staerkung-der-chip-industrie-in-der-eu-an.html>
[06.08.2023]

BUNDESMINISTERIUM FÜR BILDUNG, WISSENSCHAFT UND FORSCHUNG (2022): *Der österreichische Hochschulplan 2030*. Online verfügbar unter: URL: <https://www.bmbwf.gv.at/Themen/HS-Uni/Hochschulgovernance/Steuerungsinstrumente/hochschulplan.html>

BUNDESMINISTERIUM FÜR BILDUNG, WISSENSCHAFT UND FORSCHUNG (2023a): *Digitale Grundbildung*. Online verfügbar unter: URL: <https://www.bmbwf.gv.at/Themen/schule/zrp/dibi/dgb.html> [30.07.2023]

BUNDESMINISTERIUM FÜR BILDUNG, WISSENSCHAFT UND FORSCHUNG (2023b, 6. Oktober): *Digitale Grundbildung im Lehrplan der Allgemeinen Sonderschule*. (BUNDESMINISTERIUM FÜR BILDUNG, WISSENSCHAFT UND FORSCHUNG, Hrsg.). Bundesministerium für Bildung, Wissenschaft und Forschung Online verfügbar unter: URL: https://pubshop.bmbwf.gv.at/index.php?rex_media_type=pubshop_download&rex_media_file=231006_final_endversion_digib_aso_barrierefrei_1.pdf

CITIZEN SCIENCE (2022): *Global Citizen Science Partnership*. Online verfügbar unter: URL: <http://129.194.10.75/index.html> [04.08.2023]

CLARK, Charles (2022): *Basic DAC and ADC Circuits - How to Build*. *Circuits Gallery* Online verfügbar unter: URL: <https://www.circuitgallery.com/basic-dac-and-adc-circuits/> [07.07.2024]

CODE WEEK (2021): *EU Code Week Challenges*. Online verfügbar unter: URL: https://education.at/news/news-detail?tx_news_pi1%5Baction%5D=detail&tx_news_pi1%5Bcontroller%5D=News&tx_news_pi1%5Bnews%5D=642&cHash=e9893e6be9bf34a19ce2313f4dd44f98 [05.08.2023]

CODE WEEK (2023a): *Code Week über uns*. Online verfügbar unter: URL: <https://code-week.eu/about> [05.08.2023]

CODE WEEK (2023b): *EU Code Week Events*. Online verfügbar unter: URL: <https://codeweek.eu/events> [05.08.2023]

EDLINGER, Thomas (2020, 21. November): *Wie funktioniert I2C - einfach erklärt!* *Edis Techlab* Online verfügbar unter: URL: <https://edistechlab.com/wie-funktioniert-i2c/>
[25.08.2023]

EEDUCATION (2022): *microbit - Das Schulbuch*. Online verfügbar unter: URL: <https://microbit.eeducation.at/wiki/Hauptseite> [05.08.2023]

ELEKTRONIK-KOMPENDIUM (2024a): *PWM - Pulsweitenmodulation (PDM PLM PBM)*.

Online verfügbar unter: URL: <https://www.elektronik-kompensium.de/sites/kom/0401111.htm> [07.07.2024]

ELEKTRONIK-KOMPENDIUM (2024b): *RS-Flip-Flop / SR-Flip-Flop (NOR / NAND)*. Online verfügbar unter: URL: <https://www.elektronik-kompensium.de/sites/dig/0209302.htm> [09.07.2024]

ELEKTRONIK-KOMPENDIUM (2024c): *DRAM - Dynamic RAM*. Online verfügbar unter: URL: <https://www.elektronik-kompensium.de/sites/com/0701281.htm> [09.07.2024]

ELEKTRONIK-KOMPENDIUM (2024d): *Flash-Speicher / Flash-Memory*. Online verfügbar unter: URL: <https://www.elektronik-kompensium.de/sites/com/0312261.htm> [09.07.2024]

ESPRESSIF (2023): *esp32_datasheet_en.pdf*. Online verfügbar unter: URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf [08.08.2023]

EUROPÄISCHE KOMMISSION (2023): *Europäisches Chip-Gesetz*. Online verfügbar unter: URL: https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-chips-act_de [06.08.2023]

HODGES, Steve, SENTANCE, Sue, FINNEY, Joe, BALL, Thomas (2020): Physical Computing: A Key Element of Modern Computer Science Education. *Computer (Long Beach, Calif.)*, 53 (4), S. 20–30

HOFFMANN, Dirk W. (2023): *Grundlagen der Technischen Informatik*. (7., aktualisierte Auflage.). München: Hanser

HQNEXTPCB (2023): *SPI vs. I2C vs. UART: Unterschiede zwischen diesen Kommunikationsschnittstellen*. Online verfügbar unter: URL: <https://www.nextpcb.com/blog/spi-i2c-uart> [25.08.2023]

HÜRZELER, Daniel, MICHEL, Markus, REBER, Corinne, REGEZ, Adrian, SCHÄR, Sonja, ZAUGG, Pascal (2023): *11 Prinzipien guten Medien & Informatik Unterrichts: Ein Diskussionsanstoss*. PHBern Online verfügbar unter: URL: <https://phrepo.phbern.ch/6923/> [01.08.2023]

INTERNET-ABC (2020): *Making*. Online verfügbar unter: URL: <https://www.internet-abc.de/eltern/lexikon/m/making/> [01.08.2023]

JOOS, Thomas (2021): *Funktionen und Methoden in Python*. Online verfügbar unter: URL: <https://www.dev-insider.de/funktionen-und-methoden-in-python-a-0cf3eed3231c6a229e857ac80adb6bd7/> [09.08.2023]

KAMPERT, Daniel, SCHERBECK, Christoph (2017): *Elektronik verstehen mit Raspberry Pi: der praktische Einstieg*. (1. Auflage.). Bonn: Rheinwerk Verlag GmbH

KERSKEN, Sascha (2016): *IT-Handbuch für Fachinformatiker: der Ausbildungsbegleiter*. (7., aktualisierte und erweiterte Auflage 2015, 1. korrigierter Nachdruck 2016.). Bonn: Rheinwerk Verlag GmbH

KNAUS, Thomas, SCHMIDT, Jennifer (2020): Medienpädagogisches Making. *Medienimpulse*, S. 50

KOFLER, Michael, KÜHNAST, Charly, SCHERBECK, Christoph (2016): *Raspberry Pi: das umfassende Handbuch*. (3., aktualisierte Auflage.). Bonn: Rheinwerk Computing

- LANCASTER UNIVERSITY (2024): *radio - micro:bit runtime*. Online verfügbar unter: URL: <https://lancaster-university.github.io/microbit-docs/ubit/radio/> [13.07.2024]
- LIPP, Hans Martin, BECKER, Jürgen (2011): *Grundlagen der Digitaltechnik*. Oldenbourg Wissenschaftsverlag Online verfügbar unter: URL: <https://www.degruyter.com/document/doi/10.1524/9783486706932/html> [06.08.2023]
- LUBER, Stefan (2024): *Was ist ein Floating-Gate-Transistor (FGT) / Floating-Gate-MOSFET (FGMOS)?* Online verfügbar unter: URL: <https://www.storage-insider.de/was-ist-ein-floating-gate-transistor-fgt-floating-gate-mosfet-fgmos-a-9541a0dcd9523c18ea7e9b6749112ed/> [10.07.2024]
- MAKEBLOCK (2023a): *About Makeblock / Global STEM Education Company*. Online verfügbar unter: URL: <https://www.makeblock.com/pages/about-us> [08.08.2023]
- MAKEBLOCK (2023b): *CyberPi Go Kit*. Online verfügbar unter: URL: <https://docs.rs-online.com/a324/A700000007193134.pdf> [09.08.2023]
- MANSHOLT, Malte (2021): „*Programmieren ist für Kinder wichtiger, als eine zweite Sprache zu lernen*“. Online verfügbar unter: URL: <https://www.capital.de/wirtschaftspolitik/programmieren-ist-fuer-kinder-wichtiger-als-eine-zweite-sprache-zu-lernen> [05.08.2023]
- MICRO:BIT (2023): *Hardware, Details of the latest micro:bit hardware revision*. Online verfügbar unter: URL: <https://tech.microbit.org/hardware/> [08.08.2023]
- MICRO:BIT (2024): *BBC micro:bit overview*. Online verfügbar unter: URL: <https://microbit.org/get-started/features/overview/> [05.07.2024]
- MICROSOFT (2024): *Microsoft MakeCode*. Online verfügbar unter: URL: <https://www.microsoft.com/en-us/makecode> [04.07.2024]
- MIKROCONTROLLER (2024): *Speicher – Mikrocontroller.net*. Online verfügbar unter: URL: <https://www.mikrocontroller.net/articles/Speicher#SRAM> [09.07.2024]
- MIKROPYTHON (2023): *Glossary — MicroPython latest documentation - REPL - WebREPL*. Online verfügbar unter: URL: <https://docs.micropython.org/en/latest/reference/glossary.html#term-REPL> [09.08.2023]
- MINT SCHULE (2023): *Warum MINT?* Online verfügbar unter: URL: <https://www.mint-schule.at/warum-mint/> [04.08.2023]
- MINTSCHULE.AT (2024): *MINT_Guetesiegel_Bewertungskriterien_Ersteinreichung_2024-27.pdf*. Online verfügbar unter: URL: https://www.mintschule.at/wp-content/uploads/MINT_Guetesiegel_Bewertungskriterien_Ersteinreichung_2024-27.pdf [17.07.2024]
- MIT (2024): *Physical Computing*. Online verfügbar unter: URL: <https://k12maker.mit.edu/physical-computing.html> [10.07.2024]
- MONK, Simon (2016): *Das Action-Buch für Maker: Bewegung, Licht und Sound mit Arduino und Raspberry Pi - Experimente und Projekte*. (1. Auflage.). Heidelberg: Dpunkt.verlag
- MURPHY, Brendan (2022): *Analog vs. Digital Signals (ADC vs. DAC)*. Online verfügbar unter: URL: <https://www.totalphase.com/blog/2022/06/analog-vs-digital-signals-adc-vs->

dac/ [07.07.2024]

NORDIC SEMICONDUCTOR (2024): *nRF52833_OPS_v0.7.pdf*. Online verfügbar unter: URL: https://infocenter.nordicsemi.com/pdf/nRF52833_OPS_v0.7.pdf [08.07.2024]

OBERMAYR, Volker (2023): *AT&S-Chef Andreas Gerstenmayer im Ö1-„Journal zu Gast“*. Online verfügbar unter: URL: <https://oe1.orf.at/player/20230805/729008/1691231048167> [06.08.2023]

OEAD (2023a): *Schule - Unterstufe - Das österreichische Bildungssystem*. Online verfügbar unter: URL: <https://www.bildungssystem.at/schule-unterstufe> [30.07.2023]

OEAD (2023b): *Citizen Science*. Online verfügbar unter: URL: <https://zentrumfuercitizenscience.at/de/citizen-science> [04.08.2023]

ORACLE (2023): *Was versteht man unter Internet of Things (IoT)?* Online verfügbar unter: URL: <https://www.oracle.com/at/internet-of-things/what-is-iot/> [30.07.2023]

ORF (2023): *Arbeitsgruppen zu Programmieren an Schulen eingerichtet*. Online verfügbar unter: URL: <https://orf.at/stories/3308533/> [05.08.2023]

PLATT, Charles (2022): *Make: Elektronik: eine unterhaltsame Einführung für Maker, Kids, Tüftlerinnen und Bastler*. (Frank LANGENAU, Übers.) (3., komplett überarbeitete Auflage.). Heidelberg: dpunkt.verlag

RACHELLE (2012): *Why is Tinkering Important?* Online verfügbar unter: URL: <https://tinkerlab.com/why-is-tinkering-important/> [01.08.2023]

REICHL, Peter (2023): *Eine kurze Geschichte der Technischen Informatik : Wie ein Computer funktioniert – anschaulich erklärt*. (1. Auflage 2023.). Wiesbaden: Springer Fachmedien Wiesbaden GmbH Springer Vieweg

ROHDE & SCHWARZ GMBH & CO KG (2023): *Understanding UART*. Online verfügbar unter: URL: https://www.rohde-schwarz.com/cz/products/test-and-measurement/essentials-test-equipment/digital-oscilloscopes/understanding-uart_254524.html [24.08.2023]

ROUSE, Margaret (2023): *Kommunikationsprotokoll. Techopedia Deutschland* Online verfügbar unter: URL: <https://www.techopedia.com/de/definition/kommunikationsprotokoll> [24.08.2023]

RS (2023): *Alles, was Sie über Mikrocontroller wissen müssen | RS Components*. Online verfügbar unter: URL: <https://at.rs-online.com/web/content/discovery-portal/produktatgeber/mikrocontroller-leitfaden#die-wichtigsten-mikrocontroller-im-%C3%BCberblick> [06.08.2023]

SACHS, Michael (2007): *Wahrscheinlichkeitsrechnung und Statistik: für Ingenieurstudenten an Fachhochschulen ; mit 93 Beispielen und 71 Aufgaben*. (2., erw. Aufl.). München: Fachbuchverl. Leipzig im Hanser-Verlag

SCHULBUCHAKTION (2024): *Schulbuchaktion Online*. Online verfügbar unter: URL: <https://www.schulbuchaktion.at/guidelines> [04.07.2024]

SIEMERS, Christian, SIKORA, Axel (Hrsg.) (2007): *Taschenbuch Digitaltechnik: mit 46 Tabellen*. (2., vollst. überarb. Aufl.). München Wien: Fachbuchverl. Leipzig im Carl Hanser Verl

STADLER, David, GRÜNBERGER, Nina (2021): Vom Hin und Her zwischen Medienbildung und Informatischer Bildung: Ein Versuch einer systematischen Verbindung für die Schulpraxis. *MedienPädagogik: Zeitschrift für Theorie und Praxis der Medienbildung*, S. 89–105

The Micro:bit Educational Foundation (2023): Online verfügbar unter: URL: <https://microbit.org/about/> [08.08.2023]

THE NATIONAL INTELLIGENCE COUNCIL (2008): Disruptive civil technologies - Six Technologies With Potential Impacts on US Interests Out to 2025., S. 1–33

VIES, Peter (2024): *4 Bit Shift Register*. Online verfügbar unter: URL: <https://www.petervis.com/dictionary-of-digital-terms/4-bit-shift-register/4-bit-shift-register.html> [09.07.2024]

WINDECK, Christoph (2021): *Was Mikrocontroller von Mikroprozessoren unterscheidet*. Online verfügbar unter: URL: <https://www.heise.de/hintergrund/Was-Mikrocontroller-von-Mikroprozessoren-unterscheidet-6237074.html> [06.08.2023]

WÖSTENKÜHLER, Gerd (2024): *Grundlagen der Digitaltechnik: elementare Komponenten, Funktionen und Steuerungen*. (3., überarbeitete und erweiterte Auflage.). München: Hanser

YALE UNIVERSITY (2024): *Flash Drives: methods and materials* | *Ismail-Beigi Research Group*. Online verfügbar unter: URL: <https://volga.eng.yale.edu/teaching-resources/flash-drives/methods-and-materials> [10.07.2024]

11. Abbildungsverzeichnis

Abbildung 1: Teilbereiche der Digitaltechnik (eigene Abbildung).....	6
Abbildung 2: Informationen als physikalische Größe mit möglicher Änderung der Parameter (LIPP & BECKER, 2011, S. 25).....	12
Abbildung 3: Darstellung der Zahl 583 durch Potenzen zur Basis 10 (BERNSTEIN, 2019, S. 2)	13
Abbildung 4: Darstellung eines Nibble, links das MSB, recht das LSB (BERNSTEIN, 2019, S. 8)	14
Abbildung 5: Wandlung und Rückwandlung von analogen Signalen (CLARK, 2022)	16
Abbildung 6: Darstellung PWM (ELEKTRONIK-KOMPENDIUM, 2024a)	17
Abbildung 7: Wahrheitstabelle samt logischer Operatoren (REICHL, 2023, S. 12)	18
Abbildung 8: Wahrheitswerttabelle der binären Addition (REICHL, 2023, S. 8)	19
Abbildung 9: Ein Halbaddierer als logisches Gatter (REICHL, 2023).....	19
Abbildung 10: Schaltungsentwurf mit Pin 0, Pin 1, XOR- und AND-Gate (eigene Abbildung).....	20
Abbildung 11: Programmierung eines Halbaddierers mit MicroPython (eigene Abbildung).....	21
Abbildung 12: Mögliche Halbleiterschaltung (eigene Abbildung, eigener Aufbau)	21
Abbildung 13: Aufbau eines RTCs (NORDIC SEMICONDUCTOR, 2024, S. 343).....	22
Abbildung 14: Übertragung des Buchstaben S (ASCII-Code dezimal 83) (ROHDE & SCHWARZ GMBH & Co KG, 2023)	23
Abbildung 15: Sende- und Empfangskomponenten TX und RX (ROHDE & SCHWARZ GMBH & Co KG, 2023).....	24
Abbildung 16: Aufbau einer SPI-Verbindung mit 3 Slaves (HQNEXTPCB, 2023)	25
Abbildung 17: Grundaufbau einer I2C-Kommunikation mit vier Geräten (EDLINGER, 2020)	26

Abbildung 18: Offene Übertragung mit dem Radio-Protokoll (LANCESTER UNIVERSITY, 2024)	27
Abbildung 19: RS-Flip-Flop samt Wahrheitstabelle (ELEKTRONIK-KOMPENDIUM, 2024b)....	28
Abbildung 20: Aufbau eines DRAM (ELEKTRONIK-KOMPENDIUM, 2024c).....	29
Abbildung 21: 4-Bit-Register (VIES, 2024).....	30
Abbildung 22: Gegenüberstellung eines herkömmlichen MOSFET-Halbleiters mit einem Floating-Gate-Transistor (ARAVINDAN, 2018).....	32
Abbildung 23: RP Pico und RP 4 als Mikrocontroller und Mikroprozessor (WINDECK, 2021)	34
Abbildung 24: Strategien und Kompetenzen von DGB (VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 1. Unterstufe, Digitale Grundbildung, Zentrale fachliche Kompetenz (1. bis 4. Klasse)).....	37
Abbildung 25: Digitaltechnik und Programmierung DGB 5. Schulstufe VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 1. Unterstufe, Digitale Grundbildung, zentrale fachliche Kompetenz (1. bis 4. Klasse))	38
Abbildung 26: Kompetenzbereich Information mit Verweis auf binäre Struktur (VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 1. Unterstufe, Digitale Grundbildung, Zentrale fachliche Kompetenz (1. bis 4. Klasse))	39
Abbildung 27: DGB 8. Schulstufe Kompetenz Produktion (VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 1. Unterstufe, Digitale Grundbildung, Kompetenzbeschreibungen und Anwendungsbereiche, Lehrstoff (1. bis 4. Klasse), 4. Klasse).....	39
Abbildung 28: Pflichtfächer Oberstufe AHS (VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 2. Oberstufe, Pflichtgegenstände).....	40
Abbildung 29: Wahlpflichtfach Informatik über drei Jahre (VO BGBl. Nr. 88/1985 idF BGBl. II Nr. 1/2023, Anlage A, vierter Teil, Stundentafeln, 2. Oberstufe, Wahlpflichtgegenstände)	41
Abbildung 30: Lehrplaninhalte Pflichtfach Informatik 9. Schulstufe VO BGBl. Nr. 88/1985	

idF BGBl. II Nr. 1/2023, Anlage A, fünfter Teil, FÜNFTER TEIL, 2. Oberstufe, Informatik)	42
Abbildung 31: Informatische Konzepte des legistischen Vorentwurfs „Informatik Oberstufe NEU“ (siehe Anhang 13.1)	43
Abbildung 32: Kriterien Sekundarstufe für das MINT-Gütesiegel (MINTSCHULE.AT, 2024)	45
Abbildung 33: Unterstützte Datentypen und Module von MicroPython (MICROPYTHON, 2023)	57
Abbildung 34: Parallelschaltung mit 2 LEDs (eigene Abbildung)	60
Abbildung 35: Spannungsänderung über Pin 0 (eigene Abbildung)	61
Abbildung 36: Ampelschaltung mit 3 digitalen PINs (eigene Abbildung)	62
Abbildung 37: Einlesen und Ausgabe der Spannung (eigene Abbildung)	63
Abbildung 38: Programmcode für eine PWM-Modulation in Python (eigene Abbildung)	64
Abbildung 39: Listing zur binären Darstellung durch Leuchten der LEDs (eigene Abbildung)	65
Abbildung 40: Messung der Lichtstärke mit Schalten einer LED (eigene Abbildung)	66
Abbildung 41: Entwurf des Anschlusses der Lichterkette am Micro:bit (eigene Abbildung)	67
Abbildung 42: Übertragungsfunktion, programmiert in MakeCode (eigene Abbildung)	68
Abbildung 43: Schaltungsaufbau des MOSFET (eigene Abbildung)	69
Abbildung 44: Micro:bit V2 mit wichtigen Komponenten (MICRO:BIT, 2024)	73
Abbildung 45: MakeCode-Editor für den Micro:bit (eigene Abbildung)	74
Abbildung 46: Online-Editor für MicroPython (eigene Abbildung)	75
Abbildung 47: Teilauszug der Forms-Umfrage (eigene Abbildung)	77
Abbildung 48: Anzahl "weiß nicht" über alle 10 Projekte	80

Abbildung 49: Häufigkeiten Frage 1 (G1 und G2)	81
Abbildung 50: Grafische Darstellung der absoluten Häufigkeiten für Frage 1 (eigene Abbildung)	82
Abbildung 51: Häufigkeiten Frage 2 (G1 und G2)	83
Abbildung 52: Grafische Darstellung der absoluten Häufigkeiten für Frage 2 (eigene Abbildung)	84
Abbildung 53: Häufigkeiten Frage 3 (G1 und G2)	85
Abbildung 54: Grafische Darstellung der absoluten Häufigkeiten für Frage 3 (eigene Abbildung)	86
Abbildung 55: Häufigkeiten Frage 4 (G1 und G2)	87
Abbildung 56: Grafische Darstellung der absoluten Häufigkeiten für Frage 4 (eigene Abbildung)	88
Abbildung 57: Häufigkeiten Frage 5 (G1 und G2)	89
Abbildung 58: Grafische Darstellung der absoluten Häufigkeiten für Frage 5 (eigene Abbildung)	90
Abbildung 59: Häufigkeiten Frage 6 (G1 und G2)	91
Abbildung 60: Grafische Darstellung der absoluten Häufigkeiten für Frage 6 (eigene Abbildung)	92
Abbildung 61: Häufigkeiten Frage 7 (G1 und G2)	93
Abbildung 62: Grafische Darstellung der absoluten Häufigkeiten für Frage 7 (eigene Abbildung)	94

12. Tabellenverzeichnis

Tabelle 1: Erklärung wichtiger Begriffe der Informationsdarstellung (eigene Tabelle) .	15
Tabelle 2: Kriterien für Mikrocontroller (eigene Tabelle).....	72
Tabelle 3: Fragen der Evaluation	77
Tabelle 4: Formulierung von sieben Thesen zur Beantwortung der Forschungsfrage (eigene Tabelle)	79
Tabelle 5: Häufigkeitstabelle für Frage 1	82
Tabelle 6: Häufigkeitstabelle für Frage 2	84
Tabelle 7: Häufigkeitstabelle für Frage 3	86
Tabelle 8: Häufigkeitstabelle für Frage 4	88
Tabelle 9: Häufigkeitstabelle für Frage 5	90
Tabelle 10: Häufigkeitstabelle für Frage 6	92
Tabelle 11: Häufigkeitstabelle für Frage 7	94

13. Anhang

13.1 Legistischer Vorentwurf Lehrplan Informatik AHS-Oberstufe

INFORMATIK

Bildungs- und Lehraufgabe (5. bis 8. Klasse):

Bildungsziele und Bildungsinhalte sind immer ein Spiegelbild des gesellschaftlichen, politischen und ökonomischen Umfeldes. Die Informatik hat seit dem letzten Jahrhundert maßgeblich zu bahnbrechenden Entwicklungen beigetragen und prägt wesentlich die Lebenswelt, Kultur, Gesellschaft, Arbeitswelt, Wirtschaft, Produktion sowie Technik. Sie spielt ebenso eine zentrale Rolle bei der Unterstützung und Erweiterung menschlicher Denk- und Handlungsfähigkeiten und prägt als wissenschaftliche Disziplin Selbstbilder, Weltverständnis und zukünftige Berufsbilder.

Informatische Bildung ist das Ergebnis von Lernprozessen, in denen fachliche Grundlagen verdeutlicht und Anwendungskompetenzen durch planvolle Arbeitsweisen systematisch erworben werden. Sie befähigt Schülerinnen und Schüler, die gesellschaftlich, wirtschaftliche und technologische Dimension digitaler Informations- und Kommunikationstechnologien zu erfassen.

Durch die Erschließung realer Probleme sowie Phänomene (digitaler Artefakte) vor allem aus ihrer Erfahrungswelt sollen Schülerinnen und Schüler Strukturen und Zusammenhänge erkennen, handlungsorientiert gestalten, und reflektieren. Es sollen dazu kreative, kooperative, kommunikative und kritisch reflektierende Arbeitsweisen Anwendung finden.

Die Schülerinnen und Schüler sollen erkennen, dass die Informatik einer wissenschaftlichen Systematik unterliegt und Interesse und Wertschätzung verdient. Der Informatikunterricht fasst die vorhandenen Fähigkeiten von Schülerinnen und Schülern in der Informatik durch Beschäftigung mit Entwurf, Gestaltung und Anwendung von Informationssystemen zusammen und baut diese aus. Bei der kritischen Auseinandersetzung mit den dabei ablaufenden Prozessen und deren Ergebnissen sollen die Schülerinnen und Schüler ihr kognitives, emotionales und kreatives Potenzial nützen. Dies soll die Jugendlichen bei der Entwicklung und Festigung einer persönlichen Werthaltung und Weltsicht unterstützen und einen tieferen Einblick in die gesellschaftlichen Zusammenhänge und Auswirkungen moderner Informationstechnologie ermöglichen.

Das Fach Informatik bietet vielfache Möglichkeiten zum fächerverbindenden Unterricht. Lehrende sind angehalten aktuelle Themen und Entwicklungen mit Schülerinnen und Schülern kritisch und reflektiert aufzugreifen und diese gemeinsam zu behandeln.

Beiträge zu den Bildungsbereichen

Sprache und Kommunikation

Konstruktiver Informatikunterricht ist auch Sprachunterricht. Der Mensch-Maschine-Kommunikation liegt im Gegensatz zu natürlichen Sprachen eine abstrakte formale Sprache zugrunde.

Informatiksysteme tragen wesentlich zu Veränderungen der Kommunikationskultur bei. Unterschiedliche digitale Repräsentationsformen von Information ergänzen die traditionelle Verständigung und erfordern neue technologische und methodische Kompetenzen. Ebenso ermöglichen generativen KIs neue Möglichkeiten der Interaktion.

Die vielfältigen Möglichkeiten der digitalen Kommunikation ermöglichen einen Austausch über Grenzen hinweg und erleichtern die virtuelle Begegnung mit anderen Kulturen. Die davon ausgehende Motivation, Fremdsprachenkenntnisse zu erwerben, wird durch die Verfügbarkeit aktueller und authentischer fremdsprachlicher Informationen und das Fachvokabular verstärkt.

Mensch und Gesellschaft

Arbeitswelt und privates Umfeld der Menschen verändern sich durch den Einfluss der Informationstechnologien permanent. Durch die Beschäftigung mit diesen Technologien lernen Schülerinnen und Schüler deren Auswirkungen, Möglichkeiten, Grenzen und Gefahren kennen.

Die Schülerinnen und Schüler nutzen das Potenzial ihrer eigenen Fähigkeiten und erleben Selbstwirksamkeit in der Interaktion mit Maschinen. Dies erfordert einen verantwortungsvollen Umgang mit Informationstechnologien.

Natur und Technik

Durch ihren Anwendungsbezug leistet die Informatik einen wesentlichen Beitrag zur Auseinandersetzung mit Umwelt, Natur und Technik und führt zu einer besseren Entscheidungs- und Handlungskompetenz. Sie leistet damit einen wesentlichen Beitrag zu nachhaltigen Themenbereichen der Ökologie, Ökonomie und Soziales.

Kreativität und Gestaltung

Der Umgang mit Informationstechnologie gibt den Schülerinnen und Schülern Gelegenheit, selbst kreativ tätig zu sein und Gestaltungskompetenzen zu erwerben und zu vertiefen.

Gesundheit und Bewegung

Die Auseinandersetzung mit informatischen Systemen sowie digitalen Artefakten impliziert auch die Berücksichtigung von Gesundheit, Wohlbefinden und Inklusion.

Der Einsatz von Informationstechnologien zur Erfassung und Analyse von Daten in der Freizeit, im Sport- oder Gesundheitsbereich bietet zusätzliche Möglichkeiten zur kritischen Reflexion.

Didaktische Grundsätze (5. bis 8. Klasse):

Der Lehrplan bietet den Lehrerinnen und Lehrern Freiräume für die eigenständige und verantwortliche Unterrichtsgestaltung, in der eine ausgewogene Abdeckung aller Kompetenzbereiche anzustreben ist. Um alle Themenbereiche in passender Tiefe zu vermitteln wird besonders auf die gerade im Informatikunterricht besonders leicht zu implementierende Möglichkeit von Projektarbeiten hingewiesen. Dabei sind die Themen und Inhalte so auszuwählen und zu organisieren, dass sie die Vorkenntnisse und Vorerfahrungen der Schülerinnen und Schüler berücksichtigen und daran anknüpfen. Die Themen sollen dabei vielfältige Bezüge zur Lebens- und Begriffswelt der Jugendlichen herstellen. Im Informatikunterricht sind besondere fachdidaktische Überlegungen anzustellen um Defizite aus den vorangehenden Schulstufen auszugleichen und individuelle Stärken sowie Interessen einzubinden und zu fördern.

Die Unterrichtsplanung hat sich an für Schülerinnen und Schüler transparenten Lehrzielen zu orientieren. Variierende Arbeitsformen wie Einzelarbeit, Gruppenarbeit und Teamarbeit geben Schülerinnen und Schülern Gelegenheit, Neues zu erforschen und bereits Gelerntes in verschiedenen kommunikativen und inhaltlichen Kontexten anzuwenden. Selbsttätigkeit und Eigenverantwortung sind zu fördern und Möglichkeiten zur persönlichen Lernzielkontrolle anzubieten.

Gemeinschaftliches Problemlösen in einem projektorientierten Unterricht soll gefördert werden. Dabei ist kooperativen Entscheidungsstrukturen entsprechender Platz einzuräumen. Methodische Überlegungen sollen sich an den spezifischen Anforderungen von Einstieg, Entwicklung und Abschluss von Unterrichtsphasen orientieren. Explorative, systematische und exemplarische Vorgehensweisen sollen zur Vertiefung von Wissen und Erweiterung von Kompetenzen in der Informatik führen.

Schülerinnen und Schülern ist Gelegenheit zu geben, durch Transfer und Analogiebildung den Lernertrag zu sichern. Der Informatikunterricht soll beispielhaft für den sinnvollen Einsatz verfügbarer Technologien sein. Dem Erwerb einer technischen und wissenschaftlichen Arbeits- und Dokumentationsweise sind prozessorientierte Dokumentationen (wie zB Portfolio und Versionskontrollsysteme) dienlich.

Zur Motivation und zur Sicherung des Unterrichtsertrags sind den Schülerinnen und Schülern im Rahmen des Informatikunterrichts vielfältige Möglichkeiten anzubieten, ihr Wissen zu präsentieren, sich der Kritik anderer zu stellen und ihre Arbeit zu argumentieren.

Die Gestaltung eines angenehmen und erfolgreichen Lernklimas inklusive einer positiv besetzten Fehlerkultur beruht auf Vertrauen, auf der Förderung der individuellen Stärken und des kreativen Potenzials.

Exkursionen und Einladungen von Expertinnen und Experten zu Vorträgen und zur Diskussion sollen den Erfahrungshorizont erweitern.

Bildungs- und Lehraufgabe, Lehrstoff:

Dieser Lehrplan folgt den Empfehlungen der informatics for all coalition (einem Zusammenschluss des ACM Europe Councils, CEPIS - Council of European Professional Informatics Societies - und Informatics Europe) und hat eine vertiefende Allgemeinbildung zum Ziel. Am Ende der Sekundarstufe II werden Schülerinnen und Schüler:

- Digitale Werkzeuge bewusst, verantwortungsvoll, selbstsicher, kompetent und kreativ nutzen.

-
- Die Phänomene, Konzepte, Prinzipien und Praktiken der Informatik verstehen und die vielfältigen Anwendungsmöglichkeiten zur Modellierung, Interpretation und Bearbeitung der Realität erkennen.
 - Probleme analysieren, entwerfen, rahmen und lösen, indem sie Darstellungen erfinden, algorithmische Lösungen entwerfen und diese in einer Programmiersprache implementieren.
 - Computergestützte Modelle entwickeln, um natürliche und künstliche Phänomene und Systeme kreativ zu untersuchen, zu verstehen und darüber zu kommunizieren.
 - Ethische und soziale Fragen im Zusammenhang mit Computersystemen und deren Nutzung sowie potenzielle Vorteile und Risiken identifizieren, analysieren und diskutieren.

Um diese Fähigkeiten zu beherrschen, werden eine Vielzahl an informatischen Konzepten, Prinzipien sowie Praktiken benötigt. Dem “informatics for all”-Framework folgend sind diese in zehn zentrale *Themenbereiche* gegliedert.

- **Daten und Informationen (DUI).** Verstehen, wie Daten gesammelt, organisiert, analysiert und verwendet werden, um Informationen über reale Artefakte und Szenarien zu modellieren, darzustellen und zu visualisieren.
Funktionsweisen und Einsatzmöglichkeiten von künstlicher Intelligenz nachvollziehen und bewerten.
- **Algorithmen und Datenstrukturen, Programmierung (ADP).** Bewerten, spezifizieren, entwickeln und verstehen von Algorithmen; Programmiersprachen nutzen, um sich durch die Entwicklung, das Testen und das Debuggen digitaler Artefakte rechnergestützt auszudrücken; und verstehen, was eine Programmiersprache ist.
- **Computersysteme (CMP).** Verstehen, was ein (virtuelles) Computersystem ist, wie seine Bestandteile gemeinsam als Ganzes funktionieren, und seine Grenzen kennen. Kenntnisse von Einplatinenrechnern inklusive Aktoren und Sensoren bis hin zu virtuellen Systemen.
- **Netzwerke und Kommunikation (NWK).** Verstehen, wie Netzwerke den Austausch digitaler Artefakte auf vielfältige Art und Weise ermöglichen und Kommunikationsverhalten der Gesellschaft beeinflussen.
- **Mensch-Maschine Interaktion (MMI).** Bewerten, spezifizieren, entwickeln und verstehen der Interaktion zwischen Menschen und digitalen Artefakten.
- **Design und Entwicklung (DUE).** Artefakte planen und erstellen, unter Berücksichtigung der Standpunkte von Stakeholdern und durch kritische Bewertung von Alternativen und deren Ergebnissen.
- **Digitale Kreativität (DKR).** Digitale Werkzeuge erkunden und verwenden, um diverse Artefakte zu entwickeln und zu pflegen, dabei auch eine Vielfalt von Medien nutzen.
- **Modellierung und Simulation (MUS).** Bewerten, modifizieren, entwerfen, entwickeln und verstehen von Modellen und Simulationen natürlicher und künstlicher Phänomene sowie deren Weiterentwicklung.
- **Privatsphäre, Sicherheit und Schutz (PSS).** Risiken beim Einsatz digitaler Technologien verstehen und wie man Individuen und Systeme schützt.
- **Verantwortung und Selbstbestimmung (VUS).** Kritisch und konstruktiv konkrete Computerartefakte sowie fortgeschrittene und potenziell kontroverse Techniken und Anwendungen der Informatik analysieren, insbesondere aus ethischer und sozialer Perspektive.

