## Das PANaMa-Projekt

### PANaMa-Projektet

Hrsg.

Dr. Stefanie Herzog

Dr. Marc Wilken

Science Camps und Technologieverständnis – Wege zu beruflichen Kompetenzen

> Science Camps og teknologiforståelse – metoder til at opnå faglige færdigheder

#### Vorwort

Die unterschiedlichen Phasen des PANaMa-Projekts haben verschiedene Schwerpunkte. Nach den Konzepten für die Erweiterung von konkreten beruflichen Perspektiven in naturwissenschaftlich-mathematischen Unterrichtsfächern zu Beginn des Projekts in der Arbeit mit den Jugendlichen und anschließenden Ausstellungsprojekten an den Schulen sowie Workshops für Lehrkräfte, liegt zum Ende des Projekts auch ein Fokus auf dem Ausbau fächerübergreifender und digitaler Kompetenzen der Schülerinnen und Schüler. Insbesondere diese Kompetenzen sind in vielen Bereichen der zukünftigen Arbeitswelt, auf die auch in der Schule vorbereitet werden soll, elementar. Die Kompetenzen sind sehr vielschichtig und reichen von einem Grundverständnis logischer Kausalketten, über ein Verständnis von Technologie und digitalen Prozessen bis hin zu Grundlagen des Programmierens. Oft kann dies aber nicht allein im Rahmen der Curricula an Schulen vermittelt werden – hier können ergänzend außerschulische Lehrangebote und Lernorte eine Möglichkeit bieten, die digitalen Kompetenzen und das Wissen um moderne und zukunftsweisende Technologien von Jugendlichen zu fördern.

Über die Kooperationen im Projekt, insbesondere mit der Teknologiskolen an der Süddänischen Universität in Odense, wurden Konzepte für Lernorte fernab von der gewohnten Schulumgebung geschaffen. Im Vordergrund standen hier Technologieverständnis und Robotik – zwei Themen, für die an der Teknologiskolen bereits seit längerem Workshops und Camps für Kinder und Jugendliche angeboten wurden. Die Idee von Forschungscamps wurde in das PANaMa-Projekt integriert und in vielfältiger Weise auf beiden Seiten der Grenze umgesetzt. Den Auftakt des vorliegenden Bandes bildet eine kurze Übersicht über das Konzept der Forschungscamps, gefolgt von Beispielen dieser im Projekt umgesetzten Camps mit einigen Eindrücken der Beteiligten. Als ein zentrales Thema dieser Camps kristallisierte sich die Robotik heraus, da es einen guten Einstieg in ein Verständnis technischer Abläufe, Möglichkeiten und Problematiken bietet und es ermöglicht, zum Teil mit spielerischen Elementen erste Kenntnisse im Programmieren zu erwerben. Die letzten beiden Kapitel dieses Bandes, die sich auf dieses Thema fokussieren, sind aber nicht nur Erläuterungen sondern bieten auch praktische Anleitungen und Hinweise, wie Technologieverständnis in den naturwissenschaftlichen oder technischen Schulunterricht eingebracht werden kann oder sich auch selbstständig erschließen lässt. Damit bildet es eine praktische Ergänzung zum bereits im 1. Band angesprochenen theoretischen Beitrag über Roboter. Auf weiteres Material im Zusammenhang zu den Camps und anderen Informationen kann darüber hinaus auch nach Abschluss des Projekts über die Webseite www.panama-project.eu zugegriffen werden.

Projektkoordination und Herausgeber

Dr. Marc Wilken

Dr. Stefanie Herzog

#### **Forord**

De forskellige faser i PANaMa-projektet har hver deres fokus. Projektets første faser fokuserede på koncepter til at udvide specifikke faglige perspektiver inden for naturvidenskabelige og matematiske fag. Disse koncepter blev udarbejdet sammen med de unge og efterfølgende præsenteret i udstillingsprojekter på skolerne. Desuden var der yderlige tilbud til lærere om workshops omkring koncepterne. Projektets slutning fokuserede også på udvidelsen af elevernes tværfaglige og digitale kompetencer. Disse kompetencer er især elementære i mange områder af det fremtidige arbejdsliv, som eleverne også skal forberedes på i skolen. Kompetencerne er meget komplekse og rækker fra en grundlæggende forståelse af logiske kausale kæder til en forståelse af teknologi og digitale processer og en grundlæggende forståelse i programmering. Ofte kan dette dog ikke undervises i skolerne på grund af deres curriculum - her kan supplerende kurser og undervisningssteder, der ligger uden for skolen, tilbyde en mulighed for at fremme digitale færdigheder og viden om moderne og fremtidsorienterede teknologier hos unge.

Gennem samarbejdet i projektet, især med Teknologiskolen ved Syddansk Universitet i Odense, blev der skabt koncepter til undervisningssteder langt væk fra det velkendte skolemiljø. Fokus var her på en forståelse af teknologi og robotik. Teknologiskolen har i lang tid tilbudt workshops og lejre for børn og unge for især disse to emner. Idéen om forskningslejre blev integreret i PANaMa-projektet og implementeret på en række måder på begge sider af grænsen. Dette bind begynder med en kort oversigt over konceptet for forskningslejrene, efterfulgt af eksempler på lejrene, der blev implementeret i projektet, samt nogle indtryk fra de involverede. Emnet robotik opstod som et centralt tema i disse lejre, da det giver en god introduktion til en forståelse af tekniske processer, muligheder og problemer. Desuden giver emnet også mulighed for at tilegne sig første programmeringsevner igennem nogle legende elementer. De sidste to kapitler i dette bind, der fokuserer på dette emne, er ikke kun forklaringer, men giver også praktiske instruktioner og råd om, hvordan teknologiforståelse kan indarbejdes i naturfag eller teknisk skoleundervisning eller kan udvikles uafhængigt. Dermed er det en praktisk tilføjelse til det teoretiske bidrag til robotter, der allerede er nævnt i bind 1. Via hjemmesiden www.panama-project.eu forbliver yderligere materiale relateret til lejrene og anden information tilgængelig, også efter projektet er afsluttet.

Projektkoordinering og udgiver

Dr. Marc Wilken

Or. Stefanie Herzog

#### 8 – 15 I. Forschercamps

Ein kommunikatives Konzept für praktische Erfahrungen zur Verbindung von fachbezogenem Wissen und Berufsorientierung

Prof. Dr. Ilka Parchmann, Dr. Stefanie Herzog / IPN

#### 16 – 23 II. Drei Tage mit Robotern und Drohnen

Schüler-Forschungscamp des Projekts PANaMa, März 2018

Dr. Marc Wilken / IPN

#### 24 – 39 III. PANaMa Camps in Bildern

#### 40 – 55 IV. Autumn Camps in der Kieler Forschungswerkstatt

Außerschulische Lernangebote zur Kryptographie und Robotik

Dr. Stefanie Herzog / IPN

#### 56 – 67 V. Aus der Sicht von Lehrkräften und Schüler\*innen

Das meinten die Lehrerinnen und Lehrer im Nachhinein zum PANaMa Camp 2018

Exemplarische Schüleraussagen

#### 68-133 VI. Schwerpunkt Technologieverständnis

Technologische Grundbildung sichert den Zugang zur modernen digitalen Arbeitswelt – Robotik und Kryptographie im Schulunterricht

Prof. Claus Michelsen, Linda Ahrenkiel, Kaj Nedergård Jepsen (UC Syd), Annette Jäpelt (UCL) / Laboratorium for Sammenhængende Uddannelse og Læring (LSUL), SDU Odense; Prof. Jørgen Christian Larsen, Prof. Jacob Nielsen, Bjarke Kristian Maigaard Kjær Pedersen / Teknologiskolen, SDU Odense; Benedikt Karrasch, Stefanie Herzog / IPN Kiel