Sämtliche Kompetenzen werden mit den Konzepten des Dagstuhl-Dreiecks verbunden. Das Dagstuhl-Dreieck betont dabei die Balance zwischen drei wesentlichen Perspektiven:

- **Informatik als Wissenschaft:** Hier wird die Informatik als theoretische Disziplin verstanden, die auf wissenschaftlichen Grundlagen basiert und sich mit abstrakten Modellen und Konzepten wie Algorithmen, Datenstrukturen, Komplexitätstheorie und Berechenbarkeit befasst.
- **Informatik als Ingenieurdisziplin:** In diesem Aspekt geht es um die praktische Anwendung von Informatikkenntnissen, um echte technische Probleme zu lösen. Hierbei stehen Softwareentwicklung, Systemdesign, Hardware-Integration und andere ingenieurwissenschaftliche Praktiken im Vordergrund.
- **Informatik im Kontext:** Der dritte Aspekt des Dreiecks betrachtet die Informatik in ihrem gesellschaftlichen, wirtschaftlichen und kulturellen Kontext. Dazu gehören Fragen der Ethik, der sozialen Auswirkungen von Informatiksystemen, der Datenschutz und der Informatikanwendungen in unterschiedlichen Feldern wie Bildung, Politik und Wirtschaft.

Mit der Betrachtung dieser Perspektiven wird der Anschluss an die Inhalte der Digitalen Grundbildung geschaffen sowie ebenso eine vertiefende Allgemeinbildung erreicht. Das Dagstuhl-Dreieck verbindet Theorie und Praxis sowie die kontextbezogene Relevanz der Informatik und im Lehrplan werden sämtliche Themenbereiche den Dimensionen des Dagstuhl-Dreiecks folgend zugeordnet:

- T: Technologische Perspektive
- A: Anwendungsbezogene Perspektive
- G: Gesellschaftliche Perspektive

Die Struktur des Lehrplans orientiert sich nun immer an folgendem Muster. Zuerst sind die jeweiligen Themenbereiche genannt, danach folgen Hinweise zu Anwendungsbereichen sowie die jeweiligen Kompetenzen, ergänzt um die Dimensionen des Dagstuhl-Dreiecks.

5. Klasse (1. und 2. Semester):

Themenbereich: Daten und Informationen (DUI)

Anwendungsbereich: Data Science

Die Schülerinnen und Schüler können

- anhand von Beispielen erklären, wie Daten (automatisiert) erfasst, gespeichert, strukturiert, bereitgestellt und analysiert werden. (T, A)
- anhand von Beispielen schlussfolgern, welche Rolle Daten als Werkzeuge in der Gesellschaft zukommt, wie und von wem sie erzeugt und eingesetzt werden und was deren Absichten sind. (G)

Anwendungsbereich: Machine Learning

Die Schülerinnen und Schüler können

- Unterschiede und Intentionen verschiedener Machine Learning Trainingsmodelle (wie überwachtes, unbeaufsichtigtes und bestärkendes maschinelles Lernen) benennen. (T, G)
- Modellierung und Analyse von Daten durch Maschinelles Lernen nachvollziehen und ethische Implikationen erklären. (T, G)

Themenbereich: Algorithmen und Datenstrukturen, Programmierung (ADP)

Anwendungsbereich(e): Standardalgorithmen und Strukturen (objektbasierte Programmierung in textbasierten Programmen)

Die Schülerinnen und Schüler können

- kurze Programme im Team planen und in einer textbasierten Programmiersprache (ggf. auf einem Einplatinenrechner) umsetzen. (T, A)
- einfache textbasierte Algorithmen interpretieren und durch deren Kombination und Verallgemeinerung komplexere Lösungen implementieren. (T, A)
- erklären, warum Bedürfnisse von Nutzerinnen und Nutzern beim Entwurf von Programmen maßgeblich sind. (G)

Themenbereich: Computersysteme (CMP)

Anwendungsbereich: Betriebssysteme (Shell, File-Subsystem/Dateimanagement) im Besonderen von Einplatinenrechnern

Die Schülerinnen und Schüler können

- bei einem Betriebssystem Systemkonfigurationen vornehmen und wichtige Funktionen nutzen. (T, A)
- Einplatinenrechner anhand einer Reihe von Geräten konfigurieren und lösungsorientiert einsetzen. (T, A)
- die zentrale Rolle von Computersystemen und deren Einfluss auf Gesellschaft und Umwelt tiefergehend erfassen und erklären. (G)

Themenbereich: Netzwerke und Kommunikation (NWK)

Anwendungsbereich: Netzwerkkomponenten (u.a. Geräte in ein Netzwerk einbinden, einfache Fehlerquellen identifizieren/beheben)

Die Schülerinnen und Schüler können

- den Aufbau und die Funktionsweise von Netzwerken (WLAN, LAN, WAN, etc.) und deren Komponenten erklären und modellieren. (T, A)

-
- über offene und geschlossene Protokolle reflektieren und Vor- und Nachteile für die Gesellschaft aufzählen. (G, A)

Themenbereich: Mensch-Maschine Interaktion (MMI)

Anwendungsbereich: Interaktionsmöglichkeiten für Mensch-Maschine

Die Schülerinnen und Schüler können

- eine Reihe von Möglichkeiten, wie Menschen mit Computersystemen interagieren, vergleichen und beschreiben. (A, T)
- Kriterien zur Förderung der Gesundheit von Benutzerinnen und Benutzern sowie zur Vermeidung von Risiken für die physische und psychische Gesundheit in der Mensch-Maschine Interaktion benennen. (G, T)
- unterschiedliche Bedürfnisse von Rezipientinnen und Rezipienten erkennen und insbesondere im Hinblick auf Fragen der Inklusion, Gendersensibilität und Vielfalt erklären. (A, G)

Themenbereich: Design und Entwicklung (DUE)

Anwendungsbereich: Unterschiedliche Software-Entwicklungsprozesse

Die Schülerinnen und Schüler können

- die wesentlichen Phasen von dokumentengetriebenen sowie iterativen/agilen Prozess- bzw. Vorgehensmodellen beschreiben. (T, A)
- Vorteile und Grenzen dokumentengetriebener sowie agiler Prozesse (zB Einfluss der Auftraggeber) erklären. (G)

Themenbereich: Digitale Kreativität (DKR)

Anwendungsbereich: Entwurf digitaler Artefakte (Bildbearbeitung mittels freier Software)

Die Schülerinnen und Schüler können

- Gestaltungsprinzipien benennen, unterscheiden und iterativ umsetzen. (T, A)
- Lizenzmodelle bezüglich der Nutzung und Verbreitung digitaler Artefakte einsetzen. (A, G)

Themenbereich: Modellierung und Simulation (MUS)

Anwendungsbereich: 3D-Modelle in der realen Welt

Die Schülerinnen und Schüler können

- den Aufbau von grafischen Computermodellen erklären. (T)
- erklären, wie 3D-Modelle neue Möglichkeiten für Design, Architektur, Medizin, Unterhaltung und Bildung eröffnen und Innovationen in verschiedenen Branchen fördern. (G)

Themenbereich: Privatsphäre, Sicherheit und Schutz (PSS)

Anwendungsbereich: Authentifizierungs-Strategien (Benutzername und Passwort, Zwei-Faktor-Authentifizierung (2FA), Multi-Faktor-Authentifizierung (MFA), biometrisch, token-basiert, Zertifikat-basiert)

Die Schülerinnen und Schüler können

- verschiedene Authentifizierungsmaßnahmen für ihre eigenen Endgeräte nutzen und deren Möglichkeiten und Grenzen erklären. (T, A)
- Vor- sowie Nachteile von informatischen Authentifizierungsstrategien beschreiben. (G)

Themenbereich: Verantwortung und Selbstbestimmung (VUS)

Anwendungsbereiche: Nachhaltige (Weiter-) Entwicklung, Energieverbrauch

Die Schülerinnen und Schüler können

- einige Auswirkungen (Nachhaltigkeit, Energieverbrauch) neuer Technologien aufzeigen und geeignete Maßnahmen zu einer positiven Entwicklung setzen. (T, A, G)

6. Klasse:

3. Semester – Kompetenzmodul 3:

Themenbereich: Daten und Informationen (DUI)

Anwendungsgebiet: Datenbanken

Die Schülerinnen und Schüler können

- den Unterschied von strukturierten und unstrukturierten Daten an Beispielen erklären. (T, A)
- einfache Datenbankabfragen (SQL) durchführen. (T, A)
- Verknüpfungsmethoden von Daten und Datensätzen in relationalen Datenbanken (SQL) anwenden (Joins). (T)

-
- Daten und Analyseergebnisse interpretieren und über deren Wahrheitsgehalt und Verwendung diskutieren. (G, A)

Themenbereich: Mensch-Maschine Interaktion (MMI)

Anwendungsgebiet: Mobile Interaktionskonzepte

Die Schülerinnen und Schüler können

- mobile Interaktionskonzepte (wie Sprache, Touch, Gesten) und deren Bedingungen beschreiben und Inklusionsaspekte veranschaulichen. (A, G)
- grundlegende technische Aspekte mobiler Interaktionskonzepte (zB Wearables), einschließlich der Funktionsweise von Sensoren, der Datenübertragung und der Energieversorgung veranschaulichen. (T, A)
- nachvollziehen, welche Daten mobiler Interaktionskonzepte (zB Wearables) erfasst werden, wie diese Daten verwendet werden können, und beurteilen, welche persönlichen Auswirkungen Tracking im Rahmen von Omnimetrie und Selbstoptimierung haben kann. (T, G)

Themenbereich: Design und Entwicklung (DUE)

Anwendungsgebiete: Anforderungen und grafischer Modelle

Die Schülerinnen und Schüler können

- funktionale und nicht funktionale Anforderungen erkennen und einfache grafische Modelle nutzen. (T)
- digitale Artefakte im Hinblick auf Fragen der Nachhaltigkeit und Vielfalt diskutieren. (G)

Themenbereich: Digitale Kreativität (DKR)

Anwendungsgebiet: Multimediale Artefakte mit KI Unterstützung

Die Schülerinnen und Schüler können

- multimediale Artefakte mit KI-Unterstützung erzeugen und die dahinterliegenden Prinzipien ansatzweise erklären. (T, A)
- Möglichkeiten der Manipulation multimedialer Artefakte (zB Deep Fakes) benennen und deren mögliche Auswirkungen nachvollziehen. (G)

Themenbereich: Modellierung und Simulation (MUS)

Anwendungsgebiet: 3D-Modellierung

Die Schülerinnen und Schüler können

- Teile der realen Welt abstrahieren, in ein grafisches Modell überführen sowie für die praktische Umsetzung (zB Druck) vorbereiten. (T, A)
- Möglichkeiten und Grenzen des 3D-Drucks in der realen Welt aufzeigen. (G)

4. Semester – Kompetenzmodul 4:

Themenbereich: Daten und Informationen (DUI)

Anwendungsgebiet: Abläufe maschinellen Lernens

Die Schülerinnen und Schüler können

- den Prozess von Machine Learning am Beispiel von supervised learning erklären und Beispiele von overfitting und underfitting von Daten im Zusammenhang mit Biases und Fairness aufzeigen. (T, G)

Themenbereich: Algorithmen und Datenstrukturen, Programmierung (ADP)

Anwendungsgebiet: Agile Softwareentwicklungsprojekte (auf einem Einplatinenrechner)

Die Schülerinnen und Schüler können

- ein agiles Softwareprojekt (schuljahrübergreifend) planen und im Team umsetzen. (T, A)
- eine Reihe einfacher Standard-Algorithmen (zB Sortieren oder Suchen) erklären und implementieren. (T, A)
- Einsatzmöglichkeiten und gesellschaftliche Implikationen von Einplatinenrechnern im täglichen Leben zusammenfassen. (G)

Themenbereich: Computersysteme (CMP)

Anwendungsgebiete: Einplatinenrechner und Sensoren/Aktoren

Die Schülerinnen und Schüler können

-
- eigene (möglicherweise fächerverbindende) Projekte mit einem Einplatinenrechner planen, spezifizieren und mit vorhandener Hardware oder Simulatoren implementieren. (T, A)
 - auf theoretischer Ebene Computersysteme konzipieren, welche die Benachteiligung & Diskriminierung minimieren und Chancengleichheit maximieren. (G)

Themenbereich: Netzwerke und Kommunikation (NWK)

Anwendungsgebiet: Netzwerk-Parameter (Internet Protocol Suite, IP-Adresse, MAC-Adresse, Ping)

Die Schülerinnen und Schüler können

- den Weg von Netzwerk-Paketen nachvollziehen und einfache Fehler in Netzwerken identifizieren. (A)
- den Aufbau, die Funktionsweise sowie Vor- und Nachteile von dezentralen Netzwerken erklären. (T, G)

Themenbereich: Privatsphäre, Sicherheit und Schutz (PSS)

Anwendungsgebiet: Anwendung von Sicherheitsstrategien (Firewall/Ports, Virens Scanner)

Die Schülerinnen und Schüler können

- die Sicherheit von Geräten oder Services prüfen und bei Bedarf die Sicherheitsstrategien adaptieren. (A)
- Kompromisse bei der Auswahl und Umsetzung von Sicherheitsempfehlungen erklären. (G)

Themenbereich: Verantwortung und Selbstbestimmung (VUS)

Anwendungsgebiete: Ethische Aspekte, Inklusionsaspekte

Die Schülerinnen und Schüler können

- ethische Grundsätze sowie Inklusionsaspekte bei der Nutzung und für die Nutzung von digitalen Werkzeugen identifizieren und beschreiben. (A, G)

7. Klasse:

5. Semester – Kompetenzmodul 5:

Themenbereich: Daten und Informationen (DUI)

Anwendungsgebiet: Datenqualität

Die Schülerinnen und Schüler können

- Konzepte der Datenqualität (Genauigkeit, Vollständigkeit, Konsistenz, Aktualität und Vertraulichkeit) beschreiben. (T, A)
- rechtliche und ethische Aspekte von Datenqualität benennen und an Beispielen von Datenschutzgesetzen und Datensicherheit erklären. (A, G)

Themenbereich: Computersysteme (CMP)

Anwendungsgebiet: Proprietäre und Open Source Betriebssysteme

Die Schülerinnen und Schüler können

- Betriebssysteme installieren und Datenträger entsprechend vorbereiten (inkl. Partitionierung) bzw. nutzen. (T, A)
- eigene (möglicherweise fächerverbindende) Projekte planen, spezifizieren und bei vorhandener Hardware auch implementieren. (A)
- Zukunftsperspektiven und Gefahren von Computersystemen für die Menschheit benennen und erklären. (G)

Themenbereich: Netzwerke und Kommunikation (NWK)

Anwendungsgebiet: Multimedia-Kommunikation

Die Schülerinnen und Schüler können

- Technologien zur Übertragung von Multimedia-Inhalten über Netzwerke, einschließlich Videokompression, Streaming, Video-on-Demand und Videoconferencing benennen. (T)
- Möglichkeiten der Multimedia-Kommunikation erklären, um den Digital Divide zu verringern. (G)

Themenbereich: Mensch-Maschine Interaktion (MMI)

Anwendungsgebiet: Audio-Visuelle Schnittstellen und Interaktionen

Die Schülerinnen und Schüler können

-
- Konzepte und Ziele von Augmented Reality und Virtual Reality unterscheiden und Interaktionsmöglichkeiten (wie Navigation, Selektion und Manipulation) aufzeigen. (T, A)
 - den Einfluss von Audio-Visuellen Interaktionen auf die Arbeitswelt und Gesellschaft an Beispielen mit Blick auf zukünftige Entwicklungen reflektieren. (A, G)
 - Schnittstellen in Bezug auf unterschiedliche Bedürfnisse von Rezipientinnen und Rezipienten (besonders im Hinblick auf Fragen der Inklusion, Gendersensibilität und Vielfalt) beurteilen. (A, G)

Themenbereich: Digitale Kreativität (DKR)

Anwendungsgebiet: Webdesign (Design einfacher statischer und dynamischer Webseiten, ggf. mit CMS-Systemen)

Die Schülerinnen und Schüler können

- clientbezogene und serverbezogene Komponenten von dynamischen Websites unterscheiden. (T)
- den Aufbau einfacher Webseiten nachvollziehen und diese adaptieren. (T, A)
- Sicherheitsrisiken für die Manipulation von Webseiten benennen (Hacking) und Lösungsstrategien erarbeiten. (T, G)

Themenbereich: Privatsphäre, Sicherheit und Schutz (PSS)

Anwendungsgebiet: Open Source Intelligence

Die Schülerinnen und Schüler können

- Möglichkeiten der Informationsbeschaffung und -analyse durch die Auswertung öffentlich zugänglicher Quellen wie Websites, sozialer Medien, Foren, Nachrichtenartikel, Blogs und anderer frei verfügbarer Informationen erklären. (T, A)
- die Auswirkungen von OSINT (Open Source Intelligence) Angriffen auf Individuen im Hinblick auf ihre Privatsphäre und Sicherheit nachvollziehen. (G)

Themenbereich: Verantwortung und Selbstbestimmung (VUS)

Anwendungsgebiet: Informatik und Gesellschaft (u.a. eGovernment)

Die Schülerinnen und Schüler können

- Partizipationsmöglichkeiten an der digitalen Gesellschaft an Hand des eGovernment erklären und an Beispielen nachvollziehen. (T, A, G)

6. Semester – Kompetenzmodul 6:

Themenbereich: Daten und Informationen (DUI)

Anwendungsgebiet: Neuronale Netze

Die Schülerinnen und Schüler können

- die Architektur neuronaler Netzwerke veranschaulichen. (T)
- grundlegende Bausteine eines Neuronales Netzes, einschließlich Neuronen, Gewichtungen und Aktivierungsfunktionen an Beispielen aufzeigen. (T)
- Anwendungsgebiete von Neuronalen Netzen aufzeigen und gesellschaftliche und ethische Implikationen reflektieren. (G)

Themenbereich: Algorithmen und Datenstrukturen, Programmierung (ADP)

Anwendungsgebiet: Unterschiedliche Software-Entwicklungsmethoden (Pair Programming, Distributed Programming, Solo Programming, KI-unterstützte Softwareentwicklung)

Die Schülerinnen und Schüler können

- unterschiedliche Software-Entwicklungsmethoden vergleichen und in einem agilen Software-Projekt Pair Programming umsetzen. (T, A)
- Programmcode (u.a. von KI-Systemen) analysieren und Vor- und Nachteile (in Bezug auf Verständlichkeit und Effizienz) bewerten. (T, A)
- Vorteile sowie Herausforderungen von kollaborativem Arbeiten zusammenfassen. (G)

Themenbereich: Design und Entwicklung (DUE)

Anwendungsgebiet: Prototyping

Die Schülerinnen und Schüler können

- Techniken einsetzen, um digitale Artefakte in Form von Prototypen zu entwickeln. (T, A)
- beurteilen, welchen Einfluss Prototyping auf den Designprozess hat. (A)

Themenbereich: Modellierung und Simulation (MUS)

Anwendungsgebiet: Simulation einfacher Systeme

Die Schülerinnen und Schüler können

- Beispiele zellulärer Automaten erkennen und einen einfachen Automaten in einer Programmiersprache umsetzen. (T, A)
- veranschaulichen, wie zelluläre Automaten im gesellschaftlichen Kontext genutzt werden können, um komplexe Phänomene zu untersuchen, Verhaltensweisen zu analysieren und Potenziale für Veränderungen und Verbesserungen zu identifizieren. (A, G)

8. Klasse:

7. Semester – Kompetenzmodul 7:

Themenbereich: Daten und Informationen (DUI)

Anwendungsgebiete: Data Management Systems, Big Data

Die Schülerinnen und Schüler können

- Daten aus Datenbanken und anderen Quellen abfragen bzw. integrieren und verknüpfen, um eine umfassendere Sicht auf die Daten zu erhalten. (T, A)
- die wichtigsten Eigenschaften von Big Data (Volumen, Vielfalt und Geschwindigkeit) erklären und deren gesellschaftliche Auswirkungen beschreiben. (T, G)

Anwendungsgebiet: Generative Neuronale Netze

Die Schülerinnen und Schüler können

- Unterschiede von KI Systemen (wie diskriminative und generative KI) benennen und Anwendungsmöglichkeiten zuordnen. (T)
- ethische und gesellschaftliche Probleme im Training und in der Nutzung von KI Systemen (wie generativer und diskriminativer KI) benennen und Regulierungsmaßnahmen aufzeigen. (G)

Themenbereich: Algorithmen und Datenstrukturen, Programmierung (ADP)

Anwendungsgebiet: Qualitätssicherung von Programmen

Die Schülerinnen und Schüler können

- Algorithmen und Datenstrukturen nach quantitativen und qualitativen Maßstäben (zB Umfang, Kopplung, Effizienz und Korrektheit) beurteilen. (T, A)
- Programme anhand einer Reihe von Testfällen systematisch testen und überarbeiten. (T, A)
- beurteilen, ob Programme auf die Bedürfnisse von Nutzerinnen und Nutzern ausreichend eingehen. (G)

Themenbereich: Computersysteme (CMP)

Anwendungsgebiete: Cloud/Virtuelle Betriebssysteme, Physical Computing

Die Schülerinnen und Schüler können

- Konzepte virtualisierter Betriebssysteme mit lokalen Betriebssystemen vergleichen. (T, A)
- bei einem virtuellen Betriebssystem Systemkonfigurationen vornehmen und wichtige Funktionen erklären. (T, A)
- Funktionsweisen und Einsatzgebiete von Physical Computing erklären. (T, A)

Themenbereich: Netzwerke und Kommunikation (NWK)

Anwendungsgebiet: Blockchain-Technologie

Die Schülerinnen und Schüler können

- die Eigenschaften der Blockchain-Technologie erklären und wichtige Anwendungen zusammenfassen. (T, A)
- Nachhaltigkeitsaspekte und gesellschaftliche Implikationen von Blockchain-Technologien aufzeigen. (G)

Themenbereich: Mensch-Maschine Interaktion (MMI)

Anwendungsgebiete: Data Stories und Interaktionssysteme

Die Schülerinnen und Schüler können

- explorative und erklärende Data Stories unterscheiden. (T)
- Möglichkeiten interaktiver Datenvisualisierung mit Storytelling oder Gamification von Data Stories an Beispielen aufzeigen. (A)
- die Nutzung virtueller (KI-generierter) Assistenten für Datenanalyse beschreiben, Einsatzmöglichkeiten aufzeigen und gesellschaftliche Implikationen beurteilen. (A, G)

Themenbereich: Design und Entwicklung (DUE)**Anwendungsgebiete: Human-Centered Design**

Die Schülerinnen und Schüler können

- die Prinzipien, die dem Human-Centered Design zugrunde liegen, erklären. (T)
- beurteilen, wie gut Human-Centered Design aus Sicht von Designerinnen und Designern bzw. von Benutzerinnen und Benutzern funktioniert. (G)

Themenbereich: Digitale Kreativität (DKR)**Anwendungsgebiete: Multimediales Webdesign**

Die Schülerinnen und Schüler können

- eigene digitale Artefakte in Webseiten einbauen. (T, A)
- Webseiten (zumindest theoretisch) responsiv umsetzen. (T)
- Aspekte der Nachhaltigkeit und Umweltverträglichkeit in Bezug auf multimediale Webangebote (wie Streamingdienste) benennen. (G)

Themenbereich: Modellierung und Simulation (MUS)**Anwendungsgebiet: Simulation von komplexen Systemen**

Die Schülerinnen und Schüler können

- Natürliche und künstliche Phänomene in Computermodellen umsetzen, diese ausführen und deren Auswirkungen (Grenzen) vorhersagen und überprüfen. (A, G)
- die Chancen und Gefahren fortgeschrittener Simulationen (zB auf der Grundlage von virtueller oder erweiterter Realität) veranschaulichen und diskutieren. (G)

Themenbereich: Privatsphäre, Sicherheit und Schutz (PSS)**Anwendungsgebiet: Social Engineering**

Die Schülerinnen und Schüler können

- Beispiele von Social Engineering-Angriffen benennen und Gegenmaßnahmen empfehlen. (T, A)
- Herausforderungen identifizieren, die sich durch verschiedene Rechtssysteme und Kulturen bei der Bereitstellung von Richtlinien für die Computernutzung und das Verhalten ergeben. (G)

Themenbereich: Verantwortung und Selbstbestimmung (VUS)**Anwendungsgebiet: Digitalität (mit)gestalten**

Die Schülerinnen und Schüler können

- beispielhafte Kriterien empfehlen (aufzeigen), wie digitale Artefakte im Diskurs unter den Aspekten Inklusion, Gendersensibilität und Diskriminierungsfreiheit gestaltet werden können. (T, G)

8. Semester– Kompetenzmodul 8:**Sicherung der Nachhaltigkeit**

- Wiederholen, Vertiefen von Fähigkeiten und Vernetzen von Inhalten, um einen umfassenden Überblick über die Zusammenhänge unterschiedlicher informatischer Themenbereiche zu gewinnen.

WAHLPFLICHTFACH INFORMATIK

Bildungs- und Lehraufgabe (für alle Klassen):

Das Wahlpflichtfach Informatik soll die Schülerinnen und Schüler zu einer erweiterten und vertieften informatischen Bildung führen. Diese umfasst sowohl sachliche und fachsystematische als auch methodische und soziale Komponenten.

Unter Berücksichtigung der fachwissenschaftlichen Systematik soll der Unterricht zu einem zielorientierten, konstruktiven Problemlösen mit informatischen Werkzeugen und Methoden führen. Durch eine präzise, strukturierte und vollständige Beschreibung sowohl von Problemstellungen als auch von Abläufen, sowie durch die Modularisierung und Dekomposition komplexer Aufgaben soll die Informatik zur Schulung abstrakten Denkens beitragen. Schließlich soll das Wahlpflichtfach wesentlich zu einem vertieften technischen und soziokulturellen Verständnis unserer digital geprägten Welt führen und interessierte Schülerinnen und Schüler auf ein einschlägiges Studium vorbereiten.

Didaktische Grundsätze (für alle Klassen):

Die didaktischen Grundsätze für das Pflichtfach Informatik dienen auch im Wahlpflichtfach als Leitlinien. Die Hinführung zu modul- und projektartigen, vernetzten Arbeitsformen ist Grundlage für das Fach selbst und auch für fächerverbindende Kooperationen. Ein Schwerpunkt des Informatikunterrichts soll in der formalen Modellierung von Sachverhalten und in der Problemlösung liegen, welche aus Analyse, Beschreibung in verschiedenen Darstellungsformen, algorithmischer Lösung, Implementation, Überprüfung und Interpretation sowohl vom informatischen als auch sachlichen Standpunkt bestehen. Die unterschiedliche Kenntnislage der Schülerinnen und Schüler erfordert in einzelnen Fachbereichen eine differenzierte und individualisierte Unterrichtsgestaltung. Die Auswahl der Themen soll zeitgemäß und gendersensibel sein und möglichst die Interessen der Schülerinnen und Schüler berücksichtigen.

Bildungs- und Lehraufgabe, Lehrstoff:

Die Themenbereiche folgen wie auch beim Pflichtfach Informatik den Empfehlungen des informatics4all Referenzframeworks, sind spiralförmig aufgebaut und umfassen 10 zentrale Themenbereiche.

- Daten und Informationen (DUI).
- Algorithmen und Datenstrukturen, Programmierung (ADP).
- Computersysteme (CMP).
- Netzwerke und Kommunikation (NWK).
- Mensch-Maschine Interaktion (MMI).
- Design und Entwicklung (DUE).
- Digitale Kreativität (DKR).
- Modellierung und Simulation (MUS).
- Privatsphäre, Sicherheit und Schutz (PSS).
- Verantwortung und Selbstbestimmung (VUS).

Die Kompetenzbeschreibungen sollen wiederum den jeweils angegebenen Perspektiven des Dagstuhl-Dreiecks folgen:

- T: Technologische Perspektive
- A: Anwendungsbezogene Perspektive
- G: Gesellschaftliche Perspektive

6. Klasse:

3. Semester – Kompetenzmodul 3:

Themenbereich: Daten und Informationen (DUI)

Anwendungsbereich: Datenbankkonzepte

Die Schülerinnen und Schüler können

- die verschiedenen Ansätze zur Datenbankverwaltung von SQL (Structured Query Language) und NoSQL (Not only SQL) in einigen wichtigen Aspekten unterscheiden. (T)
- Auswirkungen schlechter Datenqualität an Beispielen veranschaulichen und bewerten. (A, G)

Themenbereich: Mensch-Maschine Interaktion (MMI)

Anwendungsbereich: Etablierte Interaktionsstile

Die Schülerinnen und Schüler können

- etablierte Interaktionsstile unterscheiden und an Anwendungsbeispielen überprüfen. (T)

-
- historische Entwicklungen und zukünftige Tendenzen etablierter Interaktionsstile aufzeigen. (T, G)
 - unterschiedliche Bedürfnisse von Rezipientinnen und Rezipienten im Hinblick auf Fragen der Inklusion, Gendersensibilität und Vielfalt in der Anwendung etablierter Interaktionsstile berücksichtigen. (A, G)

Themenbereich: Design und Entwicklung (DUE)

Anwendungsbereiche: Versionsverwaltungssoftware, Agiles Entwicklungsmodell / Open Source Code-sharing-Plattform

Die Schülerinnen und Schüler können

- Versionsverwaltungssoftware sowie Issue-Tracker nutzen, um Projekte im Team umzusetzen und Fehler bzw. deren Behebung zu dokumentieren. (T, A)
- Vorteile und Risiken von Codesharing-Plattformen einschätzen. (G)

Themenbereich: Digitale Kreativität (DKR)

Anwendungsbereich: Multimediale Artefakte (Gestaltung von Objekten oder Werken, die verschiedene Medienformen kombinieren)

Die Schülerinnen und Schüler können

- verschiedene Medienformen kombinieren, um Informationen zu vermitteln, Geschichten zu erzählen oder eine ästhetische Erfahrung zu schaffen. (T, A)
- Aspekte der Barrierefreiheit im Zusammenhang mit multimedialen Artefakten beurteilen. (A, G)

Themenbereich: Modellierung und Simulation (MUS)

Anwendungsbereich: Maker-Anwendungen

Die Schülerinnen und Schüler können

- Software, Sensoren, Aktoren und mechanische Elemente kombinieren und eine Maker-Anwendung praktisch umsetzen. (T, A)
- erklären wie Maker-Projekte Innovationen unterstützen können. (G)

4.Semester – Kompetenzmodul 4:

Themenbereich: Daten und Informationen (DUI)

Anwendungsbereich: Machine Learning: Trainieren

Die Schülerinnen und Schüler können

- experimentell Maschinen in vorgegebenen Umgebungen trainieren. (T)

Themenbereich: Algorithmen und Datenstrukturen, Programmierung (ADP)

Anwendungsbereich: Automatisierungs-/Robotik-Projekt (mittels Sensoren und Aktoren)

Die Schülerinnen und Schüler können

- unter Anleitung in einem agilen Software-Projekt geeignete Techniken zur Projektplanung und Dokumentation anwenden. (T, A)
- Schnittstellen wie Sensoren und Aktoren eines Einplatinenrechners erklären und diese programmtechnisch nutzen. (T, A)
- Auswirkungen digitaler Systeme auf unsere Umwelt (Ressourcenbedarf, Energieverbrauch) erklären. (G)

Themenbereich: Computersysteme (CMP)

Anwendungsbereiche: Maker-Projekte mit Einplatinenrechnern, Internet of Things

Die Schülerinnen und Schüler können

- geeignete Zugriffsrechte auf Daten und Dateien, gegebenenfalls auch über den Kommandozeileninterpreter, erklären. (T, A)
- eigene (möglicherweise fächerverbindende) Projekte mit einem Einplatinenrechner planen, spezifizieren und bei vorhandener Hardware auch implementieren. (A)
- Funktionsweisen und Einsatzgebiete von Internet of Things-Devices erklären. (T, A)

Themenbereich: Netzwerke und Kommunikation (NWK)

Anwendungsbereich: Vertiefung Netzwerk-Parameter (wie zB Netzwerk-Typ, Nameserver, Timeserver)

Die Schülerinnen und Schüler können

-
- einfache Netzwerke planen und entweder mittels vorhandener Hardware oder Netzwerksimulatoren aufbauen. (T, A)
 - die Verwundbarkeit von Netzwerk-Architekturen und deren gesellschaftlichen Implikationen erklären. (G)

Themenbereich: Privatsphäre, Sicherheit und Schutz (PSS)

Anwendungsbereiche: Verschlüsselungsstrategien, deren Codierung sowie Sicherung von Schlüsseln

Die Schülerinnen und Schüler können

- unterschiedliche Verschlüsselungsverfahren beschreiben und unterscheiden. (T)
- gesellschaftliche Aspekte bei der Verwaltung von zentralen bzw. dezentralen Schlüsseln erklären. (G)

Themenbereich: Verantwortung und Selbstbestimmung (VUS)

Anwendungsbereiche: Open Data, Datentransparenz

Die Schülerinnen und Schüler können

- Open Data sowie den gesellschaftlichen Nutzen von Open Data erklären und mit offenen Daten arbeiten. (T, A, G)

7. Klasse:

5. Semester – Kompetenzmodul 5:

Themenbereich: Daten und Informationen (DUI)

Anwendungsbereich: Data Quality Management

Die Schülerinnen und Schüler können

- Prozesse des Datenqualitätsmanagements erklären. (T, A)
- Datenexploration mit Hilfe der datenbasierten Erstellung und Evaluation von Entscheidungsbäumen durchführen. (T, A)
- Datenfehler erkennen und Lösungsmöglichkeiten zu deren Behebung umsetzen. (T, A)

Themenbereich: Computersysteme (CMP)

Anwendungsbereiche: Problemlösungsstrategien mit Betriebssystemen, Prozesse/Rechte

Die Schülerinnen und Schüler können

- bei einem Betriebssystem Systemkonfigurationen vornehmen und wichtige Funktionen erklären. (T, A)
- die Zweck einer Schichtenarchitektur erklären und von Bibliotheken in Betriebssystemen nutzen. (T, A)
- die Unterschiede von proprietären und Open Source Betriebssystemen benennen und erklären. (T)

Themenbereich: Netzwerke und Kommunikation (NWK)

Anwendungsbereich: Netzwerkkonfiguration

Die Schülerinnen und Schüler können

- die Architektur eines Netzwerkes nachvollziehen und Fehler in Netzwerken identifizieren. (A)
- Aussagen zur Verwundbarkeit von digitalen Infrastrukturen treffen und Gegenmaßnahmen empfehlen. (A, G)

Themenbereich: Mensch-Maschine Interaktion (MMI)

Anwendungsbereich: Anforderungsanalysen interaktiver Systeme

Die Schülerinnen und Schüler können

- Anforderungen von Benutzerinnen und Benutzern erheben und reflektieren. (A, G)
- mit Wireframes als schematische Darstellungen den grundlegenden Aufbau von Interaktionschnittstellen skizzieren und Storyboards für Prototypen konzipieren. (T, A)

Themenbereich: Digitale Kreativität (DKR)

Anwendungsbereich: Webdesign Vertiefung (Design dynamischer Webseiten, ggf. mit Content Management Systemen)

Die Schülerinnen und Schüler können

-
- Komponenten und Funktionsweisen für dynamische Webseiten benennen und exemplarisch mit CMS umsetzen sowie Unterschiede zu statischen Webseiten aufzeigen. (T, A)
 - einfache Webseiten gestalten. (T, A)
 - Einstellungen und Funktionen von Zugriffsrechten auf dynamische Webseiten aufzeigen und Sicherheitshinweise für die Administration benennen. (A, G)

Themenbereich: Privatsphäre, Sicherheit und Schutz (PSS)

Anwendungsbereiche: Typen von Hackern, ethisches Hacking

Die Schülerinnen und Schüler können

- Open Source Intelligence, IT-Forensik und Reverse Engineering Techniken benennen. (T)
- veranschaulichen, wie Verstöße gegen die Sicherheit und den Datenschutz die Sicherheit aller gefährden können. (A, G)

Themenbereich: Verantwortung und Selbstbestimmung (VUS)

Anwendungsbereich: Mitgestaltungsmöglichkeiten in der Gesellschaft

Die Schülerinnen und Schüler können

- Beispiele aufzeigen, wie und unter welchen Bedingungen die Gesellschaft mehr an der Gestaltung von Innovationen teilhaben kann. (T, G)

6.Semester – Kompetenzmodul 6:

Themenbereich: Daten und Informationen (DUI)