#### 134 – 223 VII. Arduinos im Robo-Camp

Ausführliche Beschreibung eines Ablaufs zum Einsatz von Arduinos

Bjarke Kristian Maigaard Kjær Pedersen / SDU

#### 8 – 15 I. Forskercamps

Et kommunikationskoncept til praktisk erfaring, der forbinder fagrelateret viden og erhvervsorientering

Prof. Dr. Ilka Parchmann, Dr. Stefanie Herzog / IPN

#### 16 – 23 II. Tre dage blandt robotter og droner

Elev-forskningslejr i PANaMa-projektet, marts 2018

Dr. Marc Wilken / IPN

#### 24-39 III. PANaMa camps i billeder

#### 40 – 55 IV. Autumn Camps i Kiels forskningsværksted

Læringsmuligheder uden for skoleundervisningen for kryptografi og robotik

Dr. Stefanie Herzog / IPN

#### 56 – 67 V. Fra lærernes og elevernes perspektiv

Det mente lærerne efterfølgende om PANaMa Campen 2018

Eksempler på elevudtalelser

#### 68–133 VI. Fokus på teknologisk forståelse

Grundlæggende teknologisk uddannelse sikrer adgang til den moderne digitale arbejdsverden – robotik og kryptografi i skoletimer

Prof. Claus Michelsen, Linda Ahrenkiel, Kaj Nedergård Jepsen (UC Syd), Annette Jäpelt (UCL) / Laboratorium for Sammenhængende Uddannelse og Læring (LSUL), SDU Odense; Prof. Jørgen Christian Larsen, Prof. Jacob Nielsen, Bjarke Kristian Maigaard Kjær Pedersen / Teknologiskolen, SDU Odense; Benedikt Karrasch, Stefanie Herzog / IPN Kiel

#### 134 – 223 VII. Arduinos i Robo-Campen

Detaljeret beskrivelse af en procedure til anvendelse af Arduinos

Bjarke Kristian Maigaard Kjær Pedersen / SDU

## I. Forschercamps



## I. Forskercamps



### Ein kommunikatives Konzept für praktische Erfahrungen zur Verbindung von fachbezogenem Wissen und Berufsorientierung

Prof. Dr. Ilka Parchmann, Dr. Stefanie Herzog / IPN

Forschercamps oder auch Science Camps können als ein Ansatz informeller oder außerschulischer Lerngelegenheiten gesehen werden, die eine Möglichkeit bieten sollen, das Interesse der Schüler\*innen an Naturwissenschaften zu steigern und dieses sowie die eigene Identifikation mit Naturwissenschaften wie auch Mathematik aufrechtzuhalten (Gibson & Chase, 2002). Diese Ziele korrespondieren mit denen des kontextbasierten und kontextorientierten Lernens, die der Gesamtkonzeption des PANaMa-Projekts zugrunde liegen. Die Ergänzung des regulären Fachunterrichts sowie schulischer Projekte durch Science Camps bietet eine Reihe von Chancen gerade mit Blick auf die Verbindung von fachlichem, schulischem Wissen und Berufsorientierung, wie etwa bei Knox et al., 2003 berichtet durch die Arbeit an authentischen Orten, die Einbindung von Fachexperten oder den Austausch mit anderen Jugendlichen über die eigene Klasse und Peer Group hinaus. Letzteres ist für ein aus dem EU-Interreg-Programm gefördertes Projekt zentral, gerade für Flächenregionen, in denen die Vielfalt der beruflichen Perspektiven oftmals nicht in der eigenen Region allein sichtbar wird.

Eine breite Summer Science Camp-Kultur gibt es insbesondere in den U. S. A., die Ansätze unterschiedlicher Dauer, Abgeschiedenheit, adressierter Altersgruppe, Art der Tätigkeiten sowie Ziele und Einbindung wissenschaftlicher Einrichtungen beinhaltet (vgl. Fields, 2009). So variierten diese Camps von eintägigen Besuchen an außerschulischen Lernorten bis zu mehrtägigen Events in oder außerhalb der Schule. Das Programm NaWi-aktiv beispielsweise hat Sommercamps als Abschluss der Arbeit in Arbeitsgemeinschaften konzipiert, um zuvor Gelerntes im Rahmen einer gemeinsamen Geschichte (vgl. Bertelsen, 2014: Konzept des Storytellings) anzuwenden und für eine weiterführende Beschäftigung mit MINT-Themen zu motivieren (Höffler et al., 2011). Wie in diesem Camp sollen mit Science Camps häufig die Darstellung der naturwissenschaftlichen Denk- und Arbeitsweisen (s. auch inquiry-based learning) oder das Naturwissenschaftsverständnis (s. auch Nature of Science) vertieft vermittelt oder erfahren werden; gleichermaßen können diese Formate auch genutzt werden, um Lernenden Berufsperspektiven aufzuzeigen (vgl. z. B. Knox et al., 2003) oder um bestimmte Zielgruppen wie Mädchen (vgl. Cavanagh, 2007) oder leistungsschwächere Schüler\*innen (vgl. Höffler et al., 2011) zu fördern. Lindner und Kubat (2014) konnten in einer Fragebogenstudie mit Camp-Teilnehmenden zeigen, dass das Interesse bei den meisten Lernenden gestiegen ist und diese gern länger im Camp verweilt hätten. Ähnliches konnte auch von Knox et al. (2003) gezeigt werden, die beschreiben, dass Camp-Teilnehmende mit einem hohen Interesse an Naturwissenschaften im Anschluss an eine universitäts-konzipierte Sommerakademie zuversichtlicher in Bezug auf ihre experimentellen Fähigkeiten sind, ein nachhaltigeres Interesse an und berufliche Ausrichtung auf Naturwissenschaften bekunden sowie im Nachhinein selbst die Ausrichtung von Camps unterstützen.

#### Et kommunikationskoncept til praktisk erfaring, der forbinder fagrelateret viden og erhvervsorientering

Prof. Dr. Ilka Parchmann, Dr. Stefanie Herzog / IPN

Forskercamps eller science camps kan ses som en strategisk tilgang til uformelle læringsmuligheder eller læringsmuligheder uden for skoleregi, der giver mulighed for at øge og at opretholde elevernes interesse for naturvidenskab og at bevare deres egen identificering med naturvidenskab og matematik (Gibson & Chase, 2002). Disse mål korresponderer med de kontekstbaserede og kontekstorienterede læringsmål, der ligger til grund for PANaMa-projektets samlede koncept. Science camps som supplement til den regulære fagundervisning og skoleprojekter giver en masse muligheder netop med fokus på forbindelsen mellem faglig, skolerelevant viden og erhvervsorientering, som fx Knox et al., 2003 fortæller, igennem arbejde på autentiske steder, inddragelse af fageksperter eller udveksling med andre unge ud over ens egen klasse og peer group. Sidstnævnte er centralt for et EU-støttet Interreg-program, og netop i tyndt befolkede områder, hvor mangfoldigheden i de erhvervsmæssige perspektiver ofte ikke er synlige alene i ens egen region.