Anwendungsbereiche: Spezielle Themengebiete und Architekturen neuronaler Netze

Die Schülerinnen und Schüler können

- Funktionen von Neuronalen Netzen in verschiedenen Bereichen, wie zB Visual Computing, Sprachverarbeitung und autonome Systeme exemplarisch umsetzen und ethische und soziale Implikationen benennen. (A, G)

Themenbereich: Algorithmen und Datenstrukturen, Programmierung (ADP)

Anwendungsbereich: Programmieren von Applikationen (zB Mini-Games oder andere Applikationen)

Die Schülerinnen und Schüler können

- eigenständig in einem agilen Software-Projekt geeignete Techniken zur Projektplanung und Dokumentation umsetzen. (T, A)
- textbasierte Algorithmen analysieren und mit geeigneten Notationsformen darstellen. (T)
- nachvollziehen, dass die meisten Applikationen in einen sozio-technischen Kontext eingebettet sind. (G)

Themenbereich: Design und Entwicklung (DUE)

Anwendungsbereich: Anforderungsanalysen

Die Schülerinnen und Schüler können

- Anforderungsanalysen durchführen und Anforderungen auf ihre Qualität und Umsetzbarkeit hin beurteilen. (T, A)
- die Auswirkungen von schlechtem Requirements Engineering erklären. (G)

Themenbereich: Modellierung und Simulation (MUS)

Anwendungsbereich: Gamedesign

Die Schülerinnen und Schüler können

- einfache Spielideen planen und umsetzen (2D und/oder 3D). (T, A)
- über den Einfluss von Spielen auf die Psyche von Individuen reflektieren. (G)

8. Klasse:

7.Semester – Kompetenzmodul 7:

Themenbereich: Daten und Informationen (DUI)

Anwendungsbereiche: Big Data, Cloud

Die Schülerinnen und Schüler können

- Konzepte von Big Data und Cloud Computing, einschließlich der Verwendung von Cloud-basierten Daten Management Systemen exemplarisch beschreiben. (T, A)

Themenbereich: Daten und Informationen (DUI)**Anwendungsbereich:** Architekturen Generativer Neuronaler Netze

- Möglichkeiten und Auswirkungen von verschiedenen Architekturen und Trainingsmethoden auf die Performance und Effektivität von generativen Modellen untersuchen. (T, A)
- Gründe von Diskriminierung/Bias in der generativen KI aufzeigen. (G)

Themenbereich: Algorithmen und Datenstrukturen, Programmierung (ADP)**Anwendungsbereich:** Refactoring von Programmen

Die Schülerinnen und Schüler können

- Programmcode nach quantitativen und qualitativen Maßstäben (zB Effizienz und Korrektheit) beurteilen und gegebenenfalls passend umformulieren. (T, A)
- erklären, welche Auswirkungen fehlerhafte Programme auf Mensch und Gesellschaft haben können. (G)

Themenbereich: Computersysteme (CMP)**Anwendungsbereich:** Cloud/Virtuelle Betriebssysteme

Die Schülerinnen und Schüler können

- virtualisierte Betriebssysteme einsetzen und verwalten. (T, A)

Themenbereich: Netzwerke und Kommunikation (NWK)**Anwendungsbereich:** Cloud Computing

Die Schülerinnen und Schüler können

- Cloud-Architekturen, Datenübertragung, Skalierbarkeit und Service Level Agreements (SLAs) beschreiben. (T)
- ökologische und gesellschaftliche Auswirkungen erklären. (G)

Themenbereich: Mensch-Maschine Interaktion (MMI)**Anwendungsbereich:** Merkmale von Data Stories in Interaktionssystemen

Die Schülerinnen und Schüler können

- Merkmale von Data Stories in Interaktionssystemen wie Interaktive Visualisierung, Benutzerzentrierung, Kontextbezogene Navigation, Echtzeitaktualisierung, Social Sharing, und Multimodale Interaktion aufzeigen. (T)
- anhand von Data Stories exemplarische Beispiele anhand der Merkmale analysieren und beurteilen. (A, G)

Themenbereich: Design und Entwicklung (DUE)**Anwendungsbereich:** User Experience Design

Die Schülerinnen und Schüler können

- verschiedene Dimensionen, die zur Entwicklung von Artefakten herangezogen werden, zusammenfassen. (T, G)

Themenbereich: Digitale Kreativität (DKR)**Anwendungsbereich:** Multimediales Webdesign

Die Schülerinnen und Schüler können

- eigene digitale Artefakte in Webseiten für den Zugriff optimieren. (T, A)
- Responsives Design exemplarisch umsetzen. (T, A)
- Aspekte der Barrierefreiheit im Zusammenhang mit multimedialen Webseiten benennen. (A, G)

Themenbereich: Modellierung und Simulation (MUS)**Anwendungsbereiche:** Virtual and Augmented Reality

Die Schülerinnen und Schüler können

- einfache Anwendungen in der virtuellen und/oder erweiterten Realität produzieren. (T, A)
- Möglichkeiten von Simulationen (zB auf der Grundlage virtueller oder erweiterter Realität) in Bereichen des täglichen Lebens aufzeigen. (G)

Themenbereich: Privatsphäre, Sicherheit und Schutz (PSS)**Anwendungsbereiche:** IT-Forensik und Reverse Engineering Techniken

Die Schülerinnen und Schüler können

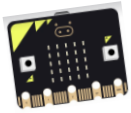
-
- im Themenfeld Internetkriminalität einfache Angriffs- und Verteidigungsaufgaben durchspielen. (T, A)
 - gesellschaftliche Bezüge im Bereich der Internetkriminalität erklären und diskutieren, einschließlich der Auswirkungen auf Rechtssysteme, Privatsphäre, Sicherheit, Ethik und Vertrauen. (A, G)

Themenbereich: Verantwortung und Selbstbestimmung (VUS)

Anwendungsbereich: Digital Divide

Die Schülerinnen und Schüler können

- aufzeigen, welche Stufen der Differenzierung des „Digital Divide“ es gibt, welche Einschränkungen damit einhergehen und welche Lösungsansätze angedacht werden können. (G)
-



13.2 Praxisprojekt 1 (Reihen-Parallelschaltung)

Projekt 1 - Reihen- und Parallelschaltung

Benötigte Komponenten (für 2 Projekte)

- Micro:bit V2 mit Breakout-Board
- Starterkit GeeekPi BBC Micro:bit V2 Ultimate Starter Kit optional¹⁹
- Steckbrett
- 2 Light-Emitting-Diode (LED) rot, grün
- 2 Widerstände mit je 270 Ohm
- Jumperkabel "male-female" und "male-male"

Benötigte Software

- TinkerCAD²⁰

Generelle Vorsichtsmaßnahmen

- Niemals den 3,3-Volt-Anschluss direkt (ohne Vorwiderstand) mit Ground verbinden
- Bei elektronischen Steckübungen oder Veränderungen immer zuerst den Micro:bit vom Strom nehmen (von Batterie oder USB-Anschluss)
- Sich selbst vor der Arbeit mit Elektronik elektrostatisch entladen (Berühren von metallischen Teilen wie Heizkörpern, Metalltischen etc.)

Projektbeschreibung

Reihenschaltung

¹⁹ Siehe online unter https://www.amazon.de/GeeekPi-BBC-Micro-Ultimate-microbit/dp/B0BKXLM97?ref=ast_sto_dp (20.03.2024)

²⁰ Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.03.2024)

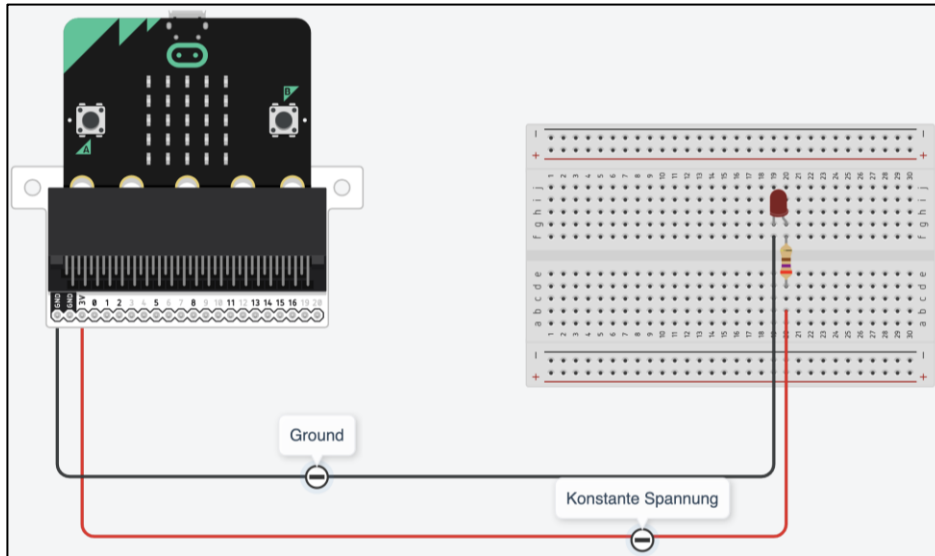
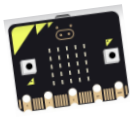
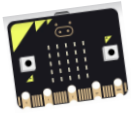


Abbildung 1: Reihenschaltung von Vorwiderstand und LED

Die Abbildung 1 zeigt eine reine Reihenschaltung. Ein Vorwiderstand von 270 Ohm und eine LED sind in Reihe (hintereinander) geschaltet und der Strom fließt durch beide Komponenten zum Ground. Die Gesamtspannung von 3,3 Volt wird zunächst durch den Vorwiderstand auf 2 Volt reduziert und gelangt dann als Strom durch die LED, die als Verbraucher fungiert. Die verbrauchte Energie wird als Licht der LED abgegeben. Die Anordnung von Vorwiderstand und LED könnte auch umgekehrt sein. Wichtig ist jedoch die Stellung der LED. Der Strom (die Elektronen) fließt zuerst durch die Anode (langes Bein der LED) und dann durch die Kathode (kurzes Bein). Bei umgekehrter Stellung würde der Stromfluss unterbrochen werden.

Merkhilfe: „Nach dem Stromfluss verliert die LED einen Teil des Beins“



Parallelschaltung

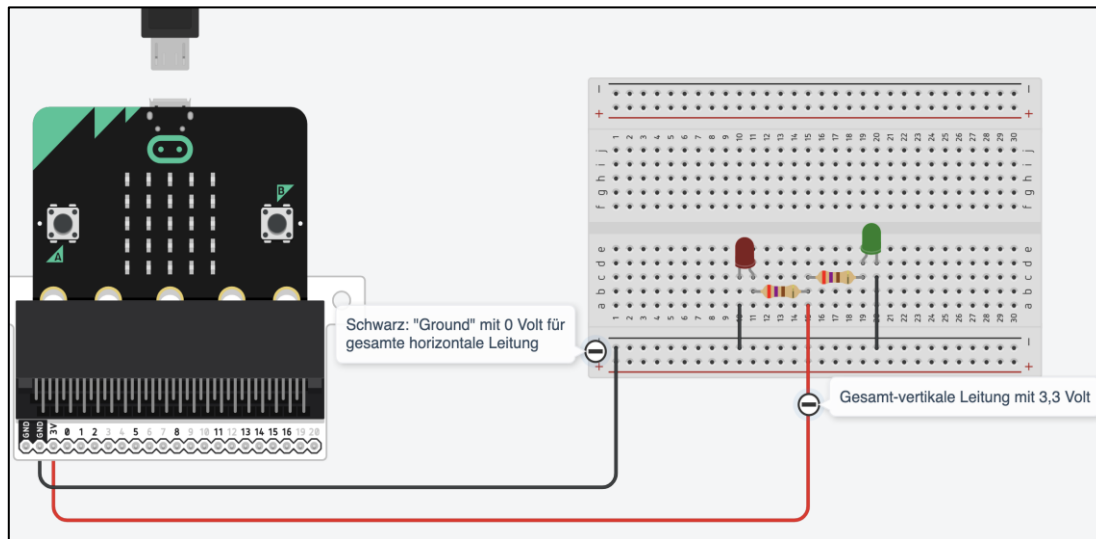


Abbildung 2: Parallelschaltung von 2 LEDs

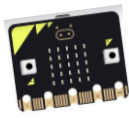
Die Abbildung 2 zeigt eine Parallelschaltung. Dabei wird die konstante Spannung von 3,3 Volt in der inneren, vertikalen Leiste des Steckbretts angelegt (siehe Abbildung 2, innere Bahn 15). Beide Widerstände werden hier gleichzeitig mit den vollen 3,3 Volt versorgt. Auch wenn ein linker Stromfluss durch die rote LED erfolgt, bleiben auch 3,3 Volt für die Verbindung mit der grünen LED übrig. Die Berechnungen für den Vorwiderstand sind ident mit der Abbildung 1 für den Serienwiderstand.

Datenblatt

Damit eine LED betrieben werden kann, muss immer zuvor ein Datenblatt²¹ der LED studiert werden (siehe Abbildung 3). Wichtig hierbei sind bei den LED-Typen rot, grün und gelb die **Durchlassspannung** und die **Stromstärke** durch die LED, in unserem Beispiel 2 V und 10 mA lt. Datenblatt. Gefährlich sind die Begriffe „**Max. zulässiger Strom**“ und **Durchbruchspannung**. Vor allem ein maximal zulässiger Strom von 30 mA ist bei den obigen Schaltungen nicht erreicht, da konkret beim Micro:bit mit maximal 5 mA²² gearbeitet wird. Die Durchbruchspannung von 5V würde auf alle Fälle die LED zerstören, wird aber natürlich nicht verwendet.

²¹ Musterdatenblatt unter Reichelt, https://cdn-reichelt.de/documents/datenblatt/C200/DS_GER_RF19.pdf (03.01.2024)

²² Siehe Empfehlung von 5 mA für Micro:bit unter <https://support.microbit.org/support/solutions/articles/19000101863-connecting-an-led-to-the-micro-bit> (25.03.2024)



Technische Daten LED

3 mm LED

(Kenndaten gültig bei 25 °C)	LED rot	LED grün	LED gelb
Max. zulässiger Strom I_F :	30 mA	30 mA	20 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,5 mA/°C	ca. 0,5 mA/°C	ca. 0,2 mA/°C
Typ. Wellenlänge:	635 nm	565 nm	586 nm
Typ. Durchlassspannung U_F bei I_F :	2 V/10 mA	2 V/10 mA	2 V/10 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 μ A	min. 5 V/100 μ A	min. 5 V/100 μ A
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C
	LED blau	LED weiss	LED grün superhell
Max. zulässiger Strom I_F :	20 mA	25 mA	30 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,6 mA/°C	-	-
Typ. Wellenlänge:	470 nm	-	510-545 nm
Typ. Durchlassspannung U_F bei I_F :	2,7 V/10 mA	3,6 V/20 mA	3,5 V/20 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 μ A	-	-
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	-30 °C ... + 100 °C

Abbildung 3: Datenblatt LED

Vorwiderstand berechnen

Für beide Schaltungen (Reihen- und Parallelschaltung) muss eine empfohlene Durchlassspannung von 2 V bei der LED erreicht werden. Somit muss ein geeigneter Vorwiderstand geschaltet werden, der die Ausgangsspannung von 3,3 V auf 2 V des LEDs senkt für eine Differenz von 1,3 Volt.

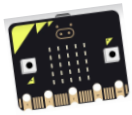
$$\text{Widerstand} = \frac{3,3V - 2V = 1,3V}{0,005\text{Ampere}(5mA)}$$

Dies ergibt einen Vorwiderstandswert von 260 Ω (Ohm). Praktisch wird aber der nächsthöhere Widerstand, in diesem Fall 270 Ω , verwendet. In der Praxis ist es selten möglich, einen genau berechneten Widerstand als Bauteil zu finden.

Korrekten Widerstand verwenden

Physische Widerstände sind durch einen speziellen Farbcode gekennzeichnet. In der häufigsten Form handelt es sich um 4 oder 5 Ringe, die zur Bestimmung des Widerstands beachtet werden müssen. Eine genau Berechnung erledigen Farbtabelle oder Online-Rechner²³. Beachte beim Lesen der Ringe die Anordnungs-Richtung. Der von

²³ Digikey, online unter <https://www.digikey.de/de/resources/conversion-calculators/conversion-calculator-resistor-color-code> (03.01.2024)



den Betrags-Ringen weiter entfernte Toleranz-Ring, zeigt das Ende der Leserichtung. Wichtig ist die Berücksichtigung der Anzahl an tatsächlich vorhandenen Ringen am Widerstand.

Anzahl der Bänder
4 Ringe **5 Ringe** 6 Ringe

Widerstandsparameter

1. Farbring
Rot 2

2. Farbring
Violett 7

3. Farbring
Schwarz 0

Multiplikator
Schwarz $\times 1 \Omega$

Toleranz
Braun $\pm 1\%$

Resistance value
270 Ω

Ausgabe

2 7 0 $\times 1 \Omega$ $\pm 1\%$

Widerstandswert:
270 Ohms 1%

Abbildung 4: Farbdarstellungen eines Vorwiderstands von 270 Ohm

LED richtig stecken

LEDs bestehen aus Anode (langes Bein) und Kathode (kurzes Bein). Zu beachten ist die Richtung des Stromflusses, da dieser zuerst durch die Anode (langes Bein) und dann erst durch die Kathode fließt. Wird die LED falsch gesteckt, findet KEIN Stromfluss statt. LEDs leiten somit Strom in die richtige Richtung, dienen aber auch als optische Hinweise (z.B. rote LED als Fehlermeldung).

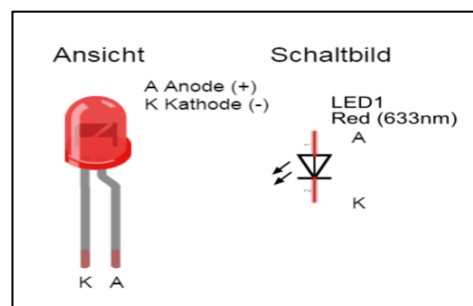
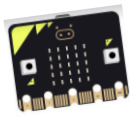


Abbildung 5: Stromfluss von Anode+ zu Kathode-(Quelle²⁴)

²⁴ Elab, LED, https://robotfreak.de/elab-wiki/index.php?title=LED_Leuchtdiode (abgerufen am 03.01.2024)



Schaltungsentwurf mit TinkerCAD

Optional kann mit oder ohne inkludierte, visuelle Programmiersprache das Funktionieren der Schaltung vorab getestet werden.

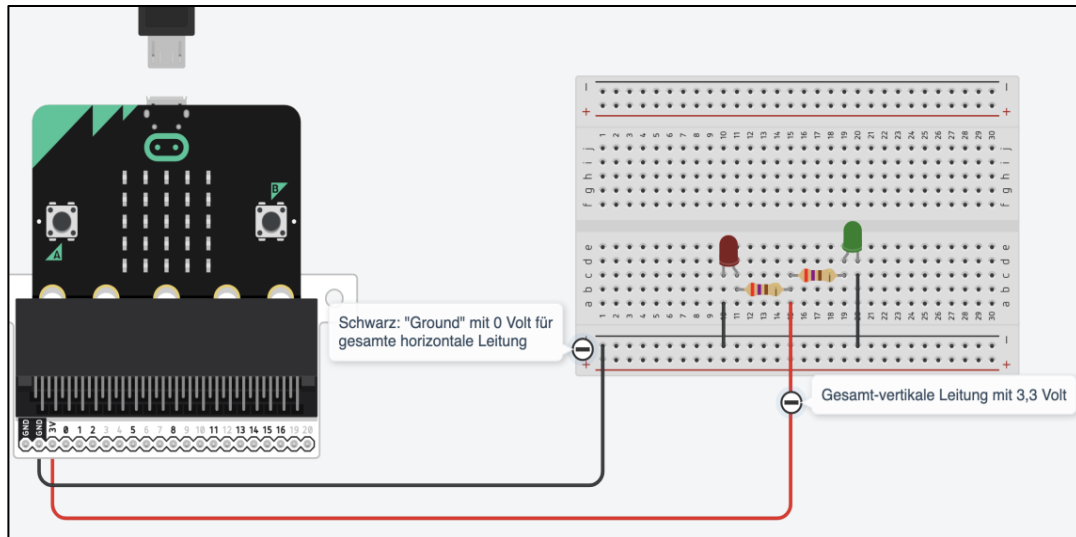


Abbildung 6: Parallelschaltung von zwei Stromkreisen in TinkerCAD

Zielgruppen

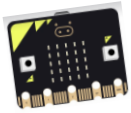
Lehrpersonen und Schüler*innen der SEK1 und SEK2

Ideen zur Erweiterung/Adaptierung/Vertiefung

- Stromversorgung über externe Batterie bzw. Batteriehalterung
- Zusätzliche Reihen- und Parallelschaltungen
- Einsatz des Micro:bit mit Krokodilklemmen ohne Breakout-Board

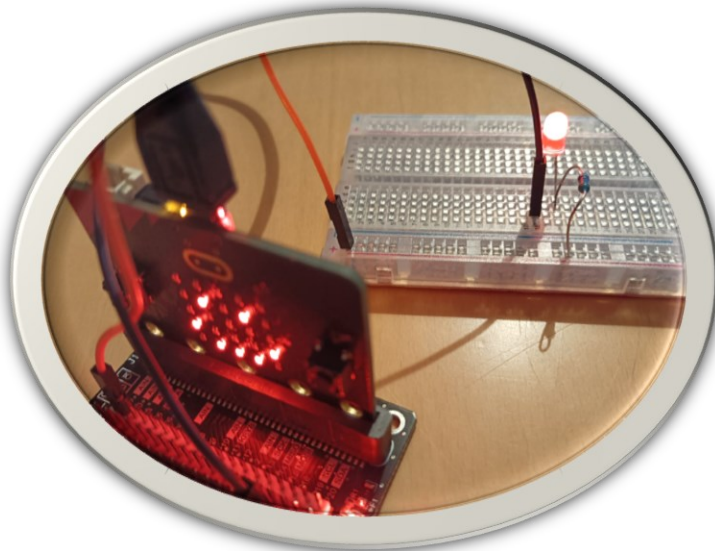
Ergänzende Lernunterlagen

- Micro:bit und Projekte bzw. Erklärungen unter <https://microbit.org/> (25.03.2024)
- Grundlagen der Elektronik unter <https://studyflix.de/elektrotechnik/thema/elektrotechnik-grundlagen-20> (25.03.2024) bzw. <https://www.elektrotechnik-einfach.de/> (25.03.2024)



13.3 Praxisprojekt 2 (Stromschalten per PIN)

Projekt 2 – Stromschalten per PIN



Benötigte Komponenten

- Micro:bit V2 mit Breakout-Board
- Starterkit GeekPi BBC Micro:bit V2 Ultimate Starter Kit optional²⁵
- Steckbrett
- 1 Light-Emitting-Diode (LED) rot
- 1 Widerstand mit 270 Ohm
- Jumperkabel "male-female"

Benötigte Software

- TinkerCAD²⁶
- MakeCode-Editor von Microsoft²⁷
- Python-Editor für MicroPython²⁸
- Mu-Editor²⁹ lokal installiert

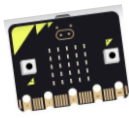
²⁵ Siehe online unter https://www.amazon.de/GeekPi-BBC-Micro-Ultimate-microbit/dp/B0BKXLM97?ref=ast_sto_dp (20.03.2024)

²⁶ Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.03.2024)

²⁷ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

²⁸ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

²⁹ Downloadbar unter <https://codewith.mu/en/download> (25.03.2024)



Generelle Vorsichtsmaßnahmen

- Niemals den 3,3-Volt-Anschluss direkt (ohne Vorwiderstand) mit Ground verbinden
- Bei elektronischen Steckübungen oder Veränderungen immer zuerst den Micro:bit vom Strom nehmen (von Batterie oder USB-Anschluss)
- Sich selbst vor der Arbeit mit Elektronik elektrostatisch entladen (Berühren von metallischen Teilen wie Heizkörpern, Metalltischen etc.)

Projektbeschreibung

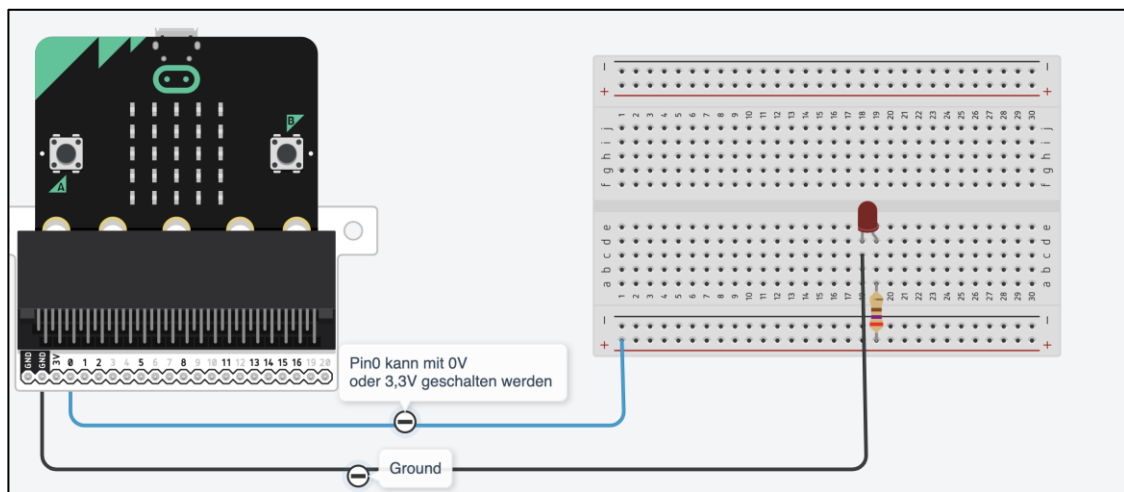
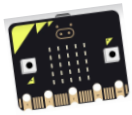


Abbildung 1: LED über Pin 0 ein- und ausschalten

Die Abbildung 1 zeigt eine reine Reihenschaltung, deren Stromversorgung über einen schaltbaren Pin0 erfolgt.

Ein Vorwiderstand von 270 Ohm und eine LED sind in Reihe (hintereinander) geschaltet und der Strom fließt durch beide Komponenten zum Ground. Die Gesamtspannung von 3,3 Volt wird zunächst durch den Vorwiderstand auf 2 Volt reduziert und fließt anschließend durch die LED als Verbraucher. Die verbrauchte Energie wird als Licht der LED abgegeben. Die Anordnung von Vorwiderstand und LED könnte auch umgekehrt sein. Wichtig ist jedoch die Stellung der LED. Der Strom (die Elektronen) fließt zuerst durch die Anode (langes Bein der LED) und dann durch die Kathode (kurzes Bein). Bei umgekehrter Stellung der LED würde der Stromfluss unterbrochen werden.

Merkhilfe: „Nach dem Stromfluss verliert die LED einen Teil des Beins“



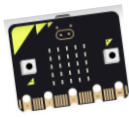
Die rote LED ist nicht dauerhaft an der Stromversorgung angeschlossen, sondern erhält ihre Spannung über den Pin 0. Wird dieser digital eingeschaltet, ist der Pin0 in einem HIGH-Zustand „1“. Dies bedeutet, dass der Pin0 auf die Betriebsspannung des Mikrocontrollers gesetzt wird – beim Micro:bit 3,3 Volt. Dadurch wird ein Stromfluss ermöglicht, wenn der Pin 0 mit einem externen Schaltkreis verbunden ist, der einen Pfad zur Masse (GND) bietet.

Datenblatt

Damit eine LED betrieben werden kann, muss immer zuvor ein Datenblatt³⁰ der LED studiert werden (siehe Abbildung 2). Wichtig hierbei sind bei den LED-Typen rot, grün und gelb die **Durchlassspannung** und die **Stromstärke** durch die LED, in unserem Beispiel 2 V und 10 mA. lt. Datenblatt. Gefährlich sind die Begriffe „**Max. zulässiger Strom**“ und **Durchbruchspannung**. Vor allem der maximal zulässige Strom von 30 mA ist bei der obigen Schaltung nicht gegeben, da konkret beim Micro:bit mit 5 mA³¹ gearbeitet wird. Die Durchbruchspannung von 5V würde auf alle Fälle das LED zerstören, wird aber natürlich nie verwendet.

³⁰ Musterdatenblatt unter Reichelt, https://cdn-reichelt.de/documents/datenblatt/C200/DS_GER_RF19.pdf (abgerufen am 3.1.2024)

³¹ Siehe Empfehlung von 5 mA für Micro:bit unter <https://support.microbit.org/support/solutions/articles/19000101863-connecting-an-led-to-the-micro-bit> (25.03.2024)



Technische Daten LED

3 mm LED

(Kennwerte gültig bei 25 °C)	LED rot	LED grün	LED gelb
Max. zulässiger Strom I_F :	30 mA	30 mA	20 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,5 mA/°C	ca. 0,5 mA/°C	ca. 0,2 mA/°C
Typ. Wellenlänge:	635 nm	565 nm	586 nm
Typ. Durchlassspannung U_F bei I_F :	2 V/10 mA	2 V/10 mA	2 V/10 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 μ A	min. 5 V/100 μ A	min. 5 V/100 μ A
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C
	LED blau	LED weiß	LED grün superhell
Max. zulässiger Strom I_F :	20 mA	25 mA	30 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,6 mA/°C	-	-
Typ. Wellenlänge:	470 nm	-	510-545 nm
Typ. Durchlassspannung U_F bei I_F :	2,7 V/10 mA	3,6 V/20 mA	3,5 V/20 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 μ A	-	-
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	-30 °C ... + 100 °C

Abbildung 2: Datenblatt LED

Vorwiderstand berechnen

Für die vorliegende Reihenschaltung muss eine empfohlene Durchlassspannung von 2 V erreicht werden. Somit sollte ein geeigneter Vorwiderstand geschaltet werden, der die Ausgangsspannung von 3,3 V auf 2 V für die LED senkt - eine Differenz von 1,3 Volt.

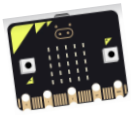
$$\text{Widerstand} = \frac{3,3V - 2V = 1,3V}{0,005\text{Ampere}(5mA)}$$

Dies ergibt einen Vorwiderstandswert von 260 Ω (Ohm). Praktisch wird aber der nächsthöhere Widerstand, in diesem Fall 270 Ω , verwendet. In der Praxis ist der berechnete Widerstand selten exakt als verfügbares Bauteil vorhanden.

Korrekten Widerstand verwenden

Physische Widerstände sind durch einen speziellen Farbcode gekennzeichnet. In der häufigsten Form handelt es sich um 4 oder 5 Ringe, die zur Bestimmung des Widerstands beachtet werden müssen. Eine genau Berechnung erledigen Farbtabelle oder Online-Rechner³². Beachte beim Lesen der Ringe die Anordnungs-Richtung. Der von

³² Digikey, online unter <https://www.digikey.de/de/resources/conversion-calculators/conversion-calculator-resistor-color-code> (03.01.2024)



den Betrags-Ringen weiter entfernte Toleranz-Ring, zeigt das Ende der Leserichtung. Wichtig ist die Berücksichtigung der Anzahl an tatsächlich vorhandenen Ringen am Widerstand.

Anzahl der Bänder

4 Ringe
5 Ringe
6 Ringe

Widerstandsparameter

1. Farbring

Rot2

2. Farbring

Violett7

3. Farbring

Schwarz0

Multiplikator

Schwarz×1 Ω

Toleranz

Braun± 1%

Resistance value

270Ω

Ausgabe

Widerstandswert:
270 Ohms 1%

Abbildung 3: Farbdarstellung eines Vorwiderstands von 270 Ohm

LED richtig stecken

LEDs bestehen aus Anode (langes Bein) und Kathode (kurzes Bein). Zu beachten ist die Richtung des Stromflusses, da dieser zuerst durch die Anode (langes Bein) und dann erst durch die Kathode fließt. Wird die LED falsch gesteckt, findet KEIN Stromfluss statt. LEDs leiten somit Strom in die richtige Richtung, dienen aber auch als optische Hinweise (z.B. rotes LED als Fehlermeldung).

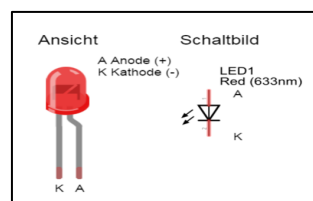
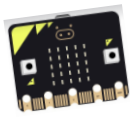


Abbildung 4: Stromfluss von Anode+ zu Kathode-(Quelle³³)

³³ Elab, LED, https://robotfreak.de/elab-wiki/index.php?title=LED_Leuchtdiode (03.01.2024)



Schaltungsentwurf mit TinkerCAD

Optional kann mit oder ohne inkludierte, visuelle Programmiersprache das Funktionieren der Schaltung vorab getestet werden.

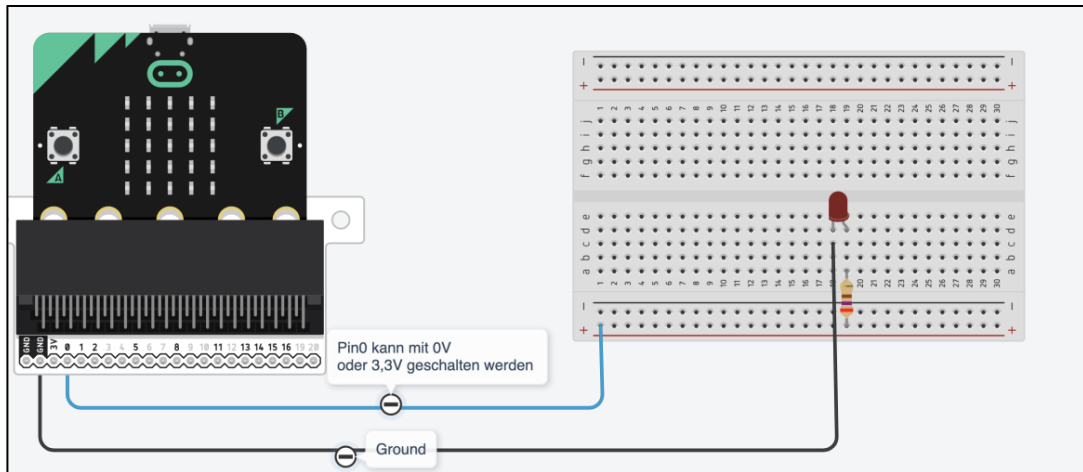


Abbildung 5: Schaltungsentwurf mit TinkerCAD

Programmierung grafisch mit MakeCode-Editor

Wichtig beim MakeCode-Editor ist die Auswahl der Gruppe PINS links. Hier befinden sich die wichtigsten Funktionen für digitale und analoge PIN-Schaltungen.

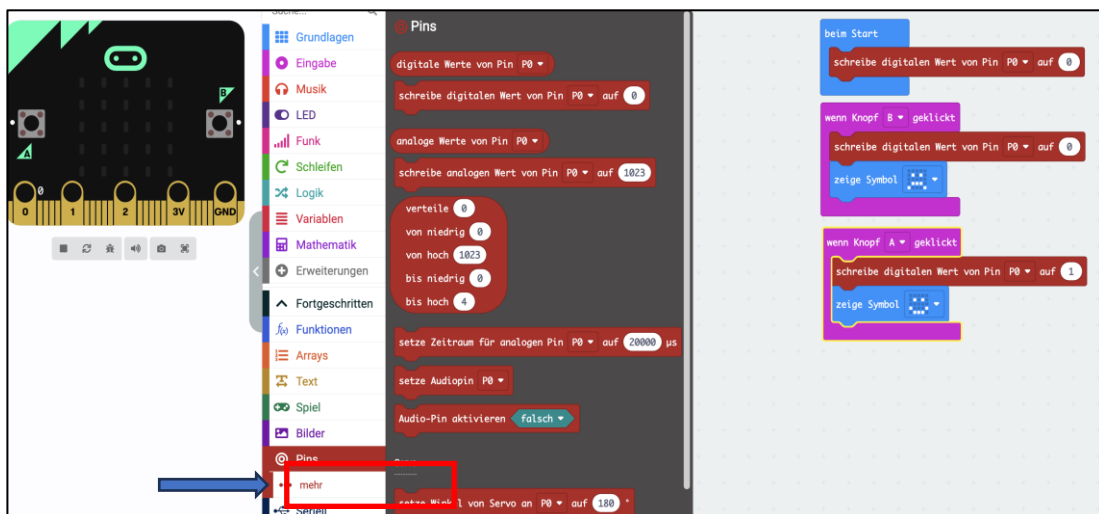
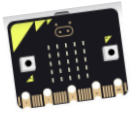


Abbildung 6: Grafische Programmierung mit MakeCode



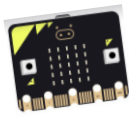
Programmierung mit MicroPython Web bzw. Mu-Editor lokal

Interessant vor allem in der MicroPython-Umsetzung sind die *While-Schleife* und das *If-Statement*. Wichtig ist zu verstehen, dass durch die Dauerschleife mit `while(True)` das Gerät in einem Dauerbereitschaftsmodus ist, der darauf wartet, dass ein Button gedrückt wird (A oder B). Ohne die *While-Schleife* wäre eine kontinuierliche Abfrage der Buttons nicht möglich.