Særligt i USA er der en bred kultur inden for summer science camps, der består af strategiske tilgange med forskellig varighed, afsondrethed, målrettede aldersgrupper, typer af beskæftigelse og mål og inddragelse af videnskabelige institutioner (jfr. Fields, 2009). På den måde varierer disse camps fra endagsbesøg på uddannelsessteder uden for skoleregi til flere dages events på eller uden for skolen. Programmet NaWi-aktiv har fx udviklet og gennemført sommercamps som afslutning på arbejdet i arbejdsfællesskaber for at bruge det, man har lært forinden inden for rammerne af en fælles historie (jfr. Bertelsen, 2014: Storytelling-konceptet) og motivere til fortsat at beskæftige sig med MINT-emnerne (Höffler et al., 2011). På samme måde som på denne camp skal der på science camps ofte laves en uddybende formidling eller erfaring af, hvordan naturvidenskaben præsenterer sin måde at tænke og arbejde på (se også inquiry-based learning) eller af forståelsen af naturvidenskab (se også Nature of Science); disse formater kan også bruges på samme måde til at vise elever karriereperspektiver (jfr. Fx Knox et al., 2003) eller at fremme bestemte målgrupper som piger (jfr. Cavanagh, 2007) eller svagere elever (jfr. Höffler et al., 2011). Lindner og Kubat (2014) har lavet en spørgeskemaundersøgelse blandt deltagerne på en camp, som har vist, at de fleste elevers interesse steg, og de gerne ville have opholdt sig i længere tid på campen. Knox et al. (2003) har også kunnet vise noget lignende, der beskriver, at deltagere på et universitetsudviklet sommerakademi med en stor interesse for naturvidenskab er blevet mere optimistiske i forhold til deres eksperimentelle evner efter campen. De giver også udtryk for en vedvarende interesse for og professionel orientering mod naturvidenskab og efterfølgende selv har støttet afholdelse af camps.

Das hier konzipierte PANaMa Camp orientiert sich an den Zielen und Erfahrungen von Tirre et al. (2015):

Ziele von Tirre et al. , 2015	Ziele auf das Projekt PANaMa übertragen
Aufbereitung des Themenschwerpunkts Nanotechnolo- gie, so dass Lernende die Aktualität von Wissenschaft erfahren und reflektieren können	Aufbereitung des Themenschwerpunktes Digitalisie- rung und Automatisierung in Berufen als Basis einer eigenen Orientierung und Auseinandersetzung mit Be- rufsperspektiven für die teilnehmenden Schüler*innen
Anpassung eines Schülerlabors, so dass dieses auch in Schulen eingesetzt werden kann, welche nicht die Möglichkeit haben, ein bestimmtes Schülerlabor zu besuchen	Konzeption eines Camps für verschiedene Lernorte, hier beispielsweise als grenzübergreifendes Ferien- camp an einer Volkshochschule in Dänemark oder einem Schülerlabor in Deutschland
Schüler*innen sollen für zunehmende Bedeutung der Wissenschaftskommunikation sensibilisiert werden, indem eine Vielfalt an Kommunikationsmöglichkeiten aufgezeigt werden	Integration von Austausch- und Kommunikationsaspek- ten zwischen Schüler*innen verschiedener Länder, hier der Grenzregion Schleswig-Holstein und Süddänemark

Das PANaMa Camp mit dem Themenschwerpunkt Digitalisierung und Automatisierung sollte den Schüler\*innen dessen Aktualität in der heutigen Berufswelt aufzeigen. Die gezeigten Aktivitäten mit verschiedenen Robotern können so adaptiert werden, dass eine weiterführende Beschäftigung an Schulen und an außerschulischen Lernorten möglich ist. Der grenzüberschreitende Austausch zwischen deutschen und dänischen Schüler\*innen, der geographische, sprachliche und kulturelle Barrieren (seien diese empfundener oder realer Natur) abbauen bzw. überwinden soll, ist ein weiteres Ziel des Camps: Die dreitägige gemeinsame Arbeit an einem neuen Ort ermöglicht eine intensivere Auseinandersetzung mit dem Thema als die im normalen Unterricht. Als Sprache wurde Englisch gewählt. Die erste Erprobung zeigte, dass es den Acht- bis Zehntklässler\*innen überwiegend gut gelang, zu den präsentierten alltagsnahen Themen in ihrer ersten Fremdsprache verständlich miteinander zu kommunizieren. Durch eine Einteilung in Viererteams, wobei jeweils zwei deutsche und zwei dänische Schüler\*innen zusammenarbeiteten, wurde darauf geachtet, dass der Rückhalt der eigenen Sprache bei Nachfragen oder kultureller Herangehensweise Vertrautheit schaffte und zugleich kulturelle Unterschiede eher indirekt wahrgenommen und z. T. reflektiert wurden.

Der Ablauf des Camps ist in diesem Booklets beschrieben während das Kapitel "Robotertechnologie" aus dänischer Sicht eine Möglichkeit aufzeigt, wie dieses Thema in den Schulunterricht der Sekundarstufe I bzw. insbesondere der Klassenstufen 7–9 integriert werden kann. Die Ideen und Konzepte, die in dem Kapitel angeführt werden, wurden in dem Science Camp in Askov bereits in wesentlichen Teilen erprobt bzw. auch konkret umgesetzt. Wie die Arbeit mit Robotern und Drohnentechnik von den Lehrkräften, die die Schulklassen im März nach Askov begleitet haben, erlebt wurde, spiegelt sich in den kurzen Statements wider, die in Kapitel V zu lesen sind.

PANaMa Campen, som vi udviklede her, orienterer sig efter målene og erfaringerne fra Tirre et al. (2015).

Målene fra Tirre et al. , 2015	Målene overført til PANaMa-projektet	
Tilrettelæggelse af fokusområdet nanoteknologi, så eleverne kan få noget at vide om videnskabens aktuali- tet og reflektere over den.	Tilrettelæggelse af fokusområdet digitalisering og automatisering i forskellige erhverv som basis for selv at kunne orientere sig og sætte sig ind i og forklare for- skellige karriereperspektiver for de deltagende elever.	
Tilpasning af et elevlaboratorium, så det også kan bruges på skoler, der ikke har mulighed for at besøge et bestemt elevlaboratorium.	Udvikling og gennemførsel af en camp til forskellige læringssteder, her fx som grænseoverskridende fe- riecamp på en højskole i Danmark eller i et elevlabora- torium i Tyskland.	
Eleverne skal sensibiliseres over for den tiltagende betydning af videnskabelig kommunikation ved at vise dem en masse forskellige kommunikationsmuligheder.	Integrering af udvekslings- og kommunikationsaspek- ter mellem elever og forskellige lande, her grænseregio- nen Schleswig-Holstein og Syddanmark.	

PANaMa campen med fokusområderne digitalisering og automatisering bør vise eleverne deres aktualitet i erhvervslivet i dag. De viste aktiviteter med forskellige robotter kan adapteres på den måde, at det er muligt fortsat at beskæftige sig med dem på skoler og læringssteder uden for skolen. Udvekslingen over grænsen mellem danske og tyske elever, der nedbryder eller skal overvinde geografiske, sproglige og kulturelle barrierer (uanset om det er noget, man bare føler, eller om det er reelt), er endnu ét af campens mål. Det tre dage lange fælles arbejde på et nyt sted gør det muligt at beskæftige sig med og undersøge emnet mere intenst end i den normale undervisning. Engelsk er blevet valgt som sprog. Den første afprøvning viste, at det lykkedes 8.–10. klasse overvejende godt at forstå hinanden og kommunikere med hinanden på deres første fremmedsprog om de hverdagsemner, der blev præsenteret. Ved at inddele dem i hold på fire, hvor to tyske og to danske elever arbejdede sammen, blev der taget hensyn til, at det skabte fortrolighed, at de også kunne spørge på deres eget sprog og finde ud af den kulturelle fremgangsmåde og samtidig observere og delvist reflektere mere indirekte over kulturforskellene.

Campens forløb er beskrevet i denne booklet, mens kapitlet "Robotteknologi" ud fra et dansk synspunkt viser en mulighed for, hvordan dette emne kan integreres i undervisningen i skolen fra mellemtrinnet og op efter, men særligt i 7.–9. klasse. Væsentlige dele af ideerne og koncepterne, der er anført i det kapitel, er allerede blevet afprøvet og også helt konkret omsat på science campen i Askov. Det afspejles i de korte statements i kapitel V, hvordan arbejdet med robotter og droneteknik blev oplevet af de lærere, der var med skoleklasserne på camp i Askov i marts.