Die Verzögerung mit `sleep(100)` unterbricht die Dauerschleife immer kurz um 100 Millisekunden, damit der Prozessor keine dauernde Höchstbelastung erfährt bzw. um einer etwaigen Entprellung der Buttons vorzugreifen. Entprellung ist ein Effekt, bei dem Buttons bei einem einzigen Tastendruck oft mehrfache Event-Signale erhalten und dann unkontrolliert oder gar nicht reagieren.

```
1  from microbit import *
2  # Pin0 zu Beginn auf 0
3  pin0.write_digital(0)
4  while True:
5      if button_a.is_pressed():
6          # Spannung auf 1 bzw. 3,3V bei Pin0
7          pin0.write_digital(1)
8          display.show(Image.HAPPY)
9
10     if button_b.is_pressed():
11         # Spannung auf 0 bzw. 0V bei Pin0
12         pin0.write_digital(0)
13         display.show(Image.SAD)
14     # kurzer Interrupt innerhalb der Schleife
15     sleep(100)
```

Abbildung 7: Programmierung im Python-Editor



```
1 from microbit import *
2 # Pin0 zu Beginn auf 0
3 pin0.write_digital(0)
4 while True:
5     if button_a.is_pressed():
6         # Spannung auf 1 bzw. 3,3V bei Pin0
7         pin0.write_digital(1)
8         display.show(Image.HAPPY)
9
10    if button_b.is_pressed():
11        # Spannung auf 0 bzw. 0V bei Pin0
12        pin0.write_digital(0)
13        display.show(Image.SAD)
14    # kurzer Interrupt innerhalb der Schleife
15    sleep(100) |
```

Abbildung 8: Gleicher Code im Mu-Editor

Zielgruppen

Lehrpersonen und Schüler*innen der SEK1 und SEK2

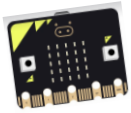
Ideen zur Erweiterung/Adaptierung/Vertiefung

- Zusätzliche Schaltkreise implementieren, die alle mit Button A und B ein- und ausgeschaltet werden
- Statt des Button A den Logo-Pin-Button verwenden
- Wenn Zustand auf „1“ ist, akustische Signale einbauen
- Ev. absichtlich die LED falsch stecken zur Demonstration der Sperrung

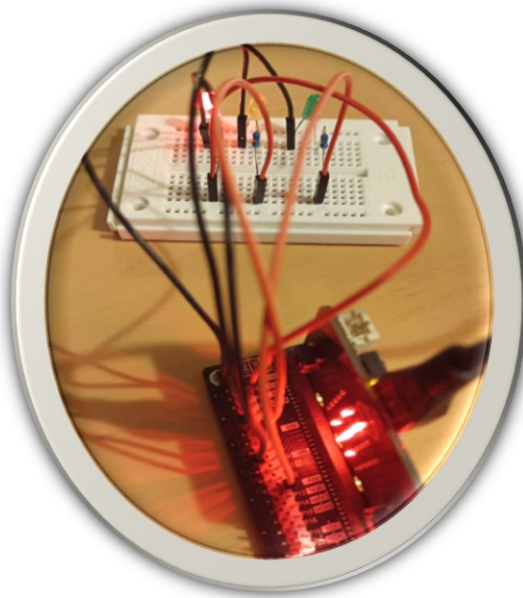
Ergänzende Lernunterlagen

- Micro:bit, Projekte und Erklärungen unter <https://microbit.org/> (25.03.2024)
- Grundlagen der Elektronik unter <https://studyflix.de/elektrotechnik/thema/elektrotechnik-grundlagen-20> (25.03.2024) bzw. <https://www.elektrotechnik-einfach.de/> (25.03.2024)
- Dokumentation für MicroPython für Micro:bit unter <https://microbit-micropython.readthedocs.io/en/v2-docs/> (24.03.2024)
- Entprellung unter <https://www.mikrocontroller.net/articles/Entprellung> (25.03.2024)

13.4 Praxisprojekt 3 (Ampelschaltung)



Projekt 3 – Ampelschaltung



Benötigte Komponenten

- Micro:bit V2 mit Breakout-Board
- Starterkit GeekPi BBC Micro:bit V2 Ultimate Starter Kit optional³⁴
- Steckbrett
- 3 Light-Emitting-Diode (LED) rot, grün, orange
- 3 Widerstände mit je ca. 270 Ohm
- Jumperkabel “male-female” und “male-male”

Benötigte Software

- TinkerCAD³⁵
- MakeCode-Editor von Microsoft³⁶
- Python-Editor für MicroPython³⁷
- Mu-Editor³⁸ lokal installiert

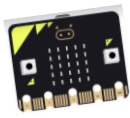
³⁴ Siehe online unter https://www.amazon.de/GeekPi-BBC-Micro-Ultimate-microbit/dp/B0BKXLM97?ref=ast_sto_dp (20.03.2024)

³⁵ Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.03.2024)

³⁶ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

³⁷ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

³⁸ Downloadbar unter <https://codewith.mu/en/download> (25.03.2024)



Generelle Vorsichtsmaßnahmen

- Niemals den 3,3-Volt-Anschluss direkt (ohne Vorwiderstand) mit Ground verbinden
- Bei elektronischen Steckübungen oder Veränderungen immer zuerst den Micro:bit vom Strom nehmen (von Batterie oder USB-Anschluss)
- Sich selbst vor der Arbeit mit Elektronik elektrostatisch entladen (Berühren von metallischen Teilen wie Heizkörpern, Metalltischen etc.)

Projektbeschreibung

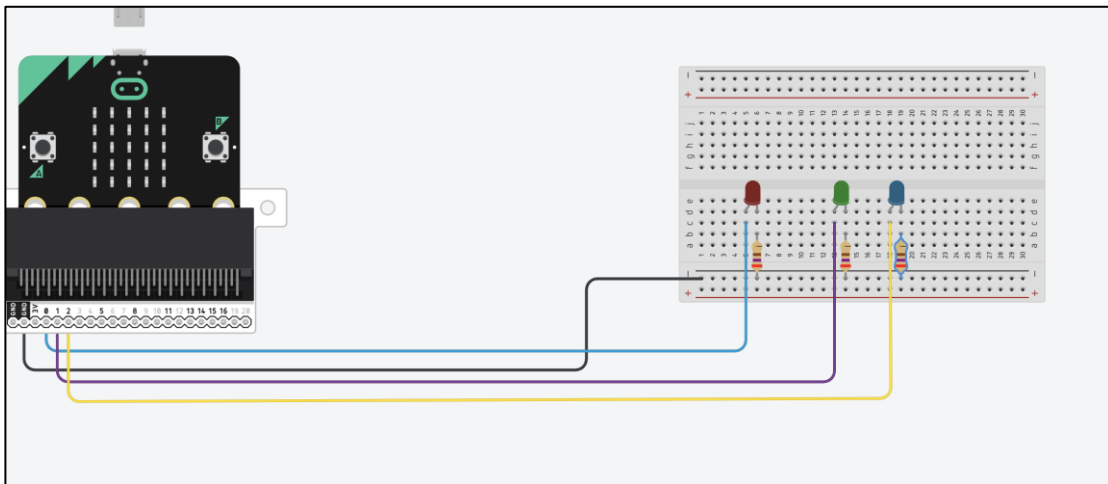
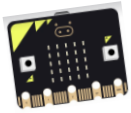


Abbildung 1 LEDs über Pin0, Pin1 und Pin2 ein- und ausschalten

Die Abbildung 1 zeigt drei reine Reihenschaltungen, deren Stromversorgung über die schaltbaren Pin 0, Pin 1 und Pin 2 erfolgt. Durch wechselhaftes Ein- und Ausschalten der PINs wird eine Lichtampel simuliert. Die Vorwiderstände sind alle direkt mit der Ground-Leiste verbunden, leiten also alle zu Ground.

Drei Vorwiderstände von 270 Ohm und drei LEDs sind in Reihe (hintereinander) geschaltet und der Strom fließt durch alle Komponenten zum Ground. Die Gesamtspannung von 3,3 Volt wird zuerst beim Vorwiderstand auf 2 Volt reduziert und fließt dann durch die LED als Verbraucher. Die verbrauchte Energie wird als Licht der LED abgegeben. Die Anordnung von Vorwiderstand und LED könnte auch umgekehrt sein. Wichtig ist jedoch die Stellung der LED. Der Strom (die Elektronen) fließt zuerst durch die Anode (langes Bein der LED) und dann durch die Kathode (kurzes Bein). Bei umgekehrter Stellung würde der Stromfluss unterbrochen werden.



Merkhilfe: „*Nach dem Stromfluss verliert die LED einen Teil des Beins*“

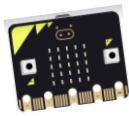
Die drei LEDs sind nicht dauerhaft an der Stromversorgung angeschlossen, sondern erhalten ihre Spannung über den Pin 0, Pin 1 und Pin 2. Wird dieser digital eingeschaltet, ist der Pin in einem HIGH-Zustand „1“. Dies bedeutet, dass der PIN auf die Betriebsspannung des Mikrocontrollers gesetzt wird – beim Micro:bit 3,3 Volt. Dadurch wird tatsächlich ein Stromfluss ermöglicht, wenn der PIN mit einem externen Schaltkreis verbunden ist, der einen Pfad zur Masse (GND) bietet.

Datenblatt

Damit eine LED beleuchtet werden kann, muss immer zuvor ein Datenblatt³⁹ der LED studiert werden (siehe Abbildung 2). Wichtig hierbei sind bei den LED-Typen rot, grün und gelb die **Durchlassspannung** und die **Stromstärke** durch die LED, in unserem Beispiel 2 V und 10 mA gemäß Datenblatt. Gefährlich sind die Begriffe „**Max. zulässiger Strom**“ und **Durchbruchspannung**. Vor allem der maximal zulässige Strom von 30 mA ist bei den obigen Schaltungen kein Thema, da konkret beim Micro:bit mit 5 mA⁴⁰ gearbeitet wird. Die Durchbruchspannung von 5V würde auf alle Fälle die LED zerstören, wird aber natürlich ebenso wenig verwendet.

³⁹ Musterdatenblatt unter Reichelt, https://cdn-reichelt.de/documents/datenblatt/C200/DS_GER_RF19.pdf (abgerufen am 3.1.2024)

⁴⁰ Siehe Empfehlung von 5 mA für Micro:bit unter <https://support.microbit.org/support/solutions/articles/19000101863-connecting-an-led-to-the-micro-bit> (25.03.2024)



Technische Daten LED

3 mm LED

(Kenndaten gültig bei 25 °C)	LED rot	LED grün	LED gelb
Max. zulässiger Strom I_F :	30 mA	30 mA	20 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,5 mA/°C	ca. 0,5 mA/°C	ca. 0,2 mA/°C
Typ. Wellenlänge:	635 nm	565 nm	586 nm
Typ. Durchlassspannung U_F bei I_F :	2 V/10 mA	2 V/10 mA	2 V/10 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 μ A	min. 5 V/100 μ A	min. 5 V/100 μ A
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C
	LED blau	LED weiß	LED grün superhell
Max. zulässiger Strom I_F :	20 mA	25 mA	30 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,6 mA/°C	-	-
Typ. Wellenlänge:	470 nm	-	510-545 nm
Typ. Durchlassspannung U_F bei I_F :	2,7 V/10 mA	3,6 V/20 mA	3,5 V/20 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 μ A	-	-
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	-30 °C ... + 100 °C

Abbildung 2: Datenblatt LED⁴¹

Vorwiderstand berechnen

Für alle Schaltungen muss eine empfohlene Durchlassspannung von 2 V erreicht werden. Somit muss ein geeigneter Vorwiderstand geschaltet werden, der die Ausgangsspannung von 3,3 V auf 2 V für die LED reduziert, eine Differenz von 1,3 Volt.

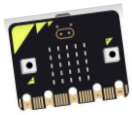
$$\text{Widerstand} = \frac{3,3V - 2V = 1,3V}{0,005\text{Ampere}(5mA)}$$

Dies ergibt einen Vorwiderstandswert von 260 Ω (Ohm). Praktisch wird aber der nächsthöhere Widerstand, in diesem Fall 270 Ω , verwendet. In der Praxis wird der berechnete Widerstand selten als exaktes Bauteil gefunden.

Korrekten Widerstand verwenden

Physische Widerstände sind durch einen speziellen Farbcode gekennzeichnet. In der häufigsten Form handelt es sich um 4 oder 5 Ringe, die zur Bestimmung des Widerstands beachtet werden müssen. Eine genau Berechnung erledigen Farbtabelle oder

⁴¹ Datenblatt unter https://cdn-reichelt.de/documents/datenblatt/C200/DS_GER_RF19.pdf (30.03.2024)



Online-Rechner⁴². Beachte beim Lesen der Ringe die Anordnungs-Richtung. Der von den Betrags-Ringen weiter entfernte Toleranz-Ring, zeigt das Ende der Leserichtung. Wichtig ist die Berücksichtigung der Anzahl an tatsächlich vorhandenen Ringen am Widerstand.

Anzahl der Bänder
4 Ringe **5 Ringe** 6 Ringe

Widerstandsparameter

1. Farbring
Rot 2

2. Farbring
Violett 7

3. Farbring
Schwarz 0

Multiplikator
Schwarz $\times 1 \Omega$

Toleranz
Braun $\pm 1\%$

Resistance value
270 Ω

Ausgabe

Widerstandswert:
270 Ohms 1%

Abbildung 3: Farbdarstellung eines Vorwiderstands von 270 Ohm

LED richtig stecken

LED bestehen aus Anode (langes Bein) und Kathode (kurzes Bein). Zu beachten ist die Richtung des Stromflusses, da dieser zuerst durch die Anode (langes Bein) und dann erst durch die Kathode fließt. Wird die LED falsch gesteckt, findet KEIN Stromfluss statt. LEDs leiten somit Strom in die richtige Richtung, dienen aber auch als optische Hinweise (z.B. rotes LED als Fehlermeldung).

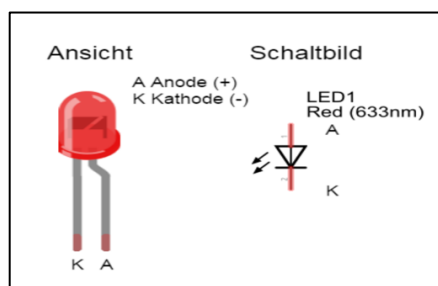
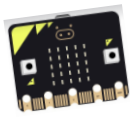


Abbildung 4: Stromfluss von Anode+ zu Kathode-(Quelle⁴³)

⁴² Digikey, online unter <https://www.digikey.de/de/resources/conversion-calculators/conversion-calculator-resistor-color-code> (03.01.2024)

⁴³ Elab, LED, https://robotfreak.de/elab-wiki/index.php?title=LED_Leuchtdiode (03.01.2024)



Schaltungsentwurf mit TinkerCAD

Optional kann mit oder ohne inkludierte, visuelle Programmiersprache das Funktionieren der Schaltung vorab getestet werden.

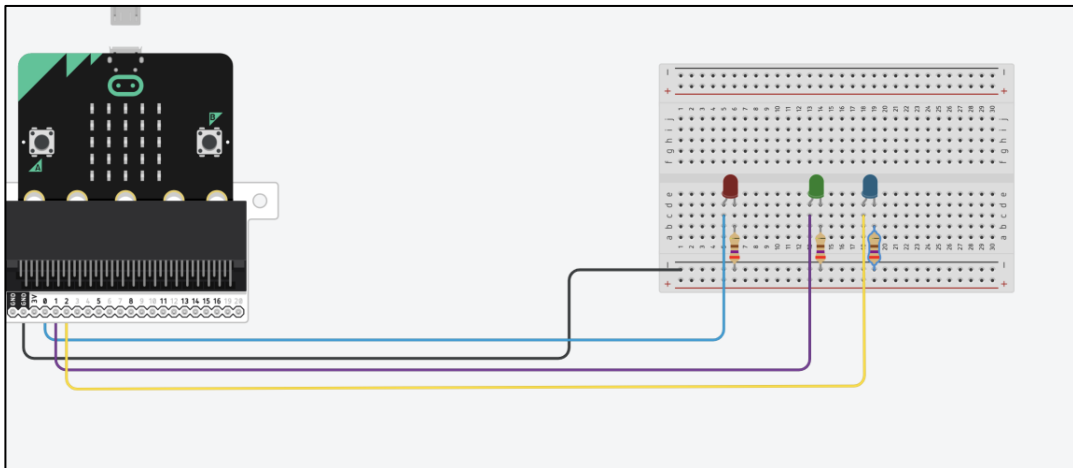


Abbildung 5: Schaltungsentwurf mit TinkerCAD

Programmierung grafisch mit MakeCode-Editor

Wichtig beim MakeCode-Editor ist die Auswahl der Gruppe PINS links. Hier befinden sich die wichtigsten Funktionen für digitale und analoge Anweisungen der PINS.

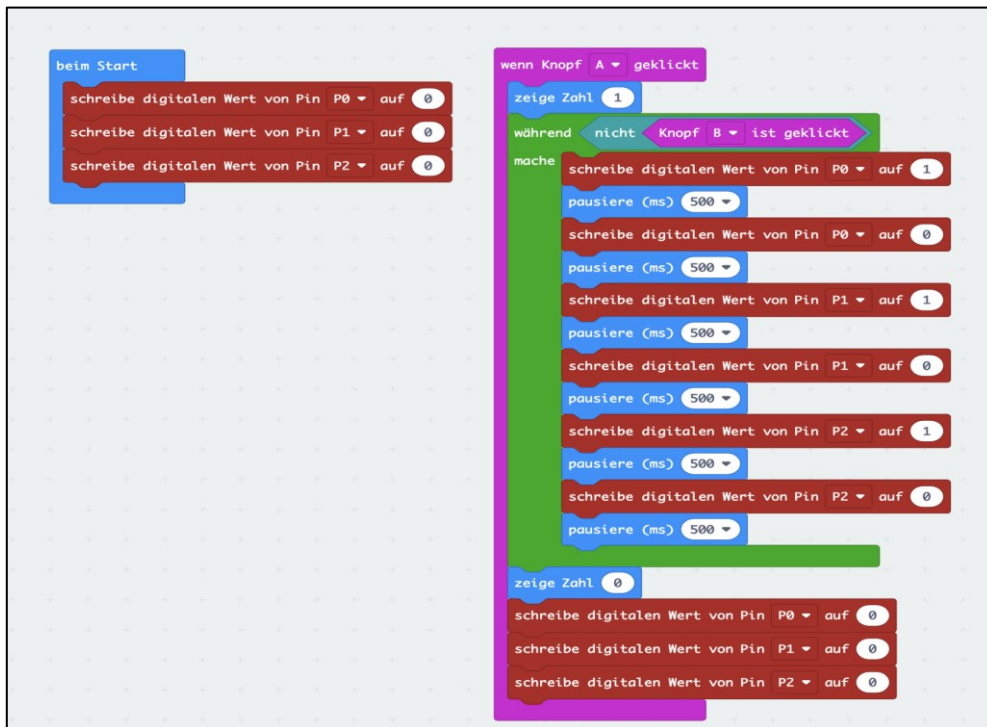
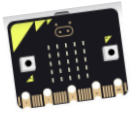
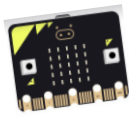


Abbildung 6: Grafische Programmierung mit MakeCode

Programmierung mit MicroPython Web bzw. Mu-Editor lokal

Interessant vor allem in der MicroPython-Umsetzung sind die *While-Schleife* und das *If-Statement*. Wichtig ist zu verstehen, dass durch die Dauerschleife mit `while(True)` das Gerät in einem Dauerbereitschaftsmodus ist, der darauf wartet, dass ein Button gedrückt wird (A oder B). Ohne die *While-Schleife* wäre eine kontinuierliche Abfrage der Buttons nicht möglich.

Da die Ampelschaltung nach Drücken von Button A so dauerhaft sein soll, bis Button B gedrückt wird, muss eine zweite Schleife mit `while not (button_b.isPressed())` eingebaut werden. Diese Schleife „lauscht“, ob eventuell der Button B gedrückt wurde. Damit wird die innere Schleife verlassen, die PINs werden auf 0 gesetzt und es beginnt wieder ein „Lauschen“, ob Button A gedrückt wird über die äußerste Schleife.

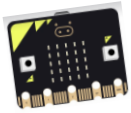


```
from microbit import *
while True:
    if button_a.is_pressed():
        # Kontrolle ob button_a gedrückt
        display.show("1")
        # solange button_b nicht gedrückt
        while not(button_b.is_pressed()):
            # pin0 ein-/ausschalten
            pin0.write_digital(1)
            sleep(500)
            pin0.write_digital(0)
            # pin1 ein-/ausschalten
            pin1.write_digital(1)
            sleep(500)
            pin1.write_digital(0)
            # pin1 ein-/ausschalten
            pin2.write_digital(1)
            sleep(500)
            pin2.write_digital(0)
        # button_b wurde gedrückt
        display.show("0")
        pin0.write_digital(0)
        pin1.write_digital(0)
        pin2.write_digital(0)
        # While Schleife kurz unterbrechen
        sleep(100)
```

Abbildung 7: Programmierung im Python-Editor

```
1 from microbit import *
2 while True:
3     if button_a.is_pressed():
4         # Kontrolle ob button_a gedrückt
5         display.show("1")
6         # solange button_b nicht gedrückt
7         while not (button_b.is_pressed()):
8             # pin0 ein-/ausschalten
9             pin0.write_digital(1)
10            sleep(500)
11            pin0.write_digital(0)
12            # pin1 ein-/ausschalten
13            pin1.write_digital(1)
14            sleep(500)
15            pin1.write_digital(0)
16            # pin1 ein-/ausschalten
17            pin2.write_digital(1)
18            sleep(500)
19            pin2.write_digital(0)
20        # button_b wurde gedrückt
21        display.show("0")
22        pin0.write_digital(0)
23        pin1.write_digital(0)
24        pin2.write_digital(0)
25        # While Schleife kurz unterbrechen
26        sleep(100)
```

Abbildung 8: Gleicher Code im Mu-Editor



Zielgruppen

Lehrpersonen und Schüler*innen der SEK1 und SEK2

Ideen zur Erweiterung/Adaptierung/Vertiefung

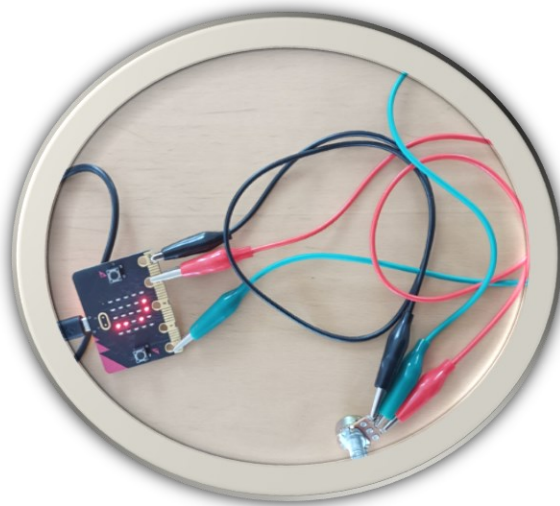
- Unterschiedliche Sleep-Sequenzen
- Einzelne LEDs öfter als einmal blinken lassen
- Alle Pins gleichzeitig ein- und ausschalten
- Erstellen einer "Light Show"
- Ampelphasen verschiedener Länder nachbauen und programmieren

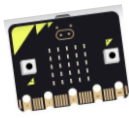
Ergänzende Lernunterlagen

- Micro:bit, Projekte und Erklärungen unter <https://microbit.org/> (25.03.2024)
 - Grundlagen der Elektronik unter <https://studyflix.de/elektrotechnik/thema/elektrotechnik-grundlagen-20> (25.03.2024) bzw. <https://www.elektrotechnik-einfach.de/> (25.03.2024)
 - Dokumentation für MicroPython für Micro:bit unter <https://microbit-micropython.readthedocs.io/en/v2-docs/> (24.03.2024)
 - Entprellung unter <https://www.mikrocontroller.net/articles/Entprellung> (25.03.2024)
-

13.5 Praxisprojekt 4 (Potentiometer)

Projekt 4 – Potentiometer





Benötigte Komponenten

- Micro:bit V2 mit Breakout-Board
- Starterkit GeeekPi BBC Micro:bit V2 Ultimate Starter Kit optional⁴⁴
- Steckbrett
- 1 Potentiometer „*Rotation Sensor*“
- Jumperkabel “female-female”

Benötigte Software

- TinkerCAD⁴⁵
- MakeCode-Editor von Microsoft⁴⁶
- Python-Editor für MicroPython⁴⁷
- Mu-Editor⁴⁸ lokal installiert

Generelle Vorsichtsmaßnahmen

- Niemals den 3,3-Volt-Anschluss direkt (ohne Vorwiderstand) mit Ground verbinden
- Bei elektronischen Steckübungen oder Veränderungen immer zuerst den Micro:bit vom Strom nehmen (von Batterie oder USB-Anschluss)
- Sich selbst vor der Arbeit mit Elektronik elektrostatisch entladen (Berühren von metallischen Teilen wie Heizkörpern, Metalltischen etc.)

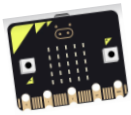
⁴⁴ Siehe online unter https://www.amazon.de/GeeekPi-BBC-Micro-Ultimate-microbit/dp/B0BKKXLM97?ref=ast_sto_dp (20.03.2024)

⁴⁵ Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.03.2024)

⁴⁶ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁴⁷ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁴⁸ Downloadbar unter <https://codewith.mu/en/download> (25.03.2024)



Projektbeschreibung

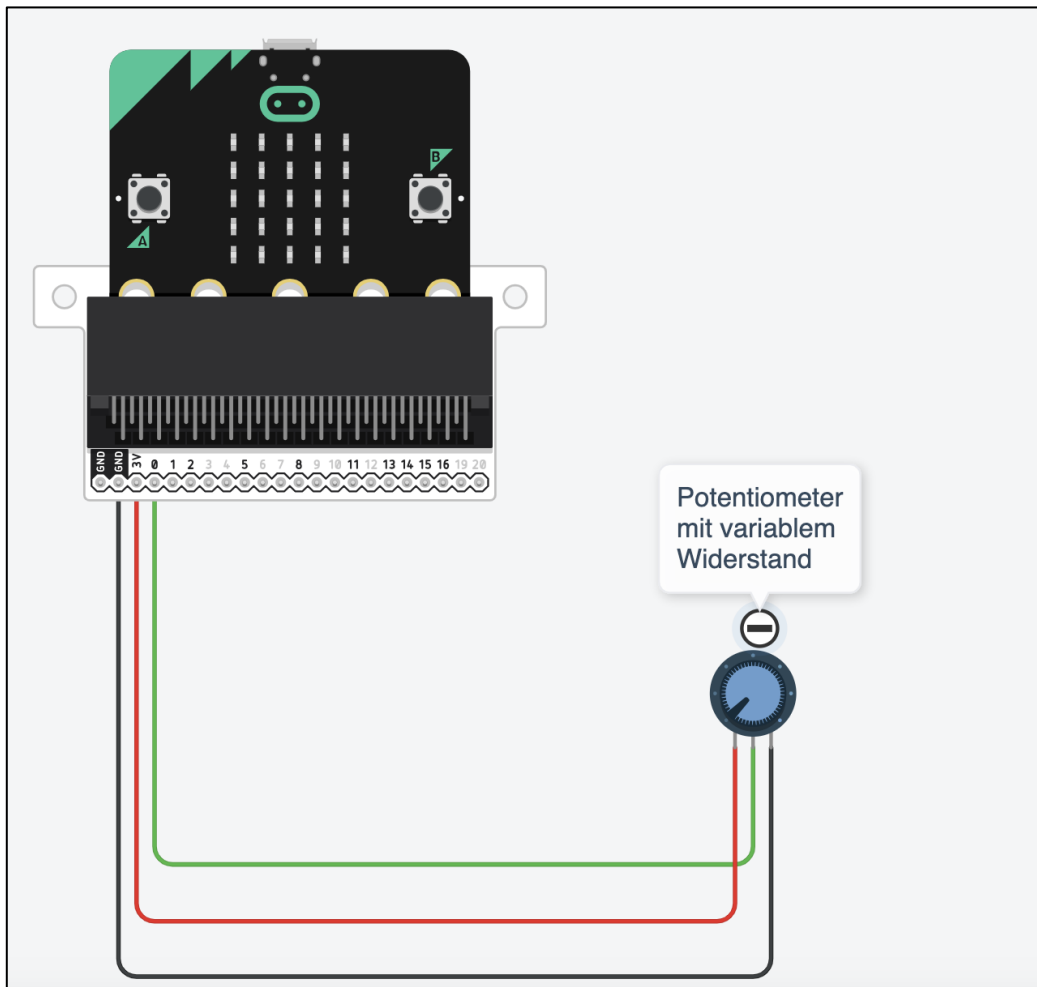
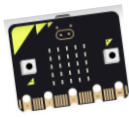


Abbildung 1: Variabler Widerstand mit einem Potentiometer

Die Abbildung 1 zeigt den Anschluss eines Potentiometers mit einem verstellbaren Widerstandswert. Der Potentiometer wird über einen VCC-Anschluss mit konstanten 3,3 Volt verbunden sowie über eine Leitung mit Ground. Der mittlere PIN am Potentiometer schickt seine Messwerte an den Pin 0 des Micro:bit.

Durch Drehen am Potentiometer wird der innere Widerstand verändert und der Pin0 liest laufend veränderte Spannungswerte zwischen 0 und 1023. Diese Werte sind keine echten Voltangaben, sondern teilen die maximalen (analogen) 3,3 Volt in 1024 (0 + 1023) digitale Einheiten. Dieser Vorgang wird durch einen im Micro:bit eingebauten Analog-Digital-Converter (ADC) bewerkstelligt. Im konkreten Fall ist dies ein 10-Bit-ADC mit 2^{10} Möglichkeiten.



Digitale Pins des Micro:bit kennen nur die Werte „1“ und „0“, analoge Pins können je nach verwendetem ADC numerische (bereits digitalisierte) Ganzzahlen einlesen (im konkreten Fall 0 bis 1023).

Es ist wichtig zu beachten, dass nicht jeder Pin am Micro:bit analoge Signale lesen kann. Bei einigen Pins ist ein direkter Wechsel zwischen digitalen und analogen Signalen möglich.

Beschreibung eines Potentiometers

Ein Potentiometer hat innerhalb einer Widerstandsbahn eine aufgerollte Widerstandsfläche. Der Stellregler ist mit einem Greifer („Schleifer“) verbunden, der die Widerstandsbahn einschränkt oder erweitert. Ist der Schleifer nahe dem Anfang der Widerstandsbahn, ist der Widerstandswert klein, da nur ein Bruchteil der gesamten Fläche verwendet wird. Je weiter der Schleifer zum anderen Ende der Widerstandsbahn wandert, desto größer wird der Widerstand.

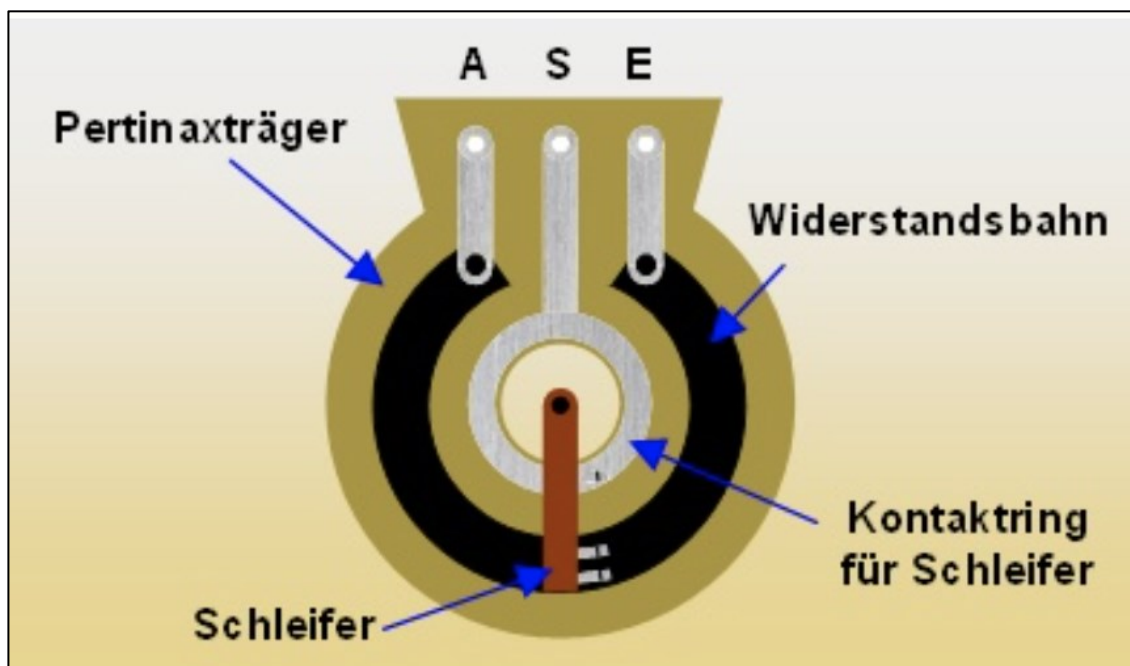
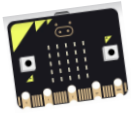


Abbildung 2: Aufbau eines Potentiometers⁴⁹

⁴⁹ Bildquelle unter <http://www.guitar-letter.de/Knowledge/Grundlagen/GrundlagendesPotentiometers.htm> (25.3.2024)



Analoge Eingänge, Analog-Digital-Converter ADC

Das analoge Signal vom Potentiometer wird direkt zum ADC geleitet, der im Mikrocontroller integriert ist. Der ADC wandelt das analoge Signal in ein digitales Signal um, indem er die Spannung in bestimmten Intervallen misst und diese Messungen in digitale Werte umrechnet, die der Mikrocontroller verarbeiten kann. Im Fall des Micro:bit wird eine Spannung zwischen 0 V und 3,3 V in digitale Werte von 0 bis 1023 umgewandelt, wobei diese Zahlen die Auflösung (das „Reservoir“) des ADCs darstellen (konkret sind das 10 Bit mit 2^{10} Möglichkeiten).

Intern wird das digitale Signal als eine Sequenz von Binärdaten (0 und 1) dargestellt, behandelt und gespeichert.

Extern, aus der Perspektive von Benutzer*innen, wird es als Ganzzahl (im Bereich von 0 bis 1023 für einen 10-Bit-ADC) präsentiert.

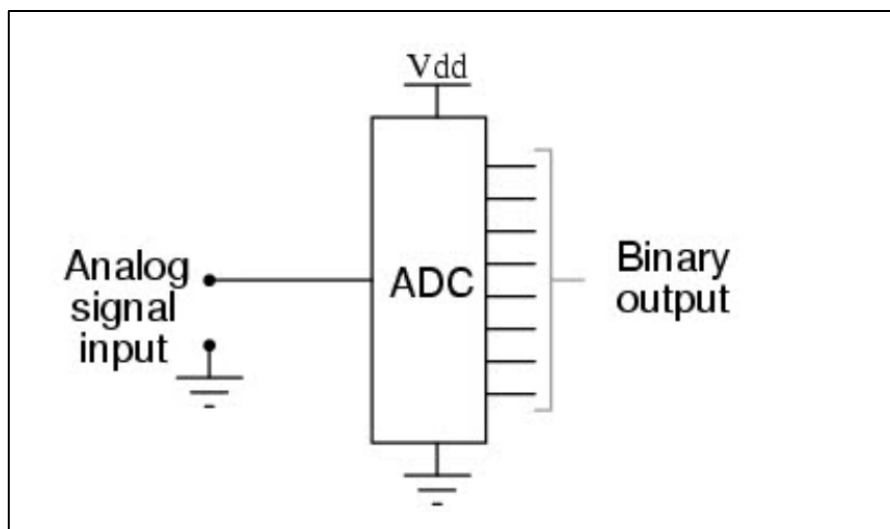
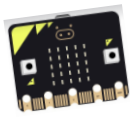


Abbildung 3: Umwandlung von analogen Werten zu binären Werten⁵⁰

⁵⁰ Quelle unter <https://www.elprocus.com/analog-to-digital-converter/> (abgerufen am 26.03.2024)



Schaltungsentwurf mit TinkerCAD

Optional kann mit oder ohne inkludierte, visuelle Programmiersprache das Funktionieren der Schaltung vorab getestet werden.

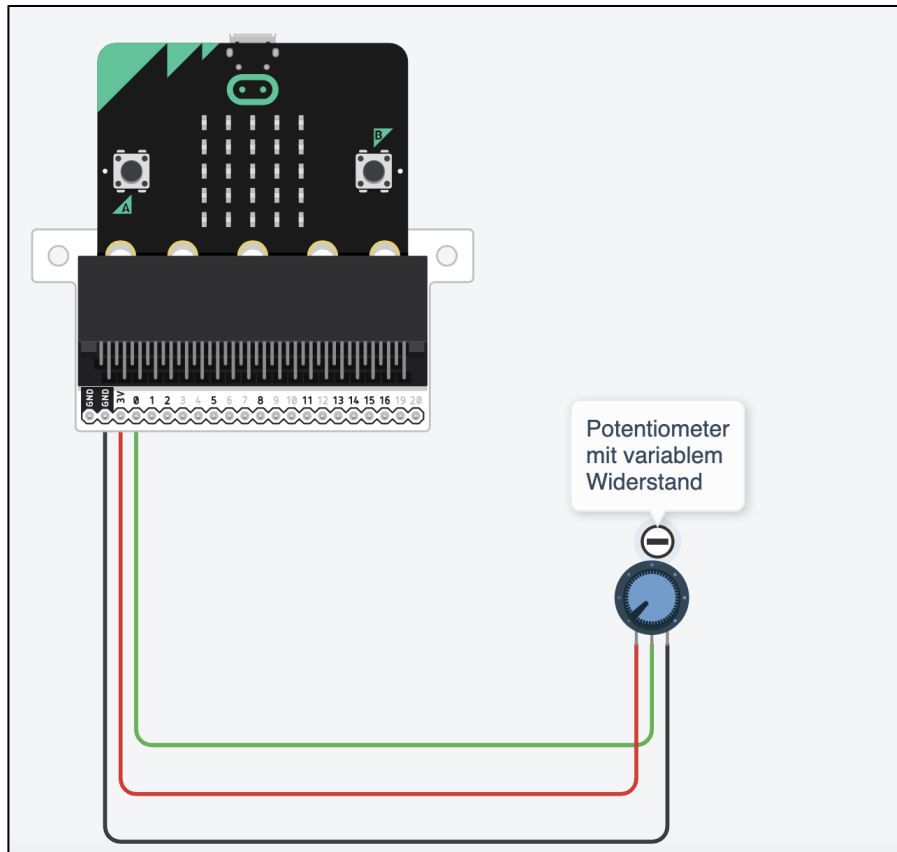


Abbildung 4: Schaltungsentwurf mit TinkerCAD

Programmierung grafisch mit MakeCode-Editor

Wichtig beim MakeCode-Editor ist die Auswahl der Gruppe PINS links. Eine eigentliche Read-Analog-Funktion ist nicht vorhanden, es muss eine Variable (hier *potWert*) mit der Eigenschaft „analoge Werte von Pin0“ erstellt und verbunden werden.

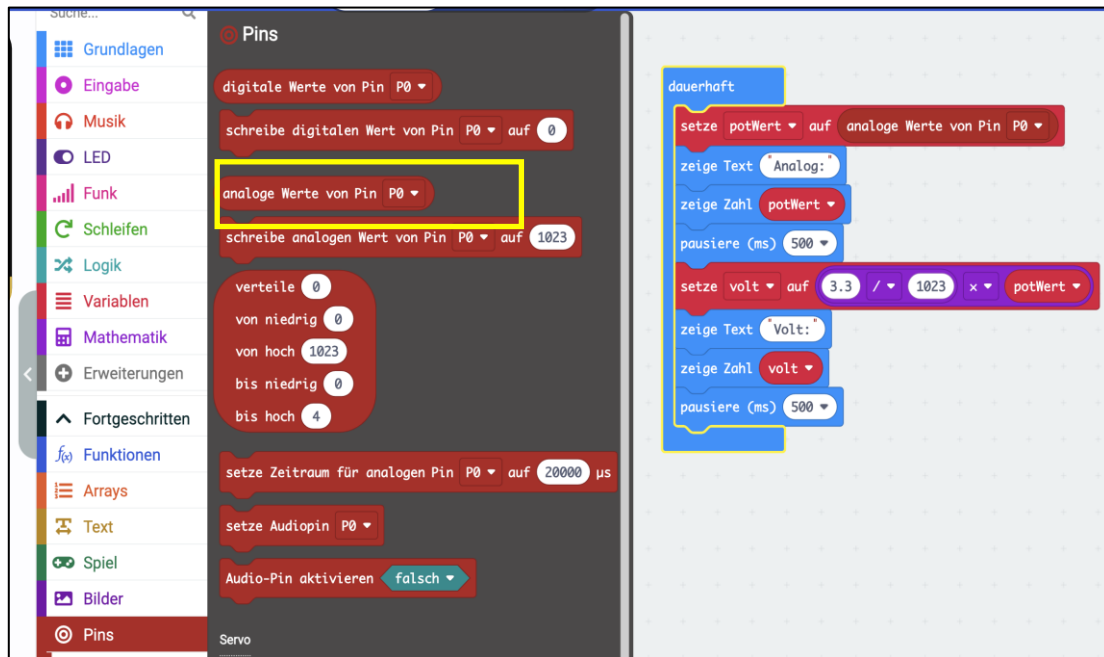
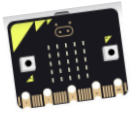
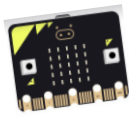


Abbildung 5 Grafische Programmierung mit MakeCode

Programmierung mit MicroPython Web bzw. Mu-Editor lokal

```
1  from microbit import *
2  # Variable potWert
3  potWert = 0
4  # Variable Volt
5  volt = 0
6
7
8  while True:
9      # analoges Einlesen der Spannung
10     potWert = pin0.read_analog()
11     display.show("Analog:" + str(potWert))
12     sleep(500)
13     # Umrechnung der ADC-Werte 0-1023 in Volt
14     volt = (3.3 / 1023) * potWert
15     # laufende Anzeige beim Drehen am Poti
16     display.show("Volt:" + str(volt))
17     sleep(500)
18
```

Abbildung 6: Programmierung im Python-Editor



The screenshot shows the Mu Editor window with a toolbar at the top containing icons for Modus, Neu, Laden, Speichern, Aufspielen, Dateien, and REPL. Below the toolbar is a tab labeled 'main34.py'. The code in the editor is as follows:

```
1 from microbit import *
2 # Variable potWert
3 potWert = 0
4 # Variable Volt
5 volt = 0
6
7
8 while True:
9     # analoges Einlesen der Spannung
10    potWert = pin0.read_analog()
11    display.show("Analog:" + str(potWert))
12    sleep(500)
13    # Umrechnung der ADC-Werte 0-1023 in Volt
14    volt = (3.3 / 1023) * potWert
15    # laufende Anzeige beim Drehen am Poti
16    display.show("Volt:" + str(volt))
17    sleep(500)
```

Abbildung 7: Gleicher Code im Mu-Editor

Zielgruppen

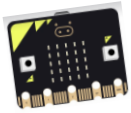
Lehrpersonen und Schüler*innen der SEK1 und SEK2

Ideen zur Erweiterung/Adaptierung/Vertiefung

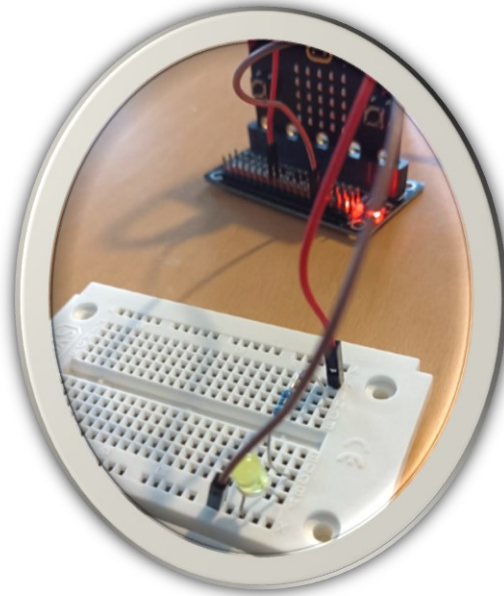
- Hinzufügen von Tönen in Abhängigkeit der Variable *potWert*
- Ausgabe der Messung über LCD- oder OLED-Bildschirme
- Verkabelung mit Krokodilklemmen

Ergänzende Lernunterlagen

- Micro:bit, Projekte und Erklärungen unter <https://microbit.org/> (25.03.2024)
- Grundlagen der Elektronik unter <https://studyflix.de/elektrotechnik/thema/elektrotechnik-grundlagen-20> (25.03.2024) bzw. <https://www.elektrotechnik-einfach.de/> (25.03.2024)
- Dokumentation für MicroPython für Micro:bit unter <https://microbit-micropython.readthedocs.io/en/v2-docs/> (24.03.2024)
- Grundlagen von Potentiometern unter https://www.elektronik-kompodium.de/sites/praxis/bauteil_potentiometer.htm (26.03.2024)
- Was ist ein ADC, ein DAC? unter <https://www.youtube.com/watch?v=jJxGobfDY3c> (27.03.2024)



13.6 Praxisprojekt 5 (PWM)



Projekt 5 – Pulsweiten-Modulation (PWM)

Benötigte Komponenten

- Micro:bit V2 mit Breakout-Board
- Starterkit GeeekPi BBC Micro:bit V2 Ultimate Starter Kit optional⁵¹
- Steckbrett
- 1 Light-Emitting-Diode (LED) rot
- 1 Widerstand mit 270 Ohm
- Jumperkabel "male-female"

Benötigte Software

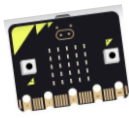
- TinkerCAD⁵²
- MakeCode-Editor von Microsoft⁵³
- Python-Editor für MicroPython⁵⁴

⁵¹ Siehe online unter https://www.amazon.de/GeeekPi-BBC-Micro-Ultimate-microbit/dp/B0BKXLM97?ref=ast_sto_dp (20.03.2024)

⁵² Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.03.2024)

⁵³ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁵⁴ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)



- Mu-Editor⁵⁵ lokal installiert

Generelle Vorsichtsmaßnahmen

- Niemals den 3,3-Volt-Anschluss direkt (ohne Vorwiderstand) mit Ground verbinden
- Bei elektronischen Steckübungen oder Veränderungen immer zuerst den Micro:bit vom Strom nehmen (von Batterie oder USB-Anschluss)
- Sich selbst vor der Arbeit mit Elektronik elektrostatisch entladen (Berühren von sonstigen metallischen Teilen wie Heizkörpern, Metalltischen etc.)

Projektbeschreibung

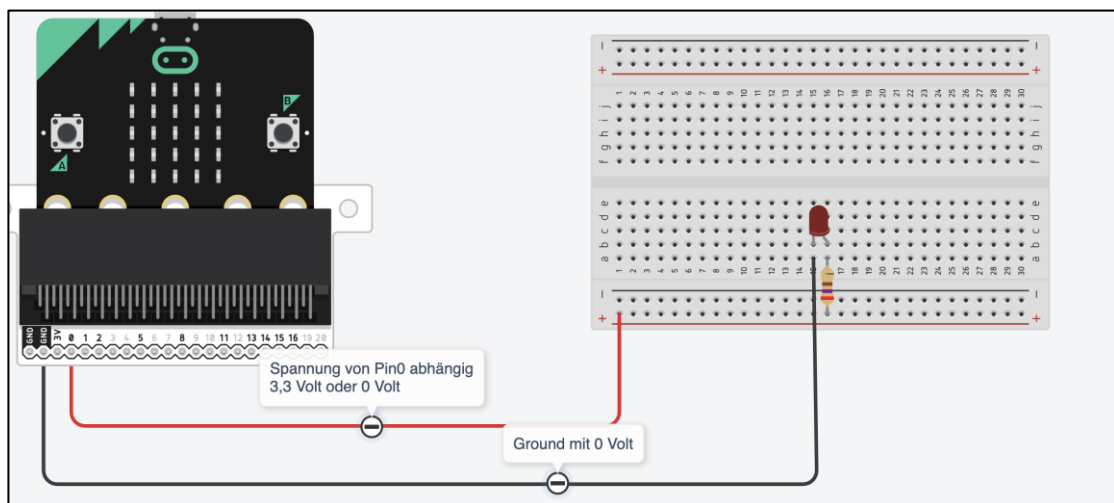
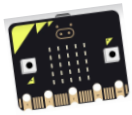


Abbildung 1: LED über Pin0 in der Helligkeit steuern

Die Abbildung 1 zeigt eine reine Reihenschaltung, deren Stromversorgung über einen schaltbaren Pin 0 erfolgt. Der Zweck des Projektes besteht darin, die Helligkeit der LED stufenweise zu erhöhen und anschließend wieder zu verringern. Diese Steuerung erfolgt mit der sogenannten Pulsweiten-Modulation (PWM).

Ein Vorwiderstand von 270 Ohm und eine LED sind in Reihe (hintereinander) geschaltet und der Strom fließt durch beide Komponenten zum Ground. Die Gesamtspannung von 3,3 Volt wird zuerst beim Vorwiderstand auf 2 Volt reduziert und fließt dann durch die LED als Verbraucher. Die verbrauchte Energie wird als Licht der LED abgegeben.

⁵⁵ Downloadbar unter <https://codewith.mu/en/download> (25.03.2024)



Die Anordnung von Vorwiderstand und LED könnte auch umgekehrt sein. Wichtig ist jedoch die Stellung der LED. Der Strom (die Elektronen) fließt zuerst durch die Anode (langes Bein der LED) und dann durch die Kathode (kurzes Bein). Bei umgekehrter Stellung würde der Stromfluss unterbrochen werden.

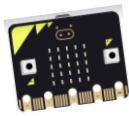
Merkhilfe: „*Nach dem Stromfluss verliert die LED einen Teil des Beins*“

Datenblatt

Damit ein LED beleuchtet werden kann, muss immer zuvor ein Datenblatt⁵⁶ der LED studiert werden (siehe Abbildung 2). Wichtig hierbei sind bei den LED-Typen rot, grün und gelb die **Durchlassspannung** und die **Stromstärke** durch die LED, in unserem Beispiel 2 V und 10 mA lt. Datenblatt. Gefährlich sind die Begriffe „**Max. zulässiger Strom**“ und **Durchbruchspannung**. Vor allem der maximal zulässige Strom von 30 mA ist bei den obigen Schaltungen kein Thema, da konkret beim Microbit mit 5 mA⁵⁷ gearbeitet wird. Die Durchbruchspannung von 5V würde auf alle Fälle die LED zerstören, wird aber natürlich nie verwendet.

⁵⁶ Musterdatenblatt unter Reichelt, https://cdn-reichelt.de/documents/datenblatt/C200/DS_GER_RF19.pdf (03.01.2024)

⁵⁷ Siehe Empfehlung von 5 mA für Micro:bit unter <https://support.microbit.org/support/solutions/articles/19000101863-connecting-an-led-to-the-micro-bit> (25.03.2024)



Technische Daten LED

3 mm LED

(Kenndaten gültig bei 25 °C)	LED rot	LED grün	LED gelb
Max. zulässiger Strom I_F :	30 mA	30 mA	20 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,5 mA/°C	ca. 0,5 mA/°C	ca. 0,2 mA/°C
Typ. Wellenlänge:	635 nm	565 nm	586 nm
Typ. Durchlassspannung U_F bei I_F :	2 V/10 mA	2 V/10 mA	2 V/10 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 μ A	min. 5 V/100 μ A	min. 5 V/100 μ A
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C
	LED blau	LED weiß	LED grün superhell
Max. zulässiger Strom I_F :	20 mA	20 mA	30 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,6 mA/°C	-	-
Typ. Wellenlänge:	470 nm	-	510-545 nm
Typ. Durchlassspannung U_F bei I_F :	2,7 V/10 mA	3,6 V/20 mA	3,5 V/20 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 μ A	-	-
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	-30 °C ... + 100 °C

Abbildung 2: Datenblatt LED

Vorwiderstand berechnen

Für die Schaltung muss eine empfohlene Durchlassspannung von 2 V erreicht werden. Somit muss ein geeigneter Vorwiderstand geschaltet werden, der die Ausgangsspannung von 3,3 V auf 2 V des LEDs senkt, eine Differenz von 1,3 Volt.

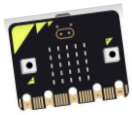
$$\text{Widerstand} = \frac{3,3V - 2V = 1,3V}{0,005\text{Ampere}(5mA)}$$

Dies ergibt einen Vorwiderstandswert von 260 Ω (Ohm). Praktisch wird aber der nächsthöhere Widerstand, in diesem Fall 270 Ω , verwendet. In der Praxis ist der errechnete Widerstand als Bauteil selten exakt verfügbar.

Korrekten Widerstand verwenden

Physische Widerstände sind durch einen speziellen Farbcode gekennzeichnet. In der häufigsten Form handelt es sich um 4 oder 5 Ringe, die zur Bestimmung des Widerstands beachtet werden müssen. Eine genau Berechnung erledigen Farbtabellen oder Online-Rechner⁵⁸. Beachte beim Lesen der Ringe die Anordnungs-Richtung. Der von

⁵⁸ Digikey, online unter <https://www.digikey.de/de/resources/conversion-calculators/conversion-calculator-resistor-color-code> (03.01.2024)



den Betrags-Ringen weiter entfernte Toleranz-Ring, zeigt das Ende der Leserichtung. Wichtig ist die Berücksichtigung der Anzahl an tatsächlich vorhandenen Ringen am Widerstand.

Anzahl der Bänder
4 Ringe 5 Ringe 6 Ringe

Widerstandsparameter

1. Farbring
Rot 2

2. Farbring
Violett 7

3. Farbring
Schwarz 0

Multiplikator
Schwarz $\times 1 \Omega$

Toleranz
Braun $\pm 1\%$

Resistance value
270 Ω

Ausgabe

Widerstandswert:
270 Ohms 1%

Abbildung 3: Farbdarstellung des Vorwiderstands von 270 Ohm

LED richtig stecken

LED bestehen aus Anode (langes Bein) und Kathode (kurzes Bein). Zu beachten ist die Richtung des Stromflusses, da dieser zuerst durch die Anode (langes Bein) und dann erst durch die Kathode fließt. Wird die LED falsch gesteckt, findet KEIN Stromfluss statt. LEDs leiten somit Strom in die richtige Richtung, dienen aber auch als optische Hinweise (z.B. rotes LED als Fehlermeldung).

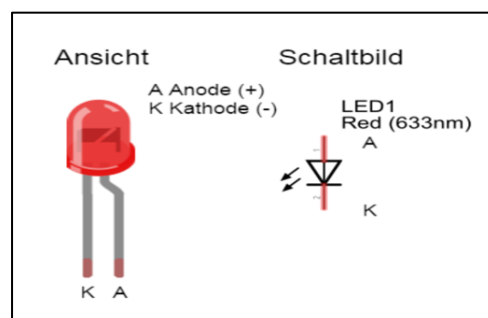
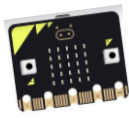


Abbildung 4: Stromfluss von Anode+ zu Kathode-(Quelle⁵⁹)

⁵⁹ Elab, LED, https://robotfreak.de/elab-wiki/index.php?title=LED_Leuchtdiode (03.01.2024)



Pulsweiten-Modulation (PWM)

Der Micro:bit verfügt über sowohl digitale als auch analoge PINs. Er ist in der Lage, sowohl analoge als auch digitale Eingangssignale zu erfassen. Im Ausgang verfügt der Micro:bit jedoch ausschließlich über digitale Pins, jedoch fehlen analoge PINs. Analoge Pins würden ein digitales Ausgangssignal in einen beliebigen, kontinuierlichen Spannungswert umwandeln. Dies würde der sogenannte Digital-Analog-Converter (DAC) übernehmen, der jedoch bei Micro:bit-Pins nicht vorhanden ist. Stattdessen behilft man sich mit der rein digitalen Technik der Pulsweiten-Modulation. Dazu müssen folgende Begriffe erklärt werden.

CPU-Takt Micro:bit: 64 MHz = 64.000.000 Herz/Sekunde = 15,629 Nano-Sekunden

PWM-Periode: Zeitdauer der kleinsten PWM-Dauer. Beim Mikro:bit beträgt diese eine Millisekunde (1 ms) bis zu einigen Mikrosekunden (μs)⁶⁰

PWM-Tastverhältnis (Duty Cycle): Innerhalb einer PWM-Periode sind Low und High in einem bestimmten Verhältnis. Es gibt aber nur zwei Zustände.

Pulsweite: Summe aus High und Low innerhalb einer PWM-Periode.

PWM hängt nicht direkt vom CPU-Takt ab, CPU-Takt ist aber Ausgangstakt. PWM softwareseitig anpassbar.

⁶⁰ Siehe online unter <https://www.i-programmer.info/programming/148-hardware/15333-microbit-pwm.html> (abgerufen am 26.03.2024)

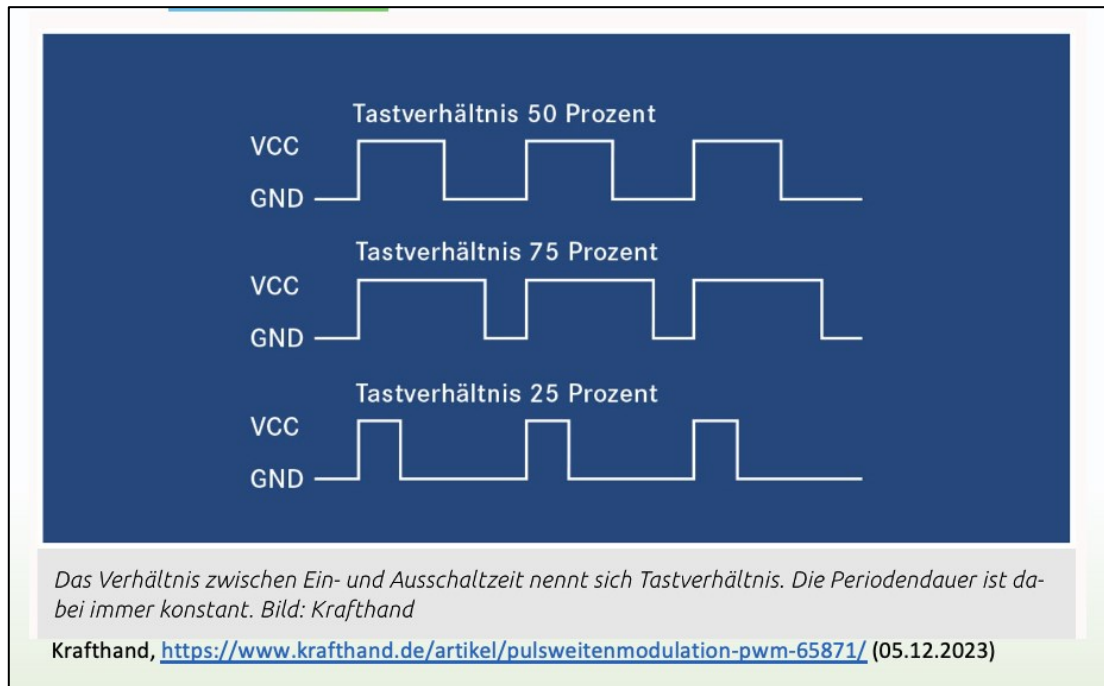
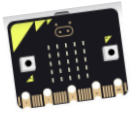


Abbildung 5: Periodendauer und Tastverhältnis von PWM

Statt einer Angabe in Prozent wie in obiger Erklärung können auch PWM-Werte von 0 bis 1023 verwendet werden. Dabei entspricht der Wert 1023 dem Verhältnis 100 %, der Wert 511 wäre ca. 50 % des oben genannten Tastverhältnisses. Je näher also der Wert zu 1023 ist, desto „heller“ leuchtet die LED.

Wichtig ist es zu verstehen, dass bei PWM KEINE Umwandlung in analoge Voltwerte erfolgt, sondern dass das Signal selbst „1“ oder „0“ bleibt. Nur das Verhältnis dieser beiden Werte wird durch die Werte 0 bis 1023 in einer PWM-Periode angepasst und für einige Zeit ausgeführt. Dieser (rein digitale) Effekt erzeugt nun eine Veränderung der Helligkeit der LED.

Schaltungsentwurf mit TinkerCAD

Optional kann mit oder ohne inkludierter, visueller Programmiersprache das Funktionieren der Schaltung vorab getestet werden.

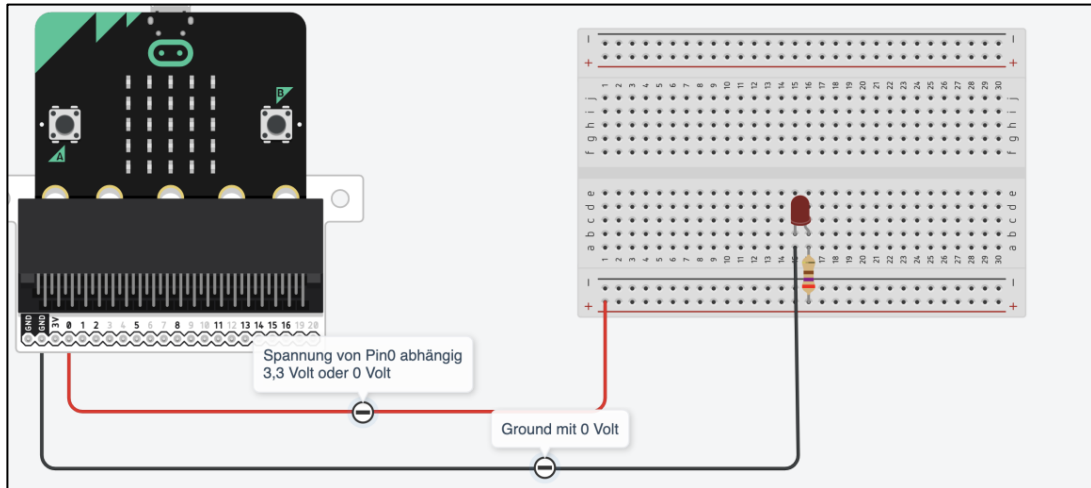
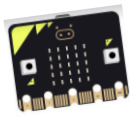


Abbildung 6: Schaltungsentwurf mit TinkerCAD

Programmierung grafisch mit MakeCode-Editor

Wichtig beim MakeCode-Editor ist die Auswahl der Gruppe PINS links. Hier befinden sich die wichtigsten Funktionen für Pins, konkret das „analoge Schreiben“ im Ausgang.

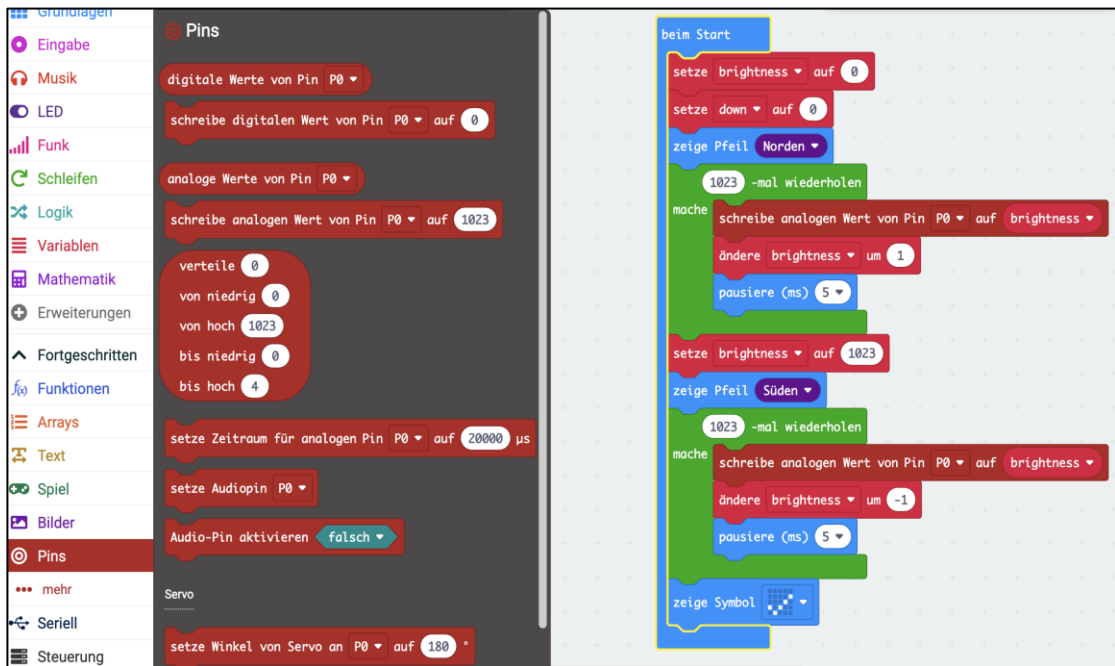
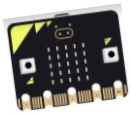


Abbildung 7: Grafische Programmierung mit MakeCode

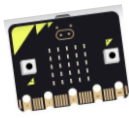


Programmierung mit MicroPython Web bzw. Mu-Editor lokal

Im Vergleich zur grafischen Lösung wird hier der PWM-Modus 5-mal durchgeführt, nicht nur beim Start (siehe Lösung mit MakeCode). Nach dem hellsten und dunkelsten Punkt wird eine kurze Pause von 2 Sekunden eingebaut. Wichtig ist auch der Befehl `sleep(5)`, da hier nur der Programmiercode ruht, das PWM-Signal aber in dieser Zeitspanne weiter erzeugt wird und dadurch der „Leuchteffekt“ besser dargestellt werden kann.

```
1  from microbit import *
2
3  # Schleife mit 5 Durchgängen
4  for x in range(5):
5      # brightness stellt das Tastverhältnis von PWM dar
6      # bei niedrigen Werten dominieren LOW, bei hohen Werten HIGH
7      # LED wird langsam heller
8
9      display.show(Image.HAPPY)
10     # Hochzählen, LED wird heller
11     for brightness in range(0, 1023):
12         pin1.write_analog(brightness)
13         # Achtung: bei sleep wird nur der Code unterbrochen, nicht die Ausführung
14         # somit geniert sleep(5): 5 Millisekunden/1 Mikrosekunde = 5000 Perioden
15         sleep(5)
16     # Pause vor dem Wechsel
17     sleep(2000)
18     # LED wird langsam dunkler
19     display.show(Image.SAD)
20     for brightness in range(1023, 0, -1):
21         pin1.write_analog(brightness)
22         # 5 millisekunden für Generierung der Perioden
23         sleep(5)
24     # Pause vor dem Wechsel
25     sleep(2000)
26
27 # Ende der Lichtshow
28 display.show(Image.ASLEEP)
29 sleep(2000)
30 display.clear()
31
32
```

Abbildung 8: Programmierung im Python-Editor



The screenshot shows the Mu-Editor window with a toolbar at the top containing icons for Modus, Neu, Laden, Speichern, Aufspielen, Dateien, REPL, Plotter, Hineinzoomen, and Raus. The main area displays a Python script named 'main40.py' with the following code:

```
1 from microbit import *
2
3 # Schleife mit 5 Durchgängen
4 for x in range(5):
5     # brightness stellt das Tastverhältnis von PWM dar
6     # bei niedrigen Werten dominieren LOW, bei hohen Werten HIGH
7     # LED wird langsam heller
8
9     display.show(Image.HAPPY)
10    # Hochzählen, LED wird heller
11    for brightness in range(0, 1023):
12        .....
13        pin1.write_analog(brightness)
14        # Achtung: bei sleep wird nur der Code unterbrochen, nicht die Ausführung
15        # somit geniert sleep(5): 5 Millisekunden/1 Mikrosekunde = 5000 Perioden
16        sleep(5)
17    # Pause vor dem Wechsel
18    sleep(2000)
19    # LED wird langsam dunkler
20    display.show(Image.SAD)
21    for brightness in range(1023, 0, -1):
22        .....
23        pin1.write_analog(brightness)
24        # 5 millisekunden für Generierung der Perioden
25        sleep(5)
26    # Pause vor dem Wechsel
27    sleep(2000)
28 # Ende der Lichtshow
29 display.show(Image.ASLEEP)
30 sleep(2000)
31 display.clear()
```

Abbildung 9: Gleicher Code im Mu-Editor

Zielgruppen

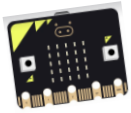
Lehrpersonen und Schüler*innen der SEK1 und SEK2

Ideen zur Erweiterung/Adaptierung/Vertiefung

- Verändern des Tastverhältnisses auf Werte von 0 bis 255
- Wenn Zustand sehr hell, Geräusche einbauen
- Besseres Visualisieren der PWM-Perioden über die Anweisung `sleep(5)`
- Zeitgleicher Aufbau einer Schaltung mit und ohne PWM um den Unterschied erfahrbar zu machen.

Ergänzende Lernunterlagen

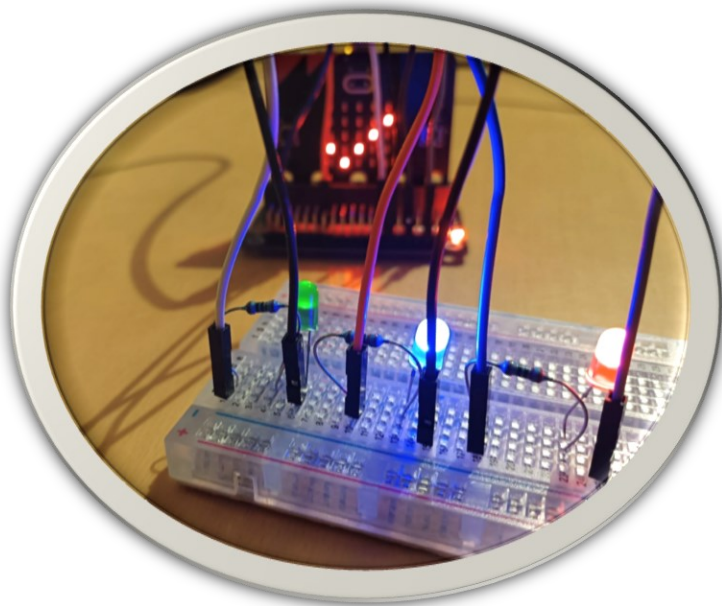
- Micro:bit, Projekte und Erklärungen unter <https://microbit.org/> (25.03.2024)
- Grundlagen der Elektronik unter <https://studyflix.de/elektrotechnik/thema/elektrotechnik-grundlagen-20> (25.03.2024) bzw. <https://www.elektrotechnik-einfach.de/> (25.03.2024)



- Dokumentation für MicroPython für Micro:bit unter <https://microbit-micropython.readthedocs.io/en/v2-docs/> (24.03.2024)
- Pulsweitenmodulation unter <https://www.elektronik-kompodium.de/sites/kom/0401111.htm> (27.03.2024)

13.7 Praxisprojekt 6 (Binärschaltung)

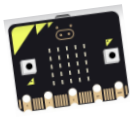
Projekt 6 – Binärschaltung



Benötigte Komponenten

- Micro:bit V2 mit Breakout-Board
- Starterkit GeeekPi BBC Micro:bit V2 Ultimate Starter Kit optional⁶¹
- Steckbrett
- 3 Light-Emitting-Diode (LED) rot, grün, blau
- 3 Widerstände mit je ca. 270 Ohm
- Jumperkabel “male-female” und ev. “male-male”

⁶¹ Siehe online unter https://www.amazon.de/GeeekPi-BBC-Micro-Ultimate-microbit/dp/B0BKXLM97?ref=ast_sto_dp (20.03.2024)



Benötigte Software

- TinkerCAD⁶²
- MakeCode-Editor von Microsoft⁶³
- Python-Editor für MicroPython⁶⁴
- Mu-Editor⁶⁵ lokal installiert

Generelle Vorsichtsmaßnahmen

- Niemals den 3,3-Volt-Anschluss direkt (ohne Vorwiderstand) mit Ground verbinden
- Bei elektronischen Steckübungen oder Veränderungen immer zuerst den Micro:bit vom Strom nehmen (von Batterie oder USB-Anschluss)
- Sich selbst vor der Arbeit mit Elektronik elektrostatisch entladen (Berühren von metallischen Teilen wie Heizkörpern, Metalltischen etc.)

Projektbeschreibung

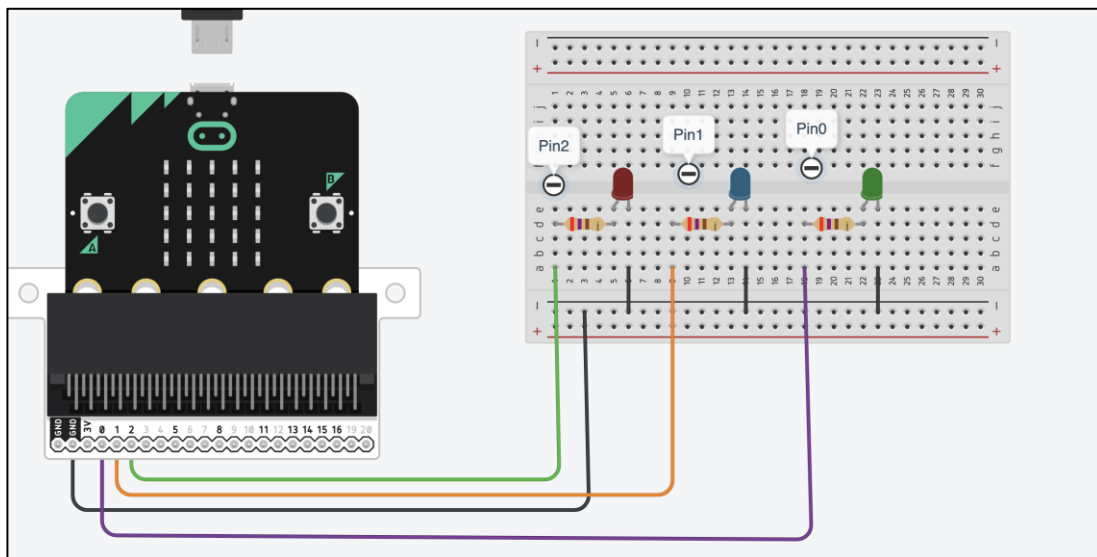


Abbildung 1: Mit LEDs über Pin 0, Pin 1 und Pin 2 Binärzahlen darstellen

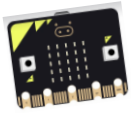
Die Abbildung 1 zeigt drei reine Reihenschaltungen, deren Stromversorgung über die schaltbaren Pin 2, Pin 1 und Pin 0 erfolgt. Durch wechselhaftes Ein- und Ausschalten der Pins wird eine digitale Darstellung der Zahlen 0 bis 7 erreicht, indem die entsprechenden Stellenwerte statt einer „1“ als leuchtende LED erscheinen.

⁶² Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.03.2024)

⁶³ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁶⁴ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁶⁵ Downloadbar unter <https://codewith.mu/en/download> (25.03.2024)



Ein Vorwiderstand von 270 Ohm und eine LED sind in Reihe geschaltet und der Strom fließt durch beide Komponenten zum Ground. Die Gesamtspannung von 3,3 Volt wird zuerst beim Vorwiderstand auf 2 Volt reduziert und fließt dann durch die LED als Verbraucher. Die verbrauchte Energie wird als Licht der LED abgegeben. Die Anordnung von Vorwiderstand und LED könnte auch umgekehrt sein. Wichtig ist jedoch die Stellung der LED. Der Strom (die Elektronen) fließt zuerst durch die Anode (langes Bein der LED) und dann durch die Kathode (kurzes Bein). Bei umgekehrter Stellung würde der Stromfluss unterbrochen werden.

Merkhilfe: „Nach dem Stromfluss verliert die LED einen Teil des Beins“

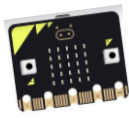
Die drei LEDs sind nicht dauerhaft an der Stromversorgung angeschlossen, sondern erhalten ihre Spannung über den Pin 2, Pin 1 und Pin 0. Durch Betätigen des Button A am MB wird eine Zahl numerisch am Display dargestellt (z.B. „3“). Die binäre Darstellung wäre „011“, was durch die Anordnung der Pins dargestellt ist. Bei einer „1“ leuchtet die LED des betreffenden Pins, hier wären es Pin 1 und Pin 0, die LED bei Pin 2 bliebe dunkel.

Datenblatt

Damit eine LED beleuchtet werden kann, muss immer zuvor ein Datenblatt⁶⁶ der LEDs studiert werden (siehe Abbildung 2). Wichtig hierbei sind bei den LED-Typen rot, grün und gelb die **Durchlassspannung** und die **Stromstärke** durch die LED, in unserem Beispiel 2 V und 10 mA lt. Datenblatt. Gefährlich sind die Begriffe „**Max. zulässiger Strom**“ und **Durchbruchspannung**. Vor allem der maximal zulässige Strom von 30 mA ist bei den obigen Schaltungen kein Thema, da konkret beim Micro:bit mit 5 mA⁶⁷ gearbeitet wird. Die Durchbruchspannung von 5V würde auf alle Fälle das LED zerstören, wird aber natürlich nie verwendet.

⁶⁶ Musterdatenblatt unter Reichelt, https://cdn-reichelt.de/documents/datenblatt/C200/DS_GER_RF19.pdf (abgerufen am 3.1.2024)

⁶⁷ Siehe Empfehlung von 5 mA für Micro:bit unter <https://support.microbit.org/support/solutions/articles/19000101863-connecting-an-led-to-the-micro-bit> (25.03.2024)



Technische Daten LED

3 mm LED

(Kenndaten gültig bei 25 °C)	LED rot	LED grün	LED gelb
Max. zulässiger Strom I_F :	30 mA	30 mA	20 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,5 mA/°C	ca. 0,5 mA/°C	ca. 0,2 mA/°C
Typ. Wellenlänge:	635 nm	565 nm	586 nm
Typ. Durchlassspannung U_F bei I_F :	2 V/10 mA	2 V/10 mA	2 V/10 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 µA	min. 5 V/100 µA	min. 5 V/100 µA
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C
	LED blau	LED weiß	LED grün superhell
Max. zulässiger Strom I_F :	20 mA	25 mA	30 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,6 mA/°C	-	-
Typ. Wellenlänge:	470 nm	-	510-545 nm
Typ. Durchlassspannung U_F bei I_F :	2,7 V/10 mA	3,6 V/20 mA	3,5 V/20 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 µA	-	-
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	-30 °C ... + 100 °C

Abbildung 2 Datenblatt LED

Vorwiderstand berechnen

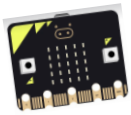
Für alle Schaltungen muss eine empfohlene Durchlassspannung von 2 V erreicht werden. Somit muss ein geeigneter Vorwiderstand geschaltet werden, der die Ausgangsspannung von 3,3 V auf 2 V des LEDs senkt, für eine Differenz von 1,3 Volt.

$$\text{Widerstand} = \frac{3,3V - 2V = 1,3V}{0,005\text{Ampere}(5mA)}$$

Dies ergibt einen Vorwiderstandswert von 260 Ω (Ohm). Den berechneten Widerstand findet man in der Praxis selten in exakt dieser Form als Bauelement.

Korrekten Widerstand verwenden

Physische Widerstände sind durch einen speziellen Farbcode gekennzeichnet. In der häufigsten Form handelt es sich um 4 oder 5 Ringe, die zur Bestimmung des Widerstands beachtet werden müssen. Eine genau Berechnung erledigen Farbtabelle oder



Online-Rechner⁶⁸. Beachte beim Lesen der Ringe die Anordnungs-Richtung. Der von den Betrags-Ringen weiter entfernte Toleranz-Ring, zeigt das Ende der Leserichtung. Wichtig ist die Berücksichtigung der Anzahl an tatsächlich vorhandenen Ringen am Widerstand.

Anzahl der Bänder
4 Ringe **5 Ringe** 6 Ringe

Widerstandsparameter

1. Farbring
Rot 2

2. Farbring
Violett 7

3. Farbring
Schwarz 0

Multiplikator
Schwarz $\times 1 \Omega$

Toleranz
Braun $\pm 1\%$

Resistance value
270 Ω

Ausgabe

Widerstandswert:
270 Ohms 1%

Abbildung 4: Farbdarstellung eines Vorwiderstands von 270 Ohm

LED richtig stecken

LED bestehen aus Anode (langes Bein) und Kathode (kurzes Bein). Zu beachten ist die Richtung des Stromflusses, da dieser zuerst durch die Anode (langes Bein) und dann erst durch die Kathode fließt. Wird die LED falsch gesteckt, findet KEIN Stromfluss statt. LEDs leiten somit Strom in die richtige Richtung, dienen aber auch als optische Hinweise (z.B. rotes LED als Fehlermeldung).

⁶⁸ Digikey, online unter <https://www.digikey.de/de/resources/conversion-calculators/conversion-calculator-resistor-color-code> (03.01.2024)

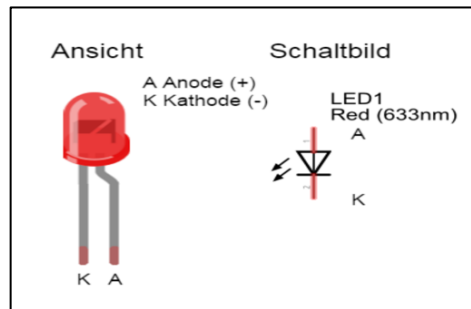
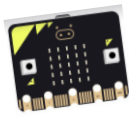


Abbildung 5: Stromfluss von Anode+ zu Kathode-(Quelle⁶⁹)

Binäre Darstellung von Zahlen

Bei Dezimalzahlen erfolgt die Darstellung einer Zahl mit Zehnerpotenzen mit einem gewissen Stellenwert, Beispiel 123.

1	2	3	
$(1 \times 10^2) + (2 \times 10^1) + (3 \times 10^0) = 123$			

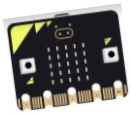
In der binären Digitaltechnik gibt es nur zwei Möglichkeiten („0“ oder „1“). Auch hier gibt es aber Stellenwerte und Potenzen, z.B. „101“ wäre dezimal der Wert 5

1	0	1	
$(1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 5$			

Das Projekt 6 stellt genau diese Visualisierung dar, über leuchtende LEDs bei bestimmten PINs.

1	0	1	
$(1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 5$			
Pin 2 (ein) + Pin 1 (aus) + Pin0 (ein) = 5			

⁶⁹ Elab, LED, https://robotfreak.de/elab-wiki/index.php?title=LED_Leuchtdiode (03.01.2024)



Schaltungsentwurf mit TinkerCAD

Optional kann mit oder ohne inkludierter, visueller Programmiersprache das Funktionieren der Schaltung vorab getestet werden.

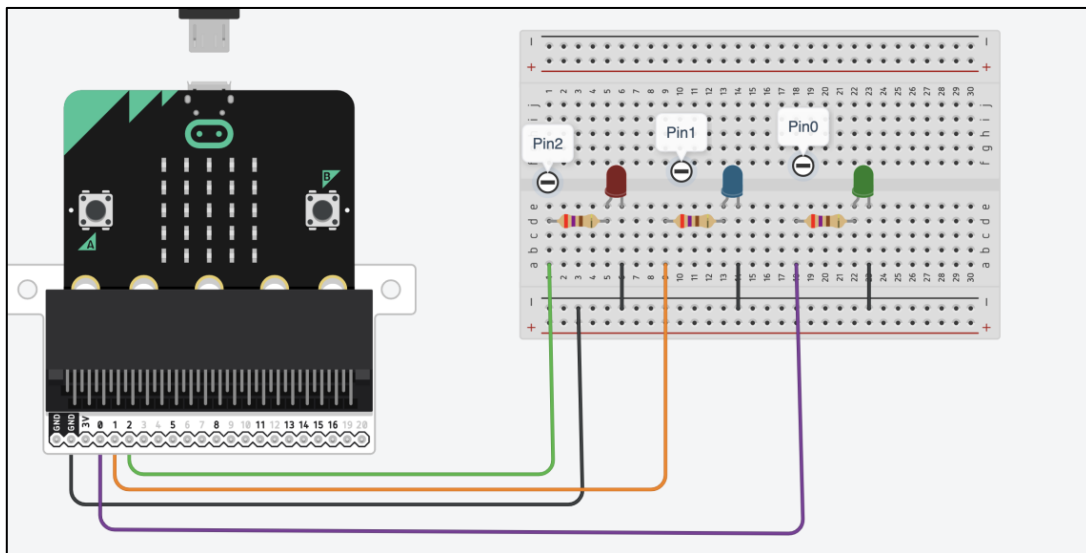


Abbildung 6: Schaltungsentwurf mit TinkerCAD

Programmierung grafisch mit MakeCode-Editor

Das MakeCode-Beispiel ist etwas einfacher gestaltet. Es gibt zwei Arrays, die genau 8 numerische und die dazugehörenden 8 binären Werte enthalten. Beim Drücken von Button A wird ein zufälliger Wert aus dem Array *list* generiert, der dann die gleichlautende Binärdarstellung aus dem Array *digList* aufruft. Aus der dreistelligen Darstellung der Binärzahl (z.B. „101“) wird von links nach rechts jede Ziffer in Pin 2, Pin 1 und Pin 0 geschrieben. Bei einer „1“ leuchtet die LED, bei „0“ nicht. Wichtig ist, dass bei MakeCode der Datentyp *TEXT* bei den Variablen *digiwert* und *digList* über Text zugeteilt werden muss.

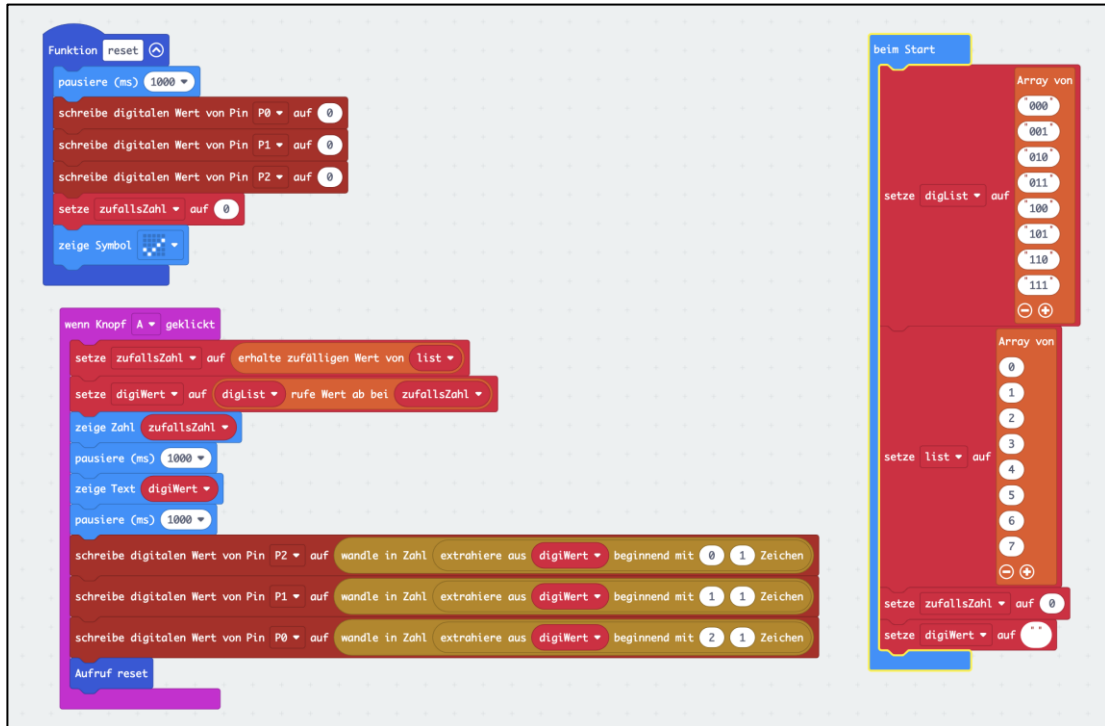
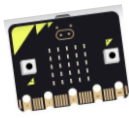


Abbildung 7: Grafische Programmierung mit MakeCode

Programmierung mit MicroPython Web bzw. Mu-Editor lokal

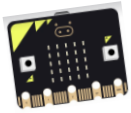
Im Gegensatz zur visuellen Programmierung in MicroPython erfolgt die Logik so, dass durch Betätigung von Button A die Zählung von 0 bis 7 inkrementiert wird. Angezeigt werden die Zahlen selbst und deren Binärdarstellung durch das Schalten der PINs. Es ist also nicht eine einzige Zufallszahl wie bei MakeCode, sondern es werden aufsteigend alle 8 Zustände (0 bis 7) dargestellt.

Ebenso setzt sich der Code aus zwei For-Schleifen zusammen: einer äußeren Schleife, die alle 8 Möglichkeiten abdeckt, und einer inneren Schleife, die den dreistelligen binären Code extrahiert und ihn an die PINs übermittelt.

Die beiden wichtigsten Befehle sind:

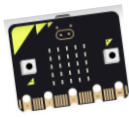
```
binaerWert = '{:03b}'.format(zaehler) # wandelt eine Ganzzahl in einen dreistelligen Binärcode
```

```
pinRow[i].write_digital(int(element)) # Eine Stelle des Binärcodes wird an den PIN übergeben
```



```
1  from microbit import *
2  # Anordnung der Pins von links nach rechts
3  pinRow = [pin2, pin1, pin0]
4
5  while True:
6      if button_a.is_pressed():
7          # zaehler von 0 bis 7
8          zaehler = 0
9          for i in range(8):# inklusive 0 bis 7
10             # Anzeigen des aktuellen numerischen Werts von zaehler
11             display.scroll(str(zaehler))
12             # Konvertieren von 'zaehler' in einen 4-stelligen String
13             # format-Funktion :03b erzwingt eine dreistellige Binärdarstellung
14             binaerWert = '{:03b}'.format(zaehler) # binaerWert ist ein String
15             # Schleife wird pro Zahl 3 mal durchlaufen von 0 bis 2
16             for i in range(3):
17                 # Zugriff auf das i-te Bit des Binärstrings
18                 element = binaerWert[i] # ist ein String
19                 # Schreiben der Werte 1 und 0 in die 3 Pins
20                 pinRow[i].write_digital(int(element))# Element wird int-Zahl
21             # nächste Zahl von 0 bis 7
22             zaehler += 1
23             # Warten für 1,5 Sekunde für nächste Zahlendarstellung
24             sleep(1500)
25             # OK Zeichen nach der Zahl 7
26             display.show(Image.YES)
```

Abbildung 8: Programmierung im Python-Editor



```
1 from microbit import *
2 # Anordnung der Pins von links nach rechts
3 pinRow = [pin2, pin1, pin0]
4
5 while True:
6     if button_a.is_pressed():
7         # zaehler von 0 bis 7
8         zaehler = 0
9         for i in range(8): # inklusive 0 bis 7
10             # Anzeigen des aktuellen numerischen Werts von zaehler
11             display.scroll(str(zaehler))
12             # Konvertieren von 'zaehler' in einen 4-stelligen String
13             # format-Funktion :03b erzwingt eine dreistellige Binärdarstellung
14             binaerWert = '{:03b}'.format(zaehler) # binaerWert ist ein String
15             # Schleife wird pro Zahl 3 mal durchlaufen von 0 bis 2
16             for i in range(3):
17                 # Zugriff auf das i-te Bit des Binärstrings
18                 element = binaerWert[i] # ist ein String
19                 # Schreiben der Werte 1 und 0 in die 3 Pins
20                 pinRow[i].write_digital(int(element)) # Element wird int-Zahl
21                 # nächste Zahl von 0 bis 7
22                 zaehler += 1
23             # Warten für 1,5 Sekunde für nächste Zahlendarstellung
24             sleep(1500)
25             # OK Zeichen nach der Zahl 7
26             display.show(Image.YES)
```

Abbildung 9: Gleicher Code im Mu-Editor

Zielgruppen

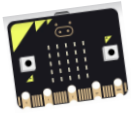
Lehrpersonen und Schüler*innen der SEK1 (Schüler*innen eventuell) und SEK2

Ideen zur Erweiterung/Adaptierung/Vertiefung

- Von einer 3-Bit-Darstellung zur 4-Bit-Darstellung erweitern
- Mit Button B ev. alle Pins auf „0“ setzen als eigene Funktion

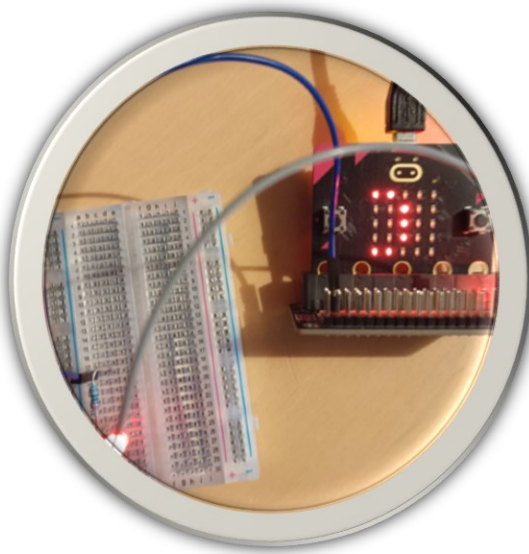
Ergänzende Lernunterlagen

- Micro:bit, Projekte und Erklärungen unter <https://microbit.org/> (25.03.2024)
- Grundlagen der Elektronik unter <https://studyflix.de/elektrotechnik/thema/elektrotechnik-grundlagen-20> (25.03.2024) bzw. <https://www.elektrotechnik-einfach.de/> (25.03.2024)
- Dokumentation für MicroPython für Micro:bit unter <https://microbit-micropython.readthedocs.io/en/v2-docs/> (24.03.2024)



- Binärdarstellung in Studyflix unter <https://studyflix.de/mathematik/binaersystem-5007> (27.03.2024)

13.8 Praxisprojekt 7 (Interner Lichtsensor)

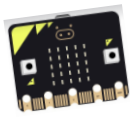


Projekt 7 – Interner Lichtsensor mit LED

Benötigte Komponenten

- Micro:bit V2 mit Breakout-Board
- Starterkit GeeekPi BBC Micro:bit V2 Ultimate Starter Kit optional⁷⁰
- Steckbrett
- 1 Light-Emitting-Diode (LED) rot
- 1 Widerstände mit je ca. 270 Ohm
- Jumperkabel "male-female"

⁷⁰ Siehe online unter https://www.amazon.de/GeeekPi-BBC-Micro-Ultimate-microbit/dp/B0BKXLM97?ref=ast_sto_dp (20.03.2024)



Benötigte Software

- TinkerCAD⁷¹
- MakeCode-Editor von Microsoft⁷²
- Python-Editor für MicroPython⁷³
- Mu-Editor⁷⁴ lokal installiert

Generelle Vorsichtsmaßnahmen

- Niemals den 3,3-Volt-Anschluss direkt (ohne Vorwiderstand) mit Ground verbinden
- Bei elektronischen Steckübungen oder Veränderungen immer zuerst den Micro:bit vom Strom nehmen (von Batterie oder USB-Anschluss).
- Sich selbst vor der Arbeit mit Elektronik elektrostatisch entladen (Berühren von sonstigen metallischen Teilen wie Heizkörpern, Metalltischen etc.)

Projektbeschreibung

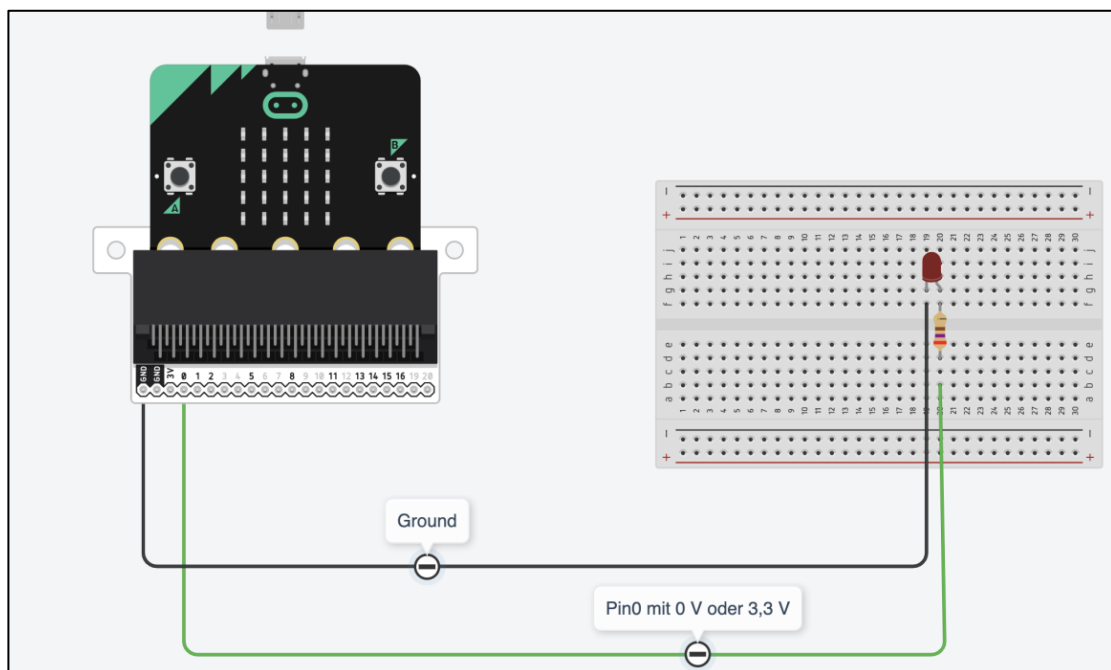


Abbildung 1: LED wird über Pin 0 aufgrund der Lichtstärke eingeschaltet.

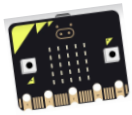
Die Abbildung 1 zeigt eine reine Reihenschaltung, deren Stromversorgung über einen schaltbaren Pin 0 erfolgt. Das Projekt misst über **den internen Lichtsensor des**

⁷¹ Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.03.2024)

⁷² Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁷³ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁷⁴ Downloadbar unter <https://codewith.mu/en/download> (25.03.2024)



Micro:bit die Lichtstärke von 0 bis 255. Ist die Lichtstärke unter 50 Lichteinheiten, wird die LED eingeschaltet - als symbolisch zusätzliche Beleuchtung.

Ein Vorwiderstand von 270 Ohm und eine LED sind in Reihe (hintereinander) geschaltet und der Strom fließt durch beide Komponenten zum Ground. Die Gesamtspannung von 3,3 Volt wird zunächst am Vorwiderstand auf 2 Volt abgesenkt und fließt anschließend durch die LED, die als Verbraucher fungiert. Die verbrauchte Energie wird als Licht der LED abgegeben. Die Anordnung von Vorwiderstand und LED könnte auch umgekehrt sein. Wichtig ist jedoch die Stellung der LED. Der Strom (die Elektronen) fließt zuerst durch die Anode (langes Bein der LED) und dann durch die Kathode (kurzes Bein). Bei umgekehrter Stellung würde der Stromfluss unterbrochen werden.

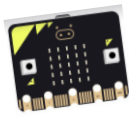
Merkhilfe: „*Nach dem Stromfluss verliert die LED einen Teil des Beins*“

Datenblatt

Damit eine LED beleuchtet werden kann, muss immer zuvor ein Datenblatt⁷⁵ der LEDs studiert werden (siehe Abbildung 2). Wichtig hierbei sind bei den LED-Typen rot, grün und gelb die **Durchlassspannung** und die **Stromstärke** durch die LED, in unserem Beispiel 2 V und 10 mA. Gefährlich sind die Begriffe „**Max. zulässiger Strom**“ und **Durchbruchspannung**. Vor allem der maximal zulässige Strom von 30 mA ist bei den obigen Schaltungen kein Thema, da konkret beim Micro:bit mit 5 mA⁷⁶ gearbeitet wird. Die Durchbruchspannung von 5V würde auf alle Fälle das LED zerstören, wird aber natürlich nie verwendet.

⁷⁵ Musterdatenblatt unter Reichelt, https://cdn-reichelt.de/documents/datenblatt/C200/DS_GER_RF19.pdf (abgerufen am 3.1.2024)

⁷⁶ Siehe Empfehlung von 5 mA für Micro:bit unter <https://support.microbit.org/support/solutions/articles/19000101863-connecting-an-led-to-the-micro-bit> (25.03.2024)



Technische Daten LED

3 mm LED

(Kenndaten gültig bei 25 °C)	LED rot	LED grün	LED gelb
Max. zulässiger Strom I_F :	30 mA	30 mA	20 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,5 mA/°C	ca. 0,5 mA/°C	ca. 0,2 mA/°C
Typ. Wellenlänge:	635 nm	565 nm	586 nm
Typ. Durchlassspannung U_F bei I_F :	2 V/10 mA	2 V/10 mA	2 V/10 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 μ A	min. 5 V/100 μ A	min. 5 V/100 μ A
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C
	LED blau	LED weiß	LED grün superhell
Max. zulässiger Strom I_F :	20 mA	25 mA	30 mA
Stromreduzierung: ab $T_0 = 50\text{ °C}$:	ca. 0,6 mA/°C	-	-
Typ. Wellenlänge:	470 nm	-	510-545 nm
Typ. Durchlassspannung U_F bei I_F :	2,7 V/10 mA	3,6 V/20 mA	3,5 V/20 mA
Durchbruchspannung U_R bei I_F :	min. 5 V/100 μ A	-	-
Arbeitstemperatur:	- 20 °C ... + 80 °C	- 20 °C ... + 80 °C	-30 °C ... + 100 °C

Abbildung 2: Datenblatt LED

Vorwiderstand berechnen

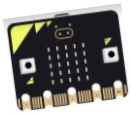
Für beide Schaltungen (Reihen- und Parallelschaltung) sollte eine empfohlene Durchlassspannung von 2 V erreicht werden. Somit muss ein geeigneter Vorwiderstand geschaltet werden, der die Ausgangsspannung von 3,3 V auf 2 V des LEDs um eine Differenz von 1,3 Volt senkt.

$$\text{Widerstand} = \frac{3,3V - 2V = 1,3V}{0,005\text{Ampere}(5mA)}$$

Dies ergibt einen Vorwiderstandswert von 260 Ω (Ohm). Praktisch wird aber der nächsthöhere Widerstand, in diesem Fall 270 Ω , verwendet. Den errechneten Widerstand gibt es in der Praxis selten genau als Bauteil.

Korrekten Widerstand verwenden

Physische Widerstände sind durch einen speziellen Farbcode gekennzeichnet. In der häufigsten Form handelt es sich um 4 oder 5 Ringe, die zur Bestimmung des Widerstands beachtet werden müssen. Eine genau Berechnung erledigen Farbtabelle oder



Online-Rechner⁷⁷. Beachte beim Lesen der Ringe die Anordnungs-Richtung. Der von den Betrags-Ringen weiter entfernte Toleranz-Ring, zeigt das Ende der Leserichtung. Wichtig ist die Berücksichtigung der Anzahl an tatsächlich vorhandenen Ringen am Widerstand.

Anzahl der Bänder
4 Ringe **5 Ringe** 6 Ringe

Widerstandsparameter

1. Farbring
Rot 2

2. Farbring
Violett 7

3. Farbring
Schwarz 0

Multiplikator
Schwarz $\times 1 \Omega$

Toleranz
Braun $\pm 1\%$

Resistance value
270 Ω

Ausgabe

Widerstandswert:
270 Ohms 1%

Abbildung 3: Farbdarstellung eines Vorwiderstands von 270 Ohm

LED richtig stecken

LED bestehen aus Anode (langes Bein) und Kathode (kurzes Bein). Zu beachten ist die Richtung des Stromflusses, da dieser zuerst durch die Anode (langes Bein) und dann erst durch die Kathode fließt. Wird die LED falsch gesteckt, findet KEIN Stromfluss statt. LEDs leiten somit Strom in die richtige Richtung, dienen aber auch als optische Hinweise (z.B. rotes LED als Fehlermeldung).

⁷⁷ Digikey, online unter <https://www.digikey.de/de/resources/conversion-calculators/conversion-calculator-resistor-color-code> (03.01.2024)

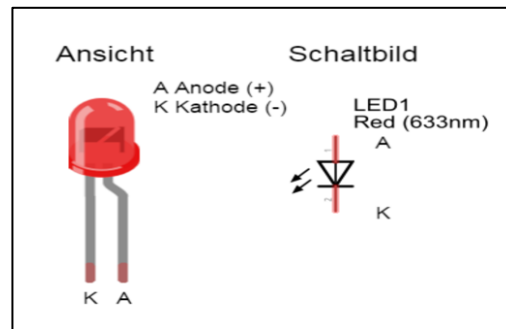
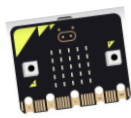


Abbildung 4: Stromfluss von Anode+ zu Kathode-(Quelle⁷⁸)

Interner Lichtsensor (Built-in-Sensor) und Lichtmessungen

Der Micro:bit misst über seine internen 25 LEDs die Lichtstärke. Diese Lichtstärke ist ein spezieller Wert, der nicht in Lumen gemessen wird, sondern eine eigene Messlogik von Micro:bit verfolgt. Die Werte gehen von 0 (dunkel) bis 255 (ganz hell) und werden softwareseitig⁷⁹ bestimmt. Nicht gesondert in der Micro:bit Dokumentation erwähnt, aber vermutlich handelt es sich um einen Analog-Digital-Converter ADC mit 8 Bits, der softwareseitig die Lichtgrenzen bestimmt. Dies weicht von den üblichen Werten der GPIO (Pins) ab, die generell mit einem ADC von 10 Bits arbeiten (0 bis 1023).

Schaltungsentwurf mit TinkerCAD

Optional kann mit oder ohne inkludierte, visuelle Programmiersprache das Funktionieren der Schaltung vorab getestet werden.

⁷⁸ Elab, LED, https://robotfreak.de/elab-wiki/index.php?title=LED_Leuchtdiode (03.01.2024)

⁷⁹ Micro:bit documentation siehe <https://tech.microbit.org/hardware/> (27.03.2024)

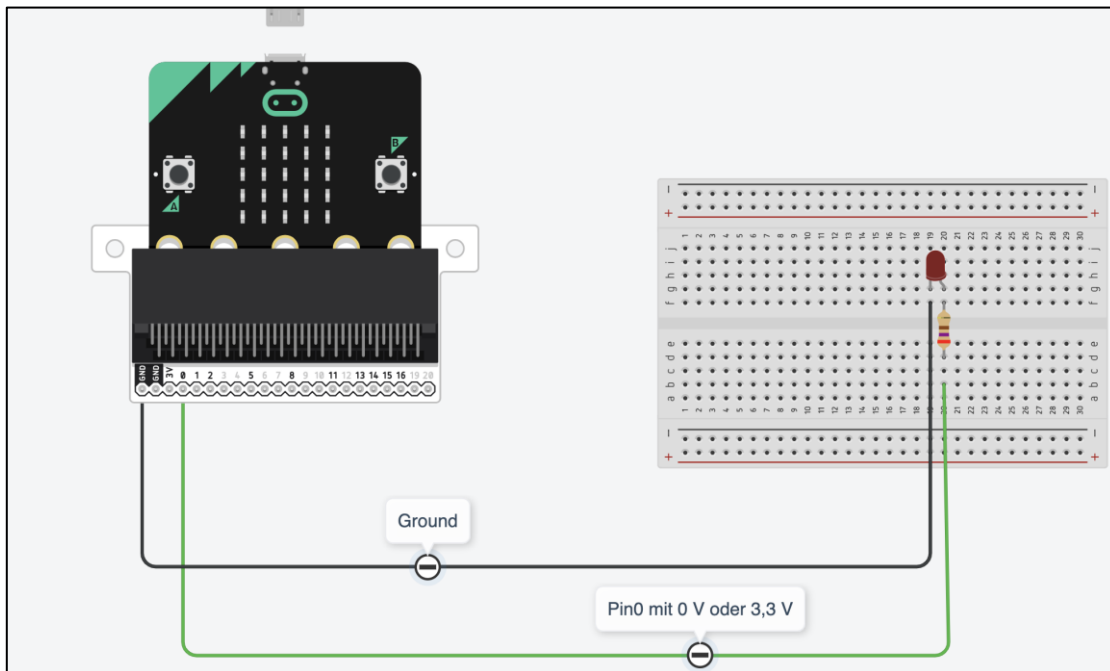
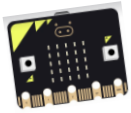


Abbildung 5: Schaltungsentwurf mit TinkerCAD

Programmierung grafisch mit MakeCode-Editor

Beim MakeCode-Editor gibt es bereits einen eigenen visuellen Baustein für die Lichtstärke. Zusätzlich wird im Code auch abgefragt, ob der Lichtwert unter 50 Lichteinheiten ist. In diesem Fall würde sich eine externe LED über Pin 0 dazuschalten.

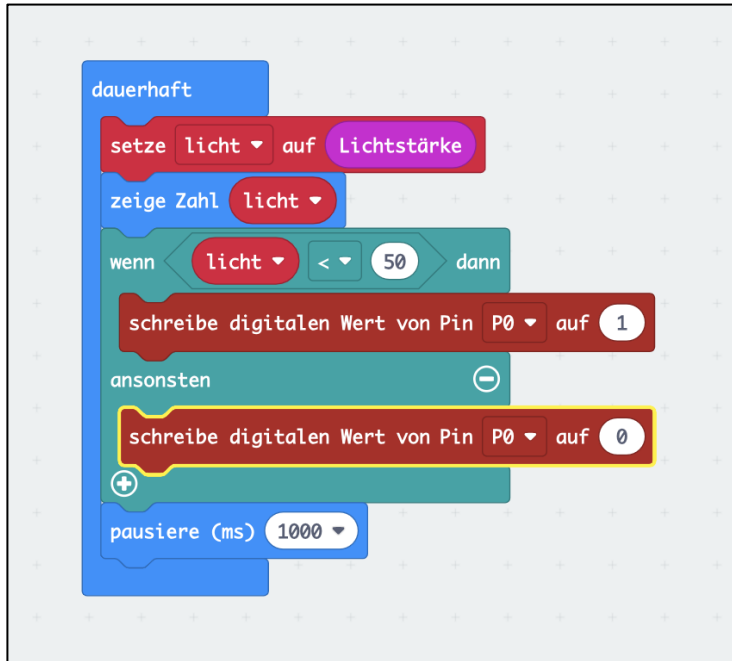
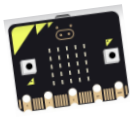
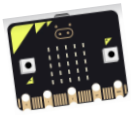


Abbildung 6: Grafische Programmierung mit MakeCode

Programmierung mit MicroPython Web bzw. Mu-Editor lokal

Der Code in MicroPython baut noch eine weitere Abfrage ein, die bei einem Lichtwert zwischen 50 und 130 die LED blinken lässt. Dies wird mit einer *While-Schleife* nach der *IF-Abfrage* bewerkstelligt. Ab einem Wert von 50 wird die LED dauerhaft eingeschaltet.



```
1 from microbit import *
2
3 display.show(Image.SMILE)
4
5 while True:
6     licht = display.read_light_level()
7     display.scroll(licht)
8     # zuerst die Abfrage nach dem kleinsten Wert
9     if licht < 50:
10         pin0.write_digital(1)
11
12     elif 50 <= licht < 130:
13         # Dauerblinken mit innerer while-Schleife
14         while 50 <= licht < 130:
15             pin0.write_digital(1)
16             sleep(100)
17             pin0.write_digital(0)
18             sleep(100)
19             licht = display.read_light_level() # Aktualisiere den Lichtwert
20         # Licht ausreichend
21     else:
22         pin0.write_digital(0)
23
24     sleep(1000)
```

Abbildung 7: Programmierung im Python-Editor

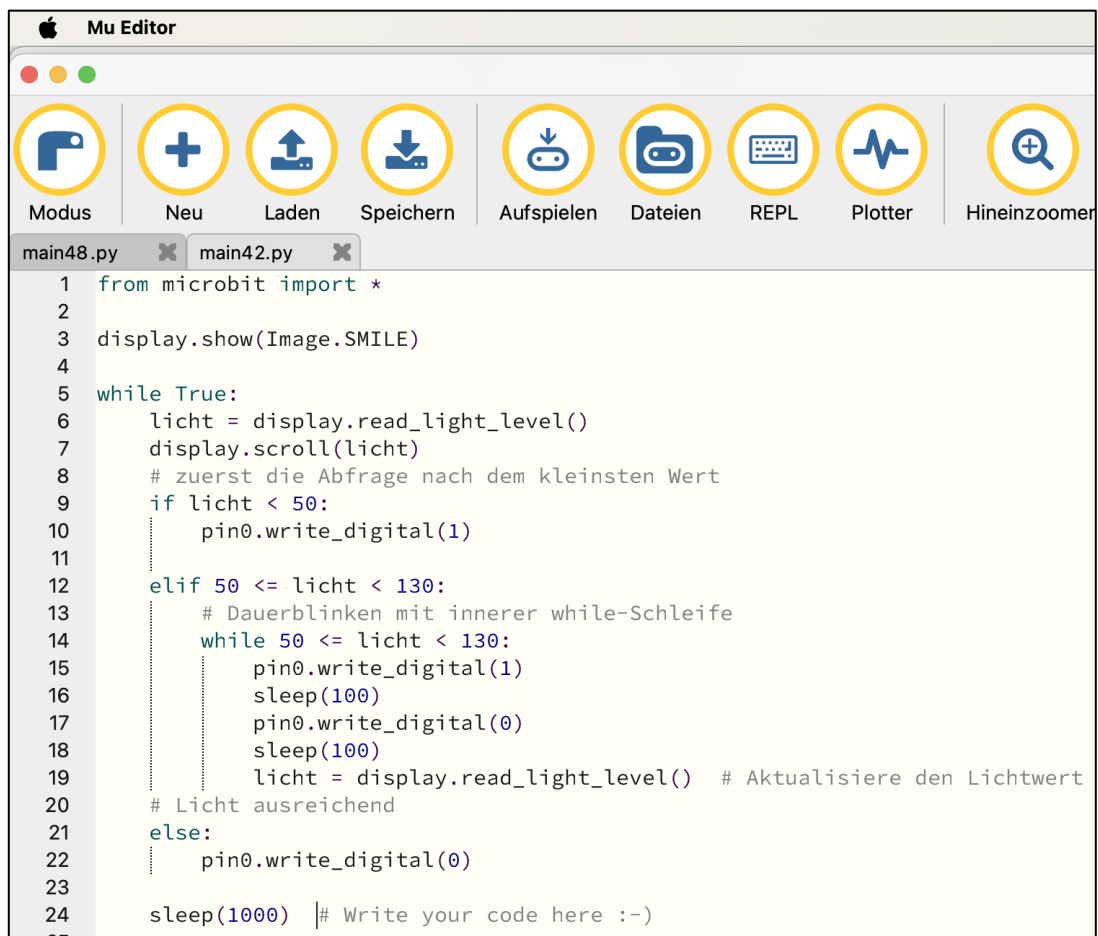
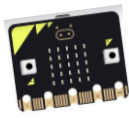


Abbildung 8: Gleicher Code im Mu-Editor



Ideen zur Erweiterung/Adaptierung/Vertiefung

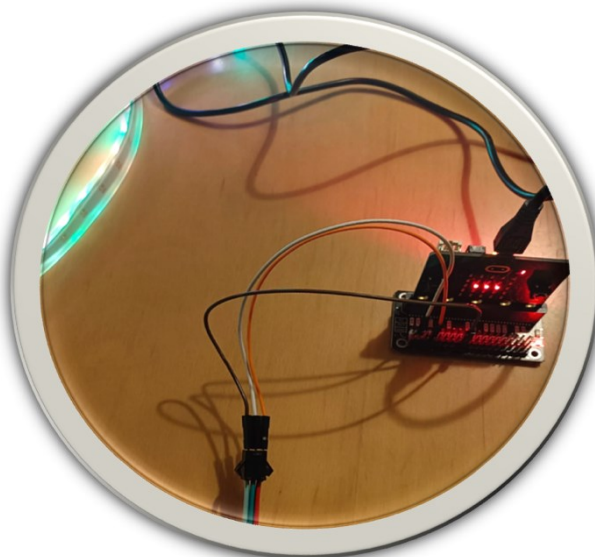
- Weitere Lichtgrenzen einbauen
- Höchsten Lichtwert mit visuellem Ton unterlegen
- Bei geringer Lichtstärke 2 LEDs einsetzen

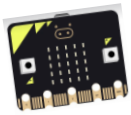
Ergänzende Lernunterlagen

- Micro:bit, Projekte und Erklärungen unter <https://microbit.org/> (25.03.2024)
- Grundlagen der Elektronik unter <https://studyflix.de/elektrotechnik/thema/elektrotechnik-grundlagen-20> (25.03.2024) bzw. <https://www.elektrotechnik-einfach.de/> (25.03.2024)
- Dokumentation für MicroPython für Micro:bit unter <https://microbit-micropython.readthedocs.io/en/v2-docs/> (24.03.2024)
- Beschreibung des Lichtsensors unter <https://support.microbit.org/support/solutions/articles/19000024023-how-does-the-light-sensing-feature-on-the-micro-bit-work-> (26.03.2024)

13.9 Praxisprojekt 8 (NeoPixel)

Projekt 8 – LED-Stripes





Benötigte Komponenten

- Micro: bit V2 mit Breakout-Board
- Starterkit GeekPi BBC Micro:bit V2 Ultimate Starter Kit optional⁸⁰
- LED-Strip Neopixel als externe Komponente
- Jumperkabel “male-female”

Benötigte Software

- TinkerCAD⁸¹
- MakeCode-Editor von Microsoft⁸²
- Python-Editor für MicroPython⁸³
- Mu-Editor⁸⁴ lokal installiert

Generelle Vorsichtsmaßnahmen

- Niemals den 3,3-Volt-Anschluss direkt (ohne Vorwiderstand) mit Ground verbinden
- Bei elektronischen Steckübungen oder Veränderungen immer zuerst den Micro:bit vom Strom nehmen (von Batterie oder USB-Anschluss)
- Sich selbst vor der Arbeit mit Elektronik elektrostatisch entladen (Berühren von metallischen Teilen wie Heizkörpern, Metalltischen etc.)

Projektbeschreibung

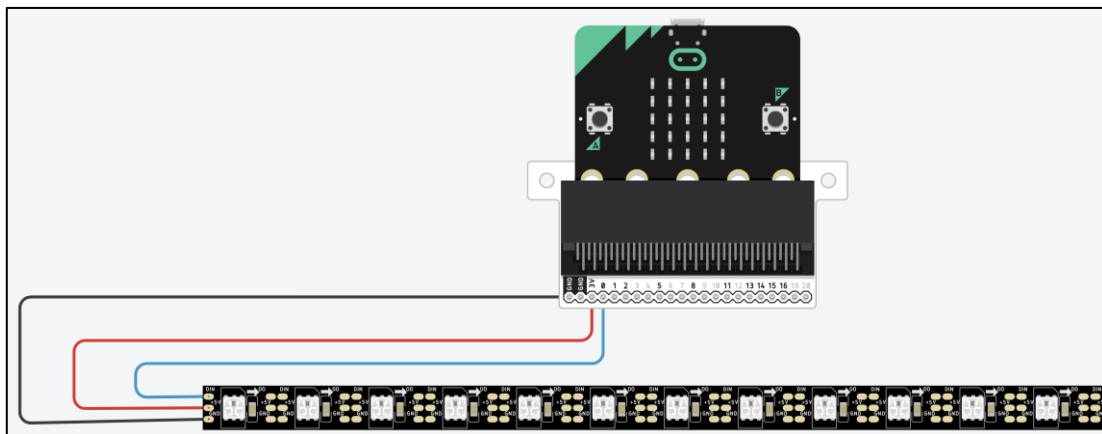


Abbildung 1: LED-Stripes über DIN (Digital In), Ground und VCC am Micro:bit anschließen.

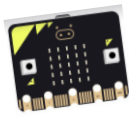
⁸⁰ Siehe online unter https://www.amazon.de/GeekPi-BBC-Micro-Ultimate-microbit/dp/B0BKXLM97?ref=ast_sto_dp (20.03.2024)

⁸¹ Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.03.2024)

⁸² Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁸³ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁸⁴ Downloadbar unter <https://codewith.mu/en/download> (25.03.2024)



Die Abbildung 1 zeigt eine LED-Strip, die über die Anschlüsse DIN (Digital In), VCC (3,3 Volt) und Ground am Micro:bit angeschlossen wird. Oberhalb des LED-Strips zeigt ein weißer Pfeil die richtige Anschlussrichtung an. Am Ende des LED-Strips könnten weitere LEDs angeschlossen werden.

Obwohl die Beschriftung mit „5V“ am LED-Strip angebracht ist, kann die LED-Kette auch mit 3,3 Volt betrieben werden. Der Anschluss erfolgt beim Strip mittels Male-Jumper, am Micro:bit-Board mittels Female-Jumper.

Datenblatt

PIN function			
NO.	Symbol	Function description	
1	VDD	Power supply LED	
2	DOUT	Control data signal output	
3	VSS	Ground	
4	DIN	Control data signal input	

Absolute Maximum Ratings			
Prameter	Symbol	Ratings	Unit
Power supply voltage	V_{DD}	+3.5~+5.3	V
Input voltage	V_I	-0.5~ $V_{DD}+0.5$	V
Operation junction temperature	T_{opt}	-25~+80	°C
Storage temperature range	T_{stg}	-40~+105	°C

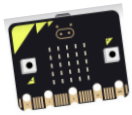
Electrical Characteristics ($T_A=-20\sim+70^{\circ}\text{C}$, $V_{DD}=4.5\sim5.5\text{V}$, $V_{SS}=0\text{V}$, unless otherwise specified)

Abbildung 2: Datenblatt der LED-Strip WS2812⁸⁵

LED-Strip „Neopixel“ Beschreibung

LED-Streifen, oft auch als Neopixel bekannt (ein spezifischer Markenname von Adafruit für LED-Streifen, die den WS2812B-LED-Chip verwenden), funktionieren durch die

⁸⁵ Siehe online unter <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf> (28.03.2024)



Kombination von kleinen Leuchtdioden (LEDs), die eng zusammen auf einem flexiblen Leiterband angeordnet sind.

Jede LED auf einem Neopixel-Streifen kann individuell gesteuert („adressiert“) werden, ermöglicht durch einen Controller-Chip (kleiner, integrierter Controller bei jedem LED). Die Farbe jeder LED wird typischerweise anhand eines RGB-Farbmodells (Rot, Grün, Blau) festgelegt. Durch das Mischen dieser drei Grundfarben in unterschiedlichen Intensitäten können verschiedene Farben erzeugt werden.

Die LEDs und Controller-Chips sind auf einer flexiblen Leiterbahn montiert, die auch die notwendigen elektrischen Verbindungen bereitstellt. Der Chip befindet sich direkt im LED-Gehäuse.

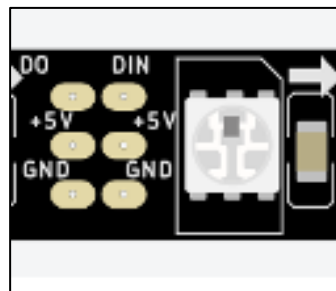
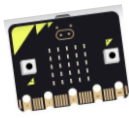


Abbildung 3: Controller ist direkt im LED-Gehäuse

Die Stromversorgung kann generell 5 V oder 3,3 Volt sein. Beim Einsatz des Micro:bits sollten stets 3,3 Volt verwendet werden. Mit dem eigentlichen Programmiercode ist auch ein Verweis auf das Modul *neopixel* mittels *import* einzubauen. Dieses Modul dient der eigentlichen Initialisierung und Steuerung der LED-Strips und dient gleichermaßen als Schnittstelle (API) und Treiber für die Hardware (Treiber ist eigentlich ein Begriff für eine Ansteuerung einer Hardware über ein Betriebssystem). Der Micro:bit hat kein eigenes Betriebssystem, sondern wird direkt über Bibliotheken und Firmware angesteuert.



NeoPixel

The `neopixel` module lets you use NeoPixel (WS2812) individually addressable RGB and RGBW LED strips with the micro:bit. Note to use the `neopixel` module, you need to import it separately with:

```
import neopixel
```

! Note

From our tests, the Microbit NeoPixel module can drive up to around 256 NeoPixels. Anything above that and you may experience weird bugs and issues. The micro:bit can only supply 90mA to external devices, larger numbers of NeoPixels require an external power supply with common ground.

NeoPixels are designed to work at 5V, but luckily they still function using the 3V supply of the BBC micro:bit. Please note that the micro:bit edge connector should not be connected to anything supplying 5V.

NeoPixels are fun strips of multi-coloured programmable LEDs. This module contains everything to plug them into a micro:bit and create funky displays, art and games such as the demo shown below.



Abbildung 4: Beschreibung der LED-Kette⁸⁶

Schaltungsentwurf mit TinkerCAD

Optional kann mit oder ohne inkludierte, visuelle Programmiersprache das Funktionieren der Schaltung vorab getestet werden.

⁸⁶ Online verfügbar unter <https://microbit-micropython.readthedocs.io/en/latest/neopixel.html> (26.03.2024)

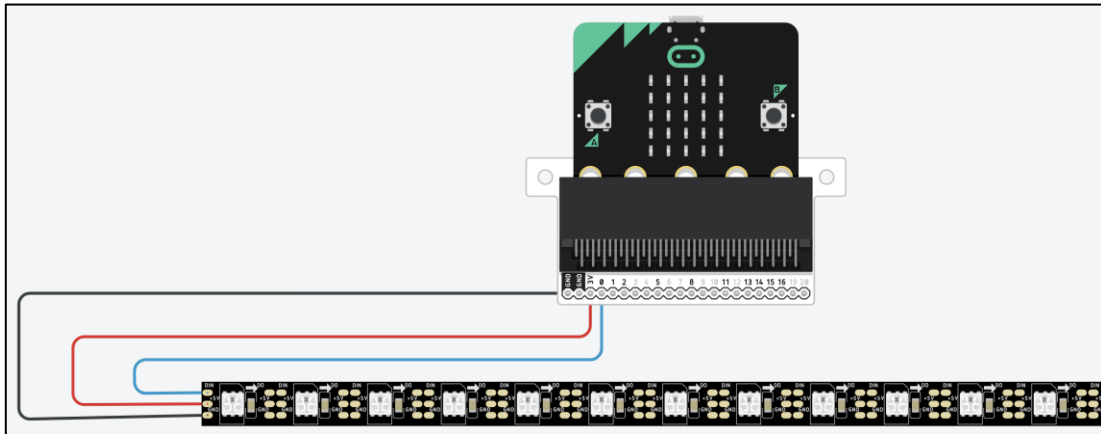
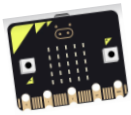


Abbildung 5: Schaltungsentwurf mit TinkerCAD

Programmierung grafisch mit MakeCode-Editor

Wichtig beim MakeCode-Editor ist vorab die Installation der Erweiterung *NeoPixel*. Zu Beginn muss das LED-Strip initialisiert werden mit den Werten 60 Pixel (LEDs am Leuchtband ZÄHLEN) und dem verwendeten Pin 0 zur Steuerung. Zu Beginn leuchtet das LED im Grundmuster „Regenbogen“. Dann werden fortlaufend die RGB-Grundfarben abwechselnd angezeigt.

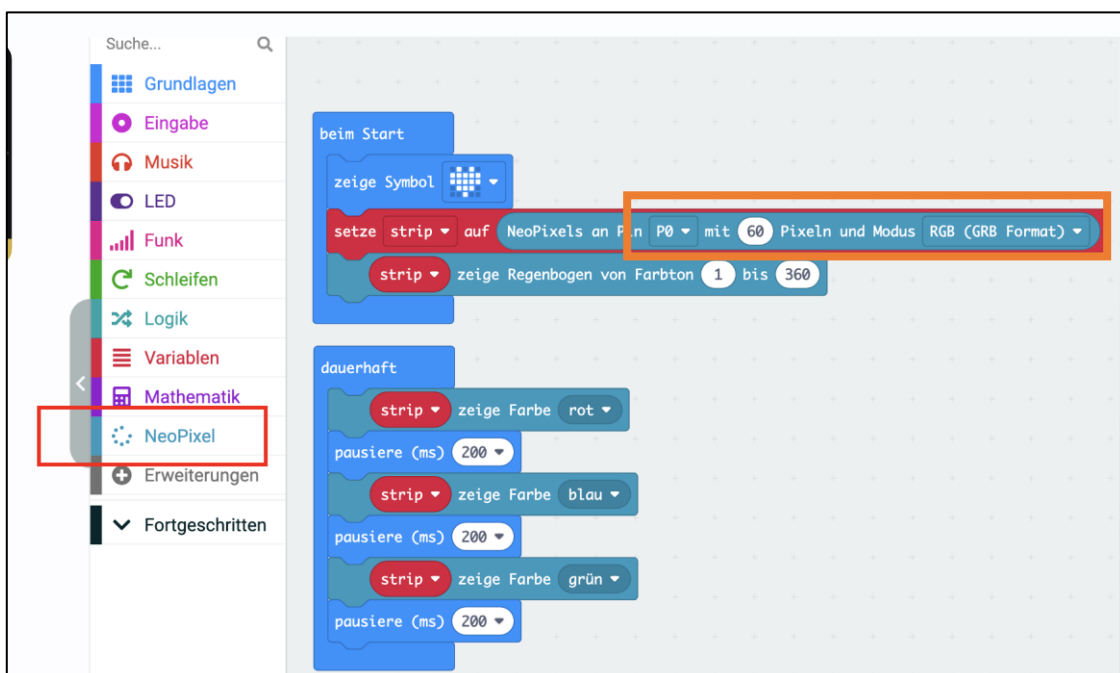
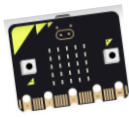


Abbildung 6: Grafische Programmierung mit MakeCode



Programmierung mit MicroPython Web bzw. Mu-Editor lokal

Die Programmierung im Python-Web-Editor ist gleichwertig mit der MakeCode-Eingabe. Neu ist eine Objekterstellung *np* aufgrund der Instanziierung der Klasse *NeoPixel*, die sich im Modul *neopixel* befindet.

```
1 from microbit import microphone, display, pin0, sleep
2 # wichtig für Ansteuerung, ist ein BuildIn Modul bei MicroPython für Microbit
3 import neopixel
4 # Setzen der vorhandenen Pixel auf 60, SteuerPin0
5 np = neopixel.NeoPixel(pin0, 60)
6
7 while True:
8     # Füllen mit Rot
9     np.fill((255,0,0))
10    np.show()
11    sleep(1000)
12    # Füllen mit Grün
13    np.fill((0,255,0))
14    np.show()
15    sleep(1000)
16    # Füllen mit Blau
17    np.fill((0,0,255))
18    np.show()
19    sleep(1000)
```

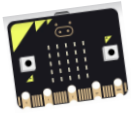
Abbildung 8: Programmierung im Python-Editor

Eine erweiterte, aufwändigere Programmierung zeigt dasselbe Beispiel im Mu-Editor. Hier wird noch eine gesonderte Lautstärkenmessung über das interne Mikrofon (erst ab Micro:bit V2) durchgeführt. Sollte der Lautstärkepegel zu hoch sein, ändern sich auch die Farben des LEDs.

Die Programmierung fokussiert den notwendigen Import der Erweiterung Neopixel. Fortgeschritten ist die Code-Zeile 5:

```
np = neopixel.NeoPixel(pin0, 60)
```

Damit wird ein Objekt (eine Objektinstanz) mit dem Namen *np* erzeugt, was ein Resultat der objektorientierten Programmierung darstellt. Die Umsetzung im Mu-Editor mit Zufallsfarben und einer Geräuschemessung ist konkret für Schüler*innen der SEK2 konzipiert.

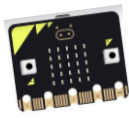


```
1 from microbit import microphone, display, pin0, sleep
2 import neopixel
3 from random import randint
4 # Setzen der vorhandenen Pixel auf 60, SteuerPin0
5 # Objekterstellung np aus Klasse NeoPixel
6 np = neopixel.NeoPixel(pin0, 60)
7 while True:
8     # Abfrage Mikrophon-Signal größer als 100
9     if microphone.sound_level() > 100:
10         display.show("++") # Anzeige sehr laut
11         sleep(200)
12         display.clear()
13         np.fill((255, 0, 0)) # mit roter Farbe füllen
14         np.show()
15         sleep(500) # blaue Pixel 2 Sekunden anzeigen
16     # Abfrage Mikrophon-Signal größer als 50 laut
17     elif microphone.sound_level() > 50:
18         display.show("+")
19         sleep(200)
20         display.clear()
21         np.fill((0, 0, 255))
22         np.show()
23         sleep(500)
24     # Normale Lautstärke
25     display.show("-")
26     sleep(200)
27     display.clear()
28     for pixel in range(0, len(np)): # alle 60 Pixel iterieren
29         # Zufallsfarbe erzeugen
30         red = randint(0, 255)
31         green = randint(0, 255)
32         blue = randint(0, 255)
33         # Weise dem aktuellen Pixel die Zufallsfarbe zu
34         np[pixel] = (red, green, blue)
35         # Zeige alle Pixel gleichzeitig an
36     np.show()
37     # eine halbe Sekunde warten um Zufallsfarben zu zeigen
38     sleep(500)
39     # unterbricht die While-Schleife für Stimmessung
40     # GANZ WICHTIG!!
41     sleep(10)
```

Abbildung 9: Gleicher Code im Mu-Editor

Zielgruppen

Lehrpersonen und Schüler*innen der SEK1 (Beispiele mit MakeCode und Python-Web-Editor bzw. SEK2 für die Adaptierung im Mu-Editor).



Ideen zur Erweiterung/Adaptierung/Vertiefung

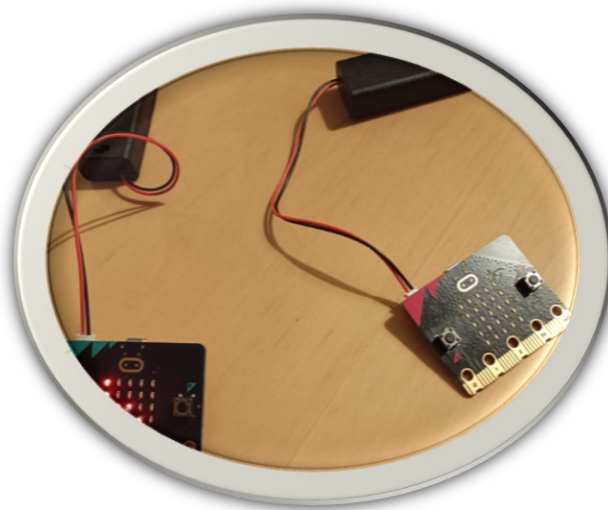
- Individuelle Farbtöne im RGB-Modus mischen
- Farbmuster mit Button A und Button B steuern
- Nicht alle LEDs verwenden
- Versorgung des LED-Stripes über ein passendes 5V-Netzteil ermöglicht die Zusammenarbeit aller Lernenden an einem zusammenhängenden LED-Stripe, bei dem einzelne Schüler*innen einen Abschnitt des Streifens programmieren.

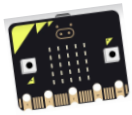
Ergänzende Lernunterlagen

- Micro:bit, Projekte und Erklärungen unter <https://microbit.org/> (25.03.2024)
- Grundlagen der Elektronik unter <https://studyflix.de/elektrotechnik/thema/elektrotechnik-grundlagen-20> (25.03.2024) bzw. <https://www.elektrotechnik-einfach.de/> (25.03.2024)
- Dokumentation für MicroPython für Micro:bit unter <https://microbit-micropython.readthedocs.io/en/v2-docs/> (24.03.2024)
- API von Neopixel für MicroPython unter <https://microbit-micropython.readthedocs.io/en/latest/neopixel.html> (28.03.2024)

13.10 Praxisprojekt 10 (Radio-Protokoll)

Projekt 10 – Radio-Protokoll





Benötigte Komponenten

- 2 (oder mehrere) Micro:bits V2
- Starterkit GeekPi BBC Micro:bit V2 Ultimate Starter Kit optional⁸⁷

Benötigte Software

- TinkerCAD⁸⁸
- MakeCode-Editor von Microsoft⁸⁹
- Python-Editor für MicroPython⁹⁰
- Mu-Editor⁹¹ lokal installiert

Generelle Vorsichtsmaßnahmen

- Niemals den 3,3-Volt-Anschluss direkt (ohne Vorwiderstand) mit Ground verbinden
- Bei elektronischen Steckübungen oder Veränderungen immer zuerst den Micro:bit vom Strom nehmen (von Batterie oder USB-Anschluss)
- Sich selbst vor der Arbeit mit Elektronik elektrostatisch entladen (Berühren von sonstigen metallischen Teilen wie Heizkörpern, Metalltischen etc.)

Projektbeschreibung

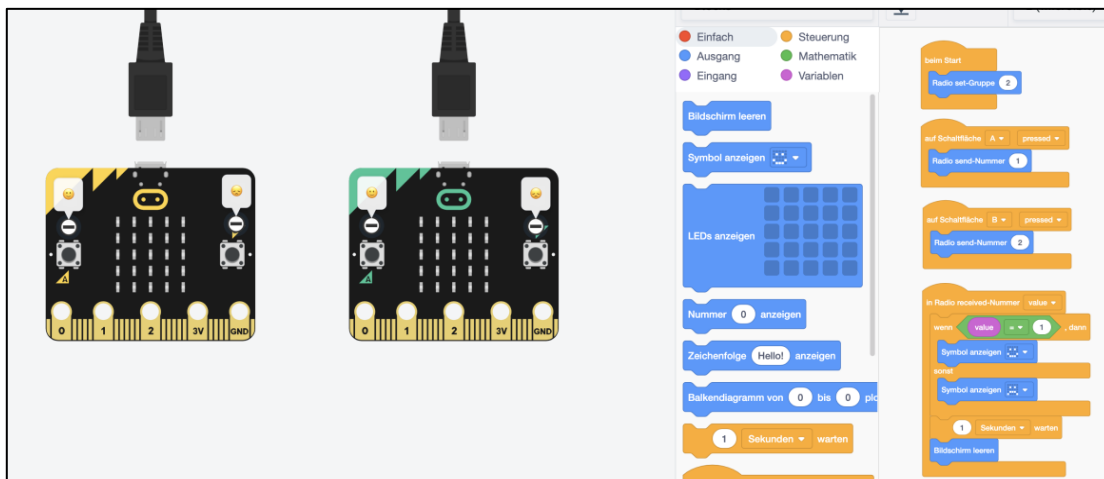


Abbildung 1: Fertiges Radio Template in TinkerCAD von Tremendous Jarv-Jaiks⁹²

⁸⁷ Siehe online unter https://www.amazon.de/GeekPi-BBC-Micro-Ultimate-microbit/dp/B0BKXLM97?ref=ast_sto_dp (20.03.2024)

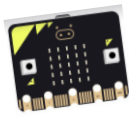
⁸⁸ Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.03.2024)

⁸⁹ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁹⁰ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

⁹¹ Downloadbar unter <https://codewith.mu/en/download> (25.03.2024)

⁹² Fertige Template nach Registrierung online verfügbar unter <https://www.tinkercad.com/things/k7AdB3qi6uG-tremendous-jarv-jaiks/editel?tenant=circuits> (abgerufen am 29.03.2024)



Die Abbildung 1 zeigt 2 Microbits, die sich gegenseitig die Zahlen 1 oder 2 schicken können. Die Übertragung erfolgt über das Funkprotokoll *Radio*. Button A generiert das Senden von 1 und damit wird am Partnergerät ein Smiley angezeigt. Button B sendet 2, der am Gegengerät eine SAD-Symbolik auslöst. Gerät A sendet an B und umgekehrt mit derselben Logik. In den TinkerCAD-Vorlagen ist bereits ein fertiges Radio-Beispiel abgebildet, das bereits auf der Website⁹³ direkt getestet werden kann (siehe auch Abbildung 1).

Datenblatt und Beschreibung

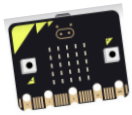
item	details
Protocol	Micro:bit Radio
Freq band	2.4GHz
Channel rate	1Mbps or 2Mbps
Encryption	None
Channels	80 (0..80)
Group codes	255
Tx power	Eight user configurable settings from 0(-30dbm) to 7 (+4dbm)
Payload size	32 (standard) 255 (if reconfigured)
More Info	Micro:bit Radio

Abbildung 2: Datenblatt LED⁹⁴

Das Radio-Protokoll wird zusammen mit dem Bluetooth-Protokoll durch ein integriertes 2,4-GHz-Funkmodul zur Verfügung gestellt. Dabei ist das Radio-Protokoll ein sehr einfaches, im Gegensatz zum aufwendigeren Bluetooth-Standard. Radio sendet standardmäßig über die Funkgruppe 0 unverschlüsselt, sodass sämtliche Informationen

⁹³ Fertige Template nach Registrierung online verfügbar unter <https://www.tinkercad.com/things/k7AdB3qi6uG-tremendous-jarv-jaiks/editel?tenant=circuits> (abgerufen am 29.03.2024)

⁹⁴ Quelle online unter <https://tech.microbit.org/hardware/> (29.03.2024)



von jedem Micro:bit abgehört werden könnten. Jedoch wäre es nicht möglich festzustellen, wer genau die Informationen versendet hat.

Radio sendet standardmäßig offen über die Funkgruppe 0, das heißt, dass jeder Micro:bit alle Infos mithören könnte. Allerdings würde man nicht erkennen, wer genau die Infos verschickt hat.

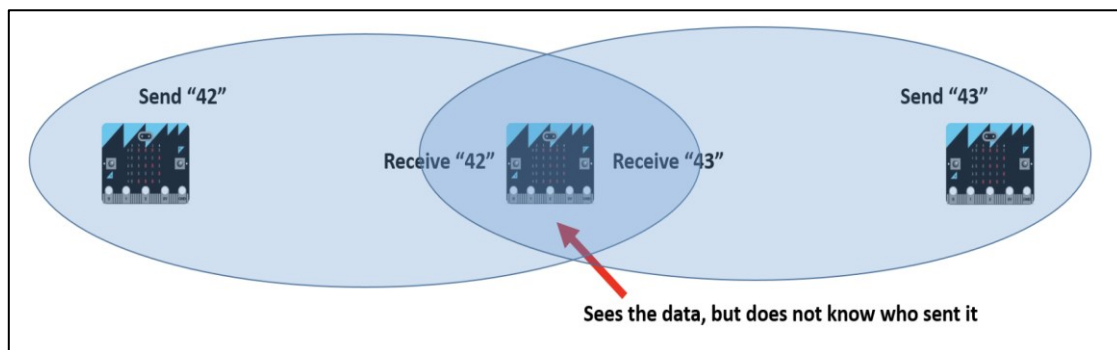


Abbildung 3: offene Übertragung von Informationen mit Radio⁹⁵

Damit die Übertragung auf einen Benutzerkreis eingeschränkt werden kann, werden Gruppen (0 bis 255) und Kanäle (0 bis 80) eingerichtet. Damit wird der Funkverkehr eingeschränkt. Die Übertragung erfolgt unverschlüsselt, ist also nicht sicher. Standardmäßig ist eine Übertragungseinheit mit 32 Bytes festgelegt, dies könnte aber auf bis zu 1024 Bytes erweitert werden. Der Funkabstand darf nur einige Meter betragen.

Schaltungsentwurf mit TinkerCAD

Optional kann mit oder ohne inkludierte, visuelle Programmiersprache das Funktionieren der Schaltung vorab getestet werden⁹⁶.

⁹⁵ Quelle online unter <https://lancaster-university.github.io/microbit-docs/ubid/radio/> (29.03.2024)

⁹⁶ Fertige Template nach Registrierung online verfügbar unter <https://www.tinkercad.com/things/k7AdB3qi6uG-tremendous-jarv-jaiks/editel?tenant=circuits> (29.03.2024)

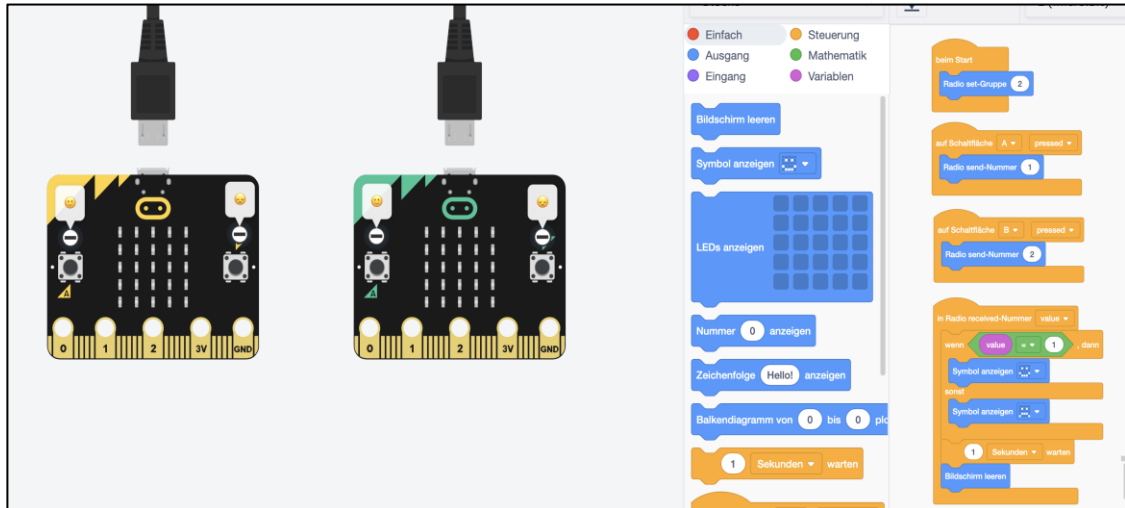
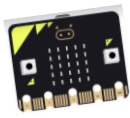


Abbildung 4: fertiger Schaltungsentwurf mit TinkercAD⁹⁷

Programmierung grafisch mit MakeCode-Editor

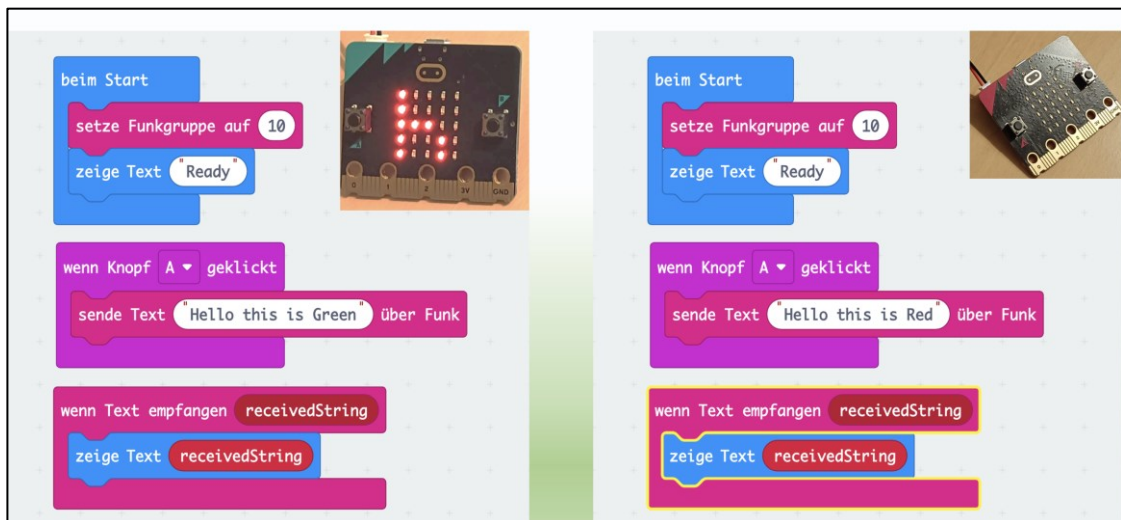
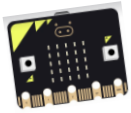


Abbildung 5: Grafische Programmierung mit MakeCode

⁹⁷ Fertige Template nach Registrierung online verfügbar unter <https://www.tinkercad.com/things/k7AdB3qi6uG-tremendous-jarv-jaijs/editel?tenant=circuits> (29.03.2024)



Programmierung mit MicroPython Web bzw. Mu-Editor lokal

Die Aktivierung und Einrichtung der individuellen Radio-Einstellungen erfolgt software-seitig über die Konfigurationen:

`radio.on()`

`radio.config(length=64, group=10)`



Abbildung 6: Programmierung im Python-Editor

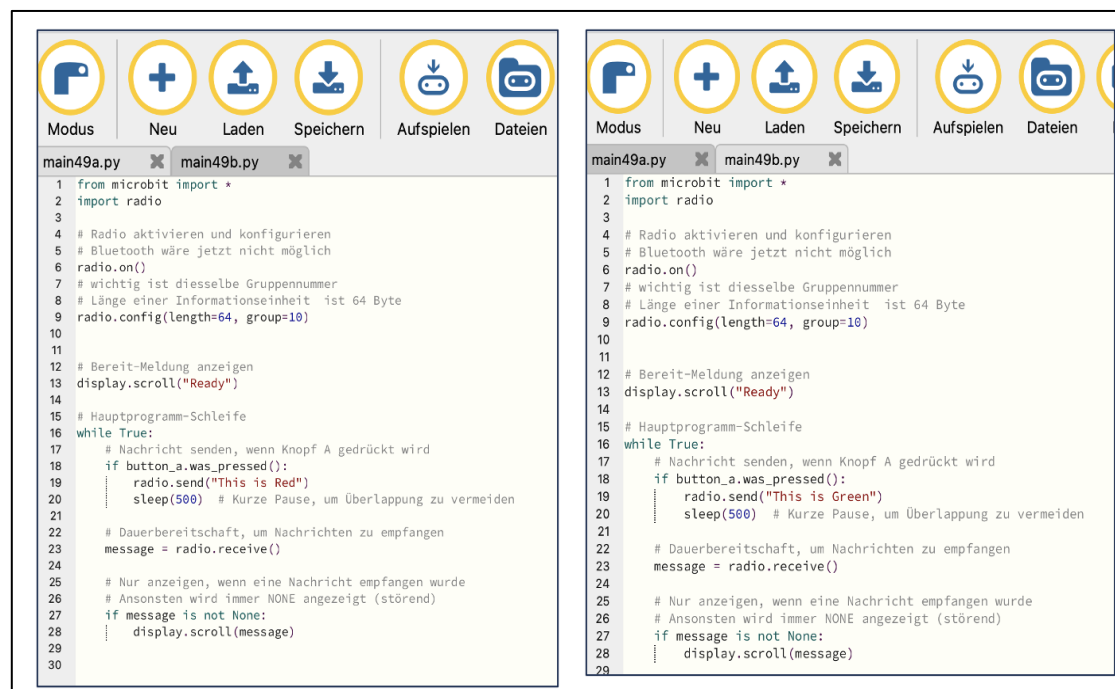
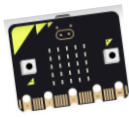


Abbildung 7: Gleicher Code im Mu-Editor



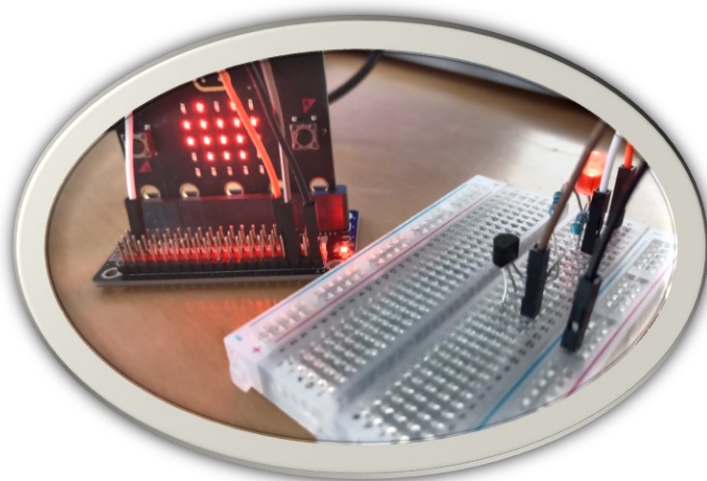
Ideen zur Erweiterung/Adaptierung/Vertiefung

- Die Anzahl der Micro:bits zusätzlich erweitern
- Filtern von Informationen auf Empfängerseite (z.B. nur Text, der mit „Hello“ beginnt)

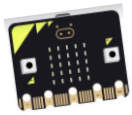
Ergänzende Lernunterlagen

- Micro:bit, Projekte und Erklärungen unter <https://microbit.org/> (25.03.2024)
- Grundlagen der Elektronik unter <https://studyflix.de/elektrotechnik/thema/elektrotechnik-grundlagen-20> (25.03.2024) bzw. <https://www.elektrotechnik-einfach.de/> (25.03.2024)
- Dokumentation für MicroPython für Micro:bit unter <https://microbit-micropython.readthedocs.io/en/v2-docs/> (24.03.2024)
- Radio-Protokoll erklärt unter <https://lancaster-university.github.io/microbit-docs/ubit/radio/> (29.03.2024)
- Hardware Micro:bit unter <https://tech.microbit.org/hardware/> (29.03.2024)

13.11 Praxisprojekt 11 (MOSFET)



Projekt 11 – MOSFET



Benötigte Komponenten

- Micro:bit V2 mit Breakout-Board
- Starterkit GeeekPi BBC Micro:bit V2 Ultimate Starter Kit optional⁹⁸
- Steckbrett
- 1 Light-Emitting-Diode (LED) rot
- 2 Widerstände mit je ca. 270 Ohm
- 1 MOSFET 2N7000
- Jumperkabel "male-female"

Benötigte Software

- TinkerCAD⁹⁹
- MakeCode-Editor von Microsoft¹⁰⁰
- Python-Editor für MicroPython¹⁰¹
- Mu-Editor¹⁰² lokal installiert

Generelle Vorsichtsmaßnahmen

- Niemals den 3,3-Volt-Anschluss direkt (ohne Vorwiderstand) mit Ground verbinden
- Bei elektronischen Steckübungen oder Veränderungen immer zuerst den Micro:bit vom Strom nehmen (Batterie oder USB-Anschluss)
- Sich selbst vor der Arbeit mit Elektronik elektrostatisch entladen (Berühren von metallischen Teilen wie Heizkörpern, Metalltischen etc.)

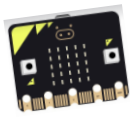
⁹⁸ Siehe online unter https://www.amazon.de/GeeekPi-BBC-Micro-Ultimate-microbit/dp/B0BKXLM97?ref=ast_sto_dp (20.03.2024)

⁹⁹ Online verfügbar unter <https://www.tinkercad.com/dashboard> (15.03.2024)

¹⁰⁰ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

¹⁰¹ Online verfügbar unter <https://microbit.org/code/> (20.03.2024)

¹⁰² Downloadbar unter <https://codewith.mu/en/download> (25.03.2024)



Projektbeschreibung

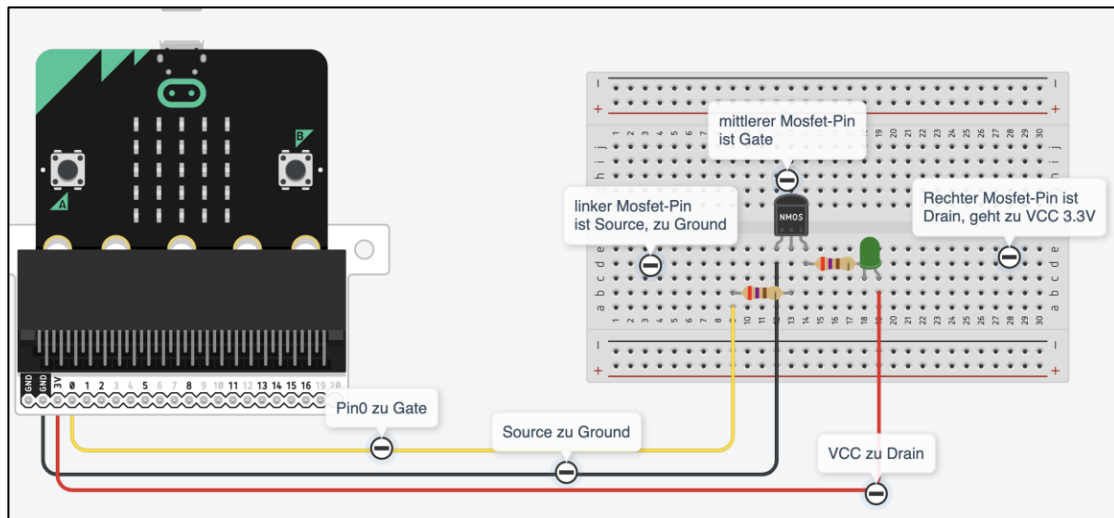
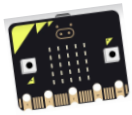


Abbildung 1: LED über Pin 0 ein- und ausschalten mit Aktivierung eines MOSFETS

Die Abbildung 1 zeigt eine Schaltung bestehend aus drei Bestandteilen. Die wichtigste Komponente ist ein MOSFET 2N7000. Dieser dient als Vermittlungsstelle zwischen seinen eigenen äußeren Anschlüssen Source (links) und Drain (rechts). In der Mitte befindet sich das Gate, das durch den Pin 0 des Micro:bit angesteuert wird.

Sendet der Pin0 eine „1“, wird das Gate unter Spannung gesetzt. Diese Spannung bewirkt, dass zwischen der MOSFET-Drain-Schaltung und der Source eine leitende Bahn entsteht und es fließt ein Strom zwischen den äußeren Komponenten. Beim Pin0-Zustand „0“ verschwindet diese „Bahn“ wieder und das Gate agiert dann als Isolator, so dass kein Strom fließt.

Zwei Vorwiderstände von 270 Ohm und eine LED sind in Reihe (hintereinander) geschaltet und der Strom fließt durch beide Komponenten. Die Gesamtspannung von 3,3 Volt wird zuerst bei jedem Vorwiderstand auf 2 Volt reduziert und fließt dann durch die LED als Verbraucher bzw. direkt in das Gate. Der Zweck des Vorwiderstands direkt am MOSFET-Gate ohne LED besteht darin, dass das Gate mit einer Gate-Schwellenspannung von 0,8 V bis 3 V betrieben wird. Durch den Vorwiderstand vor dem Gate



wird die Spannung auf 2 V reduziert, was optimal zur Schwellenspannung passt. Zusätzlich empfiehlt es sich immer, einen Vorwiderstand vor dem MOSFET-Gate zu verwenden, um vor einer zu hohen und schnellen Spannungsschaltung zu schützen.

Grundlagen MOSFET

Ein MOSFET ist eine spezielle Bauart eines Transistors. Bipolar-Transistoren arbeiten nach dem Prinzip, dass ein Stromfluss zwischen 2 Anschlüssen erst dann zustande kommt, wenn durch einen Base-Anschluss ein geringer „Reizstrom“ fließt und so zwischen Emitter und Kollektor einen stärkeren Stromfluss ermöglicht. Tatsächlich fließt aber durch den Base-Anschluss ein geringer und realer Reizstrom.

Eine ähnliche Funktion nimmt auch der sogenannte MOSFET wahr. Er besteht aus einem Gate (G), einer Source (S) und einem Drain (D). Wird am Gate eine positive Spannung angebracht, fließt zwischen Drain und Source ein Strom, da das positive Gate Elektronen anzieht und dadurch eine leitender „Kanal“ entsteht. Ohne Spannung verschwindet der Kanal.

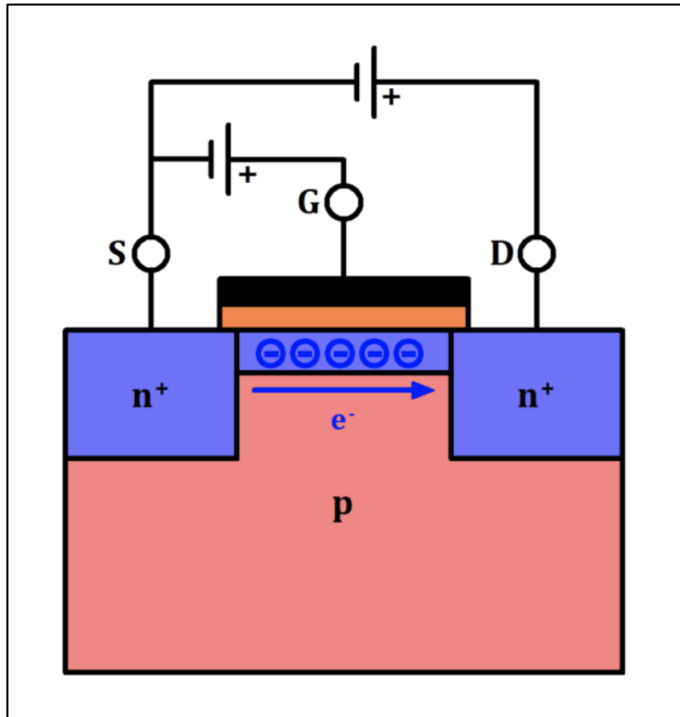
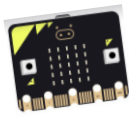


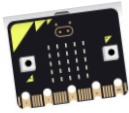
Abbildung 2: N-dotierter MOSFET¹⁰³

Wichtig ist im Unterschied zum Transistor die Tatsache, dass am Gate KEIN Strom fließt, sondern nur eine Spannung anliegt. Das ist der große Unterschied zum Bipolar-Transistor. Diese Tatsache ermöglicht schnelle Zustandswechsel, wie sie in der Digitaltechnik gebraucht und verwendet wird.

Datenblatt MOSFET 2N7000

Wichtig ist beim Datenblatt konkret die *Gate Threshold Voltage*. Diese „Reizspannung“ sollte zwischen 0,8 V und 3 Volt liegen. Theoretisch könnte zwischen Drain und Source eine Spannung von maximal 60 Volt existieren, was für das vorliegende Projekt aber nicht relevant ist.

¹⁰³ Quelle online verfügbar unter <https://studyflix.de/elektrotechnik/mosfet-2537> (abgerufen am 29.03.2024)



Electrical Characteristics $T_A = 25^{\circ}\text{C}$ unless otherwise noted							
Symbol	Parameter	Conditions	Type	Min	Typ	Max	Units
OFF CHARACTERISTICS							
BV_{DSS}	Drain-Source Breakdown Voltage	$V_{GS} = 0\text{ V}, I_D = 10\text{ }\mu\text{A}$	All	60			V
I_{DSS}	Zero Gate Voltage Drain Current	$V_{DS} = 48\text{ V}, V_{GS} = 0\text{ V}$	2N7000			1	μA
		$T_J = 125^{\circ}\text{C}$				1	mA
		$V_{DS} = 60\text{ V}, V_{GS} = 0\text{ V}$	2N7002 NDS7002A			1	μA
		$T_J = 125^{\circ}\text{C}$				0.5	mA
I_{GSSF}	Gate - Body Leakage, Forward	$V_{GS} = 15\text{ V}, V_{DS} = 0\text{ V}$	2N7000			10	nA
		$V_{GS} = 20\text{ V}, V_{DS} = 0\text{ V}$	2N7002 NDS7002A			100	nA
I_{GSSR}	Gate - Body Leakage, Reverse	$V_{GS} = -15\text{ V}, V_{DS} = 0\text{ V}$	2N7000			-10	nA
		$V_{GS} = -20\text{ V}, V_{DS} = 0\text{ V}$	2N7002 NDS7002A			-100	nA
ON CHARACTERISTICS (Note 1)							
$V_{GS(th)}$	Gate Threshold Voltage	$V_{DS} = V_{GS}, I_D = 1\text{ mA}$	2N7000	0.8	2.1	3	V
		$V_{DS} = V_{GS}, I_D = 250\text{ }\mu\text{A}$	2N7002 NDS7002A	1	2.1	2.5	
$R_{DS(ON)}$	Static Drain-Source On-Resistance	$V_{GS} = 10\text{ V}, I_D = 500\text{ mA}$	2N7000		1.2	5	Ω
		$T_J = 125^{\circ}\text{C}$			1.9	9	
		$V_{GS} = 4.5\text{ V}, I_D = 75\text{ mA}$			1.8	5.3	
		$V_{GS} = 10\text{ V}, I_D = 500\text{ mA}$	2N7002		1.2	7.5	
		$T_J = 100^{\circ}\text{C}$			1.7	13.5	
		$V_{GS} = 5.0\text{ V}, I_D = 50\text{ mA}$			1.7	7.5	
		$T_J = 100\text{C}$			2.4	13.5	
		$V_{GS} = 10\text{ V}, I_D = 500\text{ mA}$	NDS7002A		1.2	2	
		$T_J = 125^{\circ}\text{C}$			2	3.5	
		$V_{GS} = 5.0\text{ V}, I_D = 50\text{ mA}$			1.7	3	
$T_J = 125^{\circ}\text{C}$		2.8		5			

Abbildung 3: Datenblatt MOSFET 2N7000¹⁰⁴

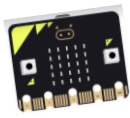
Vorwiderstand berechnen

Für beide Schaltungen für Gate und Drain muss eine empfohlene Durchlassspannung von 2 V erreicht werden. Somit muss ein geeigneter Vorwiderstand geschaltet werden, der die Ausgangsspannung von 3,3 V auf 2 V des LEDs um eine Differenz von 1,3 Volt senkt.

$$\text{Widerstand} = \frac{3,3\text{V} - 2\text{V} = 1,3\text{V}}{0,005\text{Ampere}(5\text{mA})}$$

Dies ergibt einen Vorwiderstandswert von 260 Ω (Ohm). Praktisch wird aber der nächsthöhere Widerstand, in diesem Fall 270 Ω , verwendet. In der Praxis ist es selten möglich, einen genau berechneten Widerstand als Bauteil vorzufinden.

¹⁰⁴ Datenblatt und Beschreibung unter <https://www.addicore.com/products/2n7000-logic-level-n-channel-mosfet> (29.03.2024)



Korrekten Widerstand verwenden

Physische Widerstände sind durch einen speziellen Farbcode gekennzeichnet. In der häufigsten Form handelt es sich um 4 oder 5 Ringe, die zur Bestimmung des Widerstands beachtet werden müssen. Eine genau Berechnung erledigen Farbtabeln oder Online-Rechner¹⁰⁵. Beachte beim Lesen der Ringe die Anordnungs-Richtung. Der von den Betrags-Ringen weiter entfernte Toleranz-Ring, zeigt das Ende der Leserichtung. Wichtig ist die Berücksichtigung der Anzahl an tatsächlich vorhandenen Ringen am Widerstand.

Anzahl der Bänder
4 Ringe **5 Ringe** 6 Ringe

Widerstandsparameter

1. Farbring: Rot 2

2. Farbring: Violett 7

3. Farbring: Schwarz 0

Multiplikator: Schwarz $\times 1 \Omega$

Toleranz: Braun $\pm 1\%$

Resistance value: 270 Ω

Ausgabe

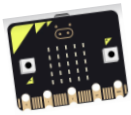
Widerstandswert: 270 Ohms 1%

Abbildung 4: Farbdarstellung eines Vorwiderstands von 270 Ohm

LED richtig stecken

LED bestehen aus Anode (langes Bein) und Kathode (kurzes Bein). Zu beachten ist die Richtung des Stromflusses, da dieser zuerst durch die Anode (langes Bein) und dann

¹⁰⁵ Digikey, online unter <https://www.digikey.de/de/resources/conversion-calculators/conversion-calculator-resistor-color-code> (03.01.2024)



erst durch die Kathode fließt. Wird die LED falsch gesteckt, findet KEIN Stromfluss statt. LEDs leiten somit Strom in die richtige Richtung, dienen aber auch als optische Hinweise (z.B. rotes LED als Fehlermeldung).

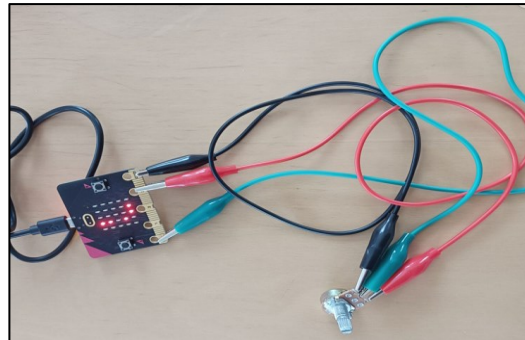


Abbildung 5: Stromfluss von Anode+ zu Kathode-(Quelle¹⁰⁶)

Schaltungsentwurf mit TinkerCAD

Optional kann mit oder ohne inkludierte, visuelle Programmiersprache das Funktionieren der Schaltung vorab getestet werden.

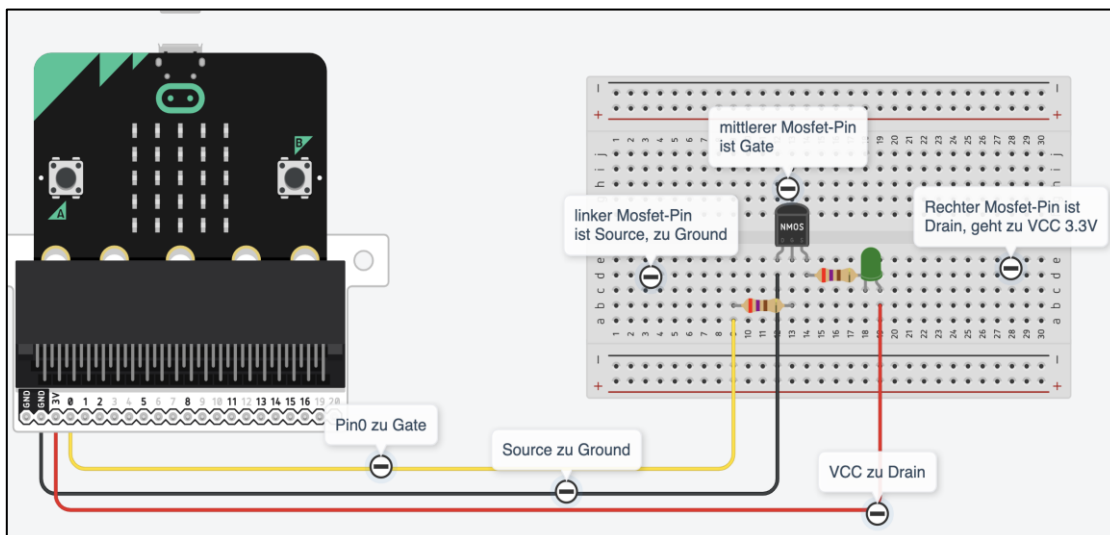
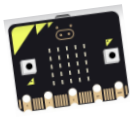


Abbildung 6: Schaltungsentwurf mit TinkerCAD

¹⁰⁶ Elab, LED, https://robotfreak.de/elab-wiki/index.php?title=LED_Leuchtdiode (03.01.2024)



Programmierung grafisch mit MakeCode-Editor

Wichtig beim MakeCode-Editor ist die Auswahl des Registers „Pins“. Hier sind alle Möglichkeiten zum Schalten von digitalen Werten am Pin 0 hinterlegt.

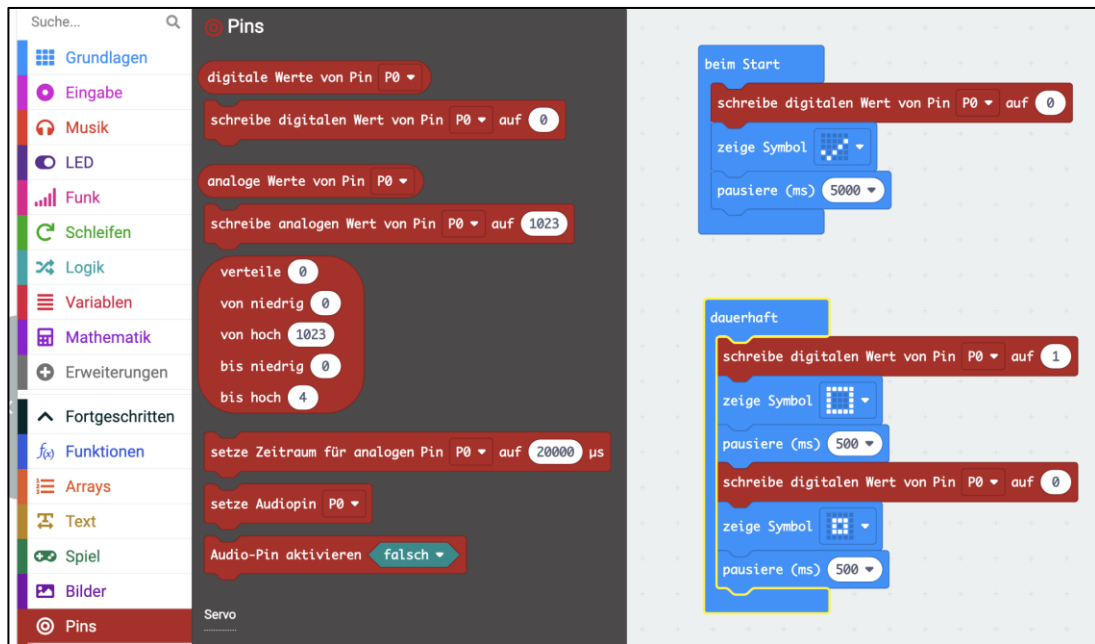
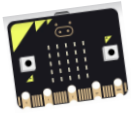


Abbildung 7: Grafische Programmierung mit MakeCode

Programmierung mit MicroPython Web bzw. Mu-Editor lokal

Bei der Lösung in MicroPython wurde noch ein akustisches Signal zur Betonung des Zustandes „1“ eingebaut.



```
1 from microbit import *
2 import music
3
4 # Konfigurieren Sie hier den gewünschten Pin
5 myMosfet_gate = pin0
6 myMosfet_gate.write_digital(0)
7 display.show(Image.SMILE)
8 # Sekunden Pause vor Start
9 sleep(5000)
10
11 while True:
12     # MOSFET-Gate einschalten (LED an), pin0 wird bestromt
13     myMosfet_gate.write_digital(1)
14     # Signalton
15     music.pitch(440)
16     display.show(Image.HEART)
17     sleep(1000)
18     music.stop()
19     # MOSFET-Gate ausschalten (LED aus), pin0 wird ausgeschalten
20     myMosfet_gate.write_digital(0)
21     display.show(Image.HEART_SMALL)
22     sleep(2000)
```

Abbildung 8: Programmierung im Python-Editor

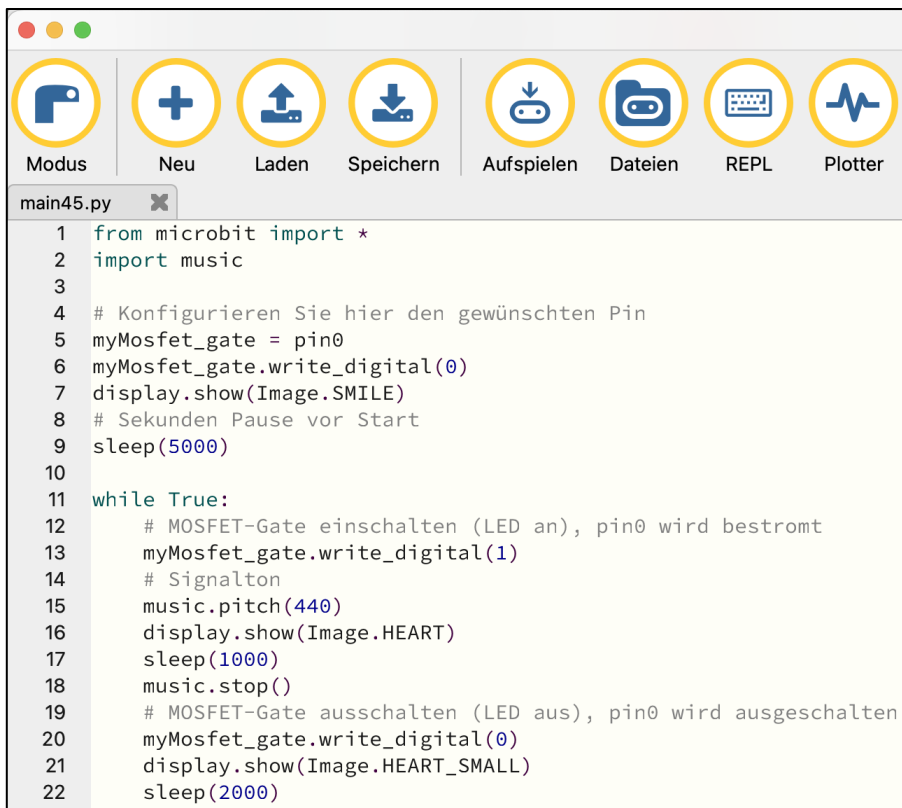
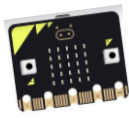


Abbildung 9: Gleicher Code im Mu-Editor



Zielgruppen

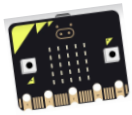
Lehrpersonen und Schüler*innen der SEK1 und SEK2

Ideen zur Erweiterung/Adaptierung/Vertiefung

- Mehrere MOSFETS schalten
- Statt eines Vorwiderstands beim Gate ein Potentiometer verwenden. Wenn Widerstand im Potentiometer zu hoch, keine Schwellspannung am Gate.
- MOSFET absichtlich falsch stecken

Ergänzende Lernunterlagen

- Micro:bit, Projekte und Erklärungen unter <https://microbit.org/> (25.03.2024)
- Grundlagen der Elektronik unter <https://studyflix.de/elektrotechnik/thema/elektrotechnik-grundlagen-20> (25.03.2024) bzw. <https://www.elektrotechnik-einfach.de/> (25.03.2024)
- Dokumentation für MicroPython für Micro:bit unter <https://microbit-micropython.readthedocs.io/en/v2-docs/> (24.03.2024)
- MOSFETS einfach erklärt unter <https://studyflix.de/elektrotechnik/mosfet-2537> (29.03.2024)



14. Abstract

Diese Masterthesis befasst sich mit der Herausforderung, die Themen „Technische Informatik“ bzw. „Digitaltechnik“ inhaltlich und didaktisch optimiert an österreichischen Schulen der Sekundarstufe 1 (SEK1) bzw. an Gymnasien der Sekundarstufe 2 (SEK2) zu unterrichten. Die hierfür relevanten Wissensbereiche aus Mathematik, Elektronik, Physik, Informatik und Kommunikationstechnik werden bis dato nicht gebündelt unterrichtet, sondern auf mehrere Lehrfächer verteilt. Dies allerdings weder fächerübergreifend noch inhaltlich ergänzend. Um die relevanten technischen Grundlagen geschlossen in den Unterrichtsfächern Digitale Grundbildung (SEK1) und Informatik (SEK2) zu bündeln, werden konkrete Schulungsunterlagen in Form von zehn Projekten für Physical Computing ausgearbeitet. Die Projekte haben einen vorgegebenen Schulungsaufbau. Dieser umfasst ein virtuelles Schaltungsdesign, den praktischen Aufbau mit Mikrocontroller samt Schaltkreis sowie die visuelle und textuelle Programmierung der gewünschten Funktionen. Eine zweimalige, ganztägige Fortbildung wird für zwei verschiedene Gruppen von Lehrkräften durchgeführt und anschließend ausgewertet. Die Umfrage bestätigt sieben definierte Thesen und beantwortet damit die Forschungsfrage nach einem erleichterten Einstieg in die Technische Informatik bzw. Digitaltechnik durch die vorhandenen Schulungsunterlagen. Eine interdisziplinäre Fortbildung für Lehrende mit zehn dokumentierten Referenzprojekten könnte aufgrund der Ergebnisse der vorliegenden Arbeit in Zukunft eine mögliche Unterrichtsgrundlage für Vorarlberger SEK1- und SEK2-Schulen sein.