#### Literatur / Litteratur

Bertelsen, G. (2014): Ein rätselhafter Todesfall. Materialien für den naturwissenschaftlichen Unterricht. Hohengehren: Schneider-Verlag.

Cavanagh, S. (2007). Science camp: Just for the girls. Education Week, 26(45), 26-28.

Fields, D. A. (2009). What do students gain from a week at science camp? Youth perceptions and the design of an immersive, research-oriented astronomy camp. International Journal of Science Education, 31(2), 151–171.

Gibson, H. L., & Chase, C. (2002). Longitudinal impact of an inquiry-based science program on middle school students' attitudes toward science. Science Education, 86(5), 693–705.

Höffler, T. N., Lüthjohann, F., Bertelsen, G., Demuth, R., & Parchmann, I. (2011). Das Projekt NaWi-aktiv: Naturwissenschaftliches Lernen fördern-Selbstkonzepte stärken. ChemKon, 18(2), 71–75.

Knox, K. L., Moynihan, J. A., & Markowitz, D. G. (2003). Evaluation of short-term impact of a high school summer science program on students' perceived knowledge and skills. Journal of Science Education and Technology, 12(4), 471–478.

Lindner, M., & Kubat, C. (2014). Science Camps in Europe--Collaboration with Companies and School, Implications and Results on Scientific Literacy. Science Education International, 25(1), 79–85.

# II. Drei Tage mit Robotern und Drohnen



# II. Tre dage blandt robotter og droner



#### Schüler-Forschungscamp des Projekts PANaMa, März 2018

Dr. Marc Wilken / IPN

Drei Tage im winterlichen März an der Askov Hochschule mitten in der dänischen Landschaft Mitteljütlands, Zeit und Ort für ein vom Projekt PANaMa initiiertes Forschungscamp für 84 dänische und deutsche Schüler\*innen der Jahrgangsstufen 8 bis 10. Drei Tage in denen sich die Jugendlichen über Sprachbarrieren hinweg mit Robotern, Drohnen und deren Programmierung befassten, gefolgt von einem Tag mit Besichtigungen von Unternehmen in der Region Flensburg, in denen Robotertechnik und Automatisierung eine Rolle spielen.

Die Hochschule im dänischen Askov bot hierfür den passenden örtlichen Rahmen. Die Hochschule ist eine sogenannte Efterskole, eine freie Schulform in Dänemark, bei der Jugendliche nach der regulären Schulzeit gezielt in unterschiedlichen Bereichen gefördert werden. Der Schwerpunkt der Schule in Askov liegt dabei auf der musisch-gestalterischen Bildung. Die Teilnehmenden des PANaMa Camps konnten die Hochschule am zweiten Abend kennenlernen als sie von den dänischen Studierenden eine Führung über den Campus erhielten.

Die 84 Schüler\*innen von vier Schulen – zwei dänische Klassen der Ølgod-Schule in Mittel-Jütland sowie je eine deutsche Klasse der Gemeinschaftsschulen in Neumünster-Brachenfeld, Schönkirchen sowie der Lilli-Martius-Schule in Kiel - waren nach Askov gekommen, um sich intensiv mit der digitalen Zukunft im Arbeitsleben in Form von Robotern, Drohnen und automatisierten Produktionsprozessen zu befassen. Für die Berufsperspektiven von Schulabsolvent\*innen werden die Digitalisierung und Automatisierung der Arbeitswelt zunehmend zu zentralen Themen, da es kaum noch ein Berufsfeld gibt, in denen digitale Medien die Arbeit nicht verändern. Daher ist es umso wichtiger ein Grundverständnis für das Arbeiten mit diesen Medien bereits in der Schule zu vermitteln. Hieran anknüpfend befasst sich das Projekt PANaMa in seiner Arbeit schwerpunktartig ebenfalls mit der Frage, wie digitale Kompetenzen im naturwissenschaftlich-mathematischen Schulunterricht auch mit den berufsorientierenden Elementen verknüpft werden können. Eine Möglichkeit bietet ein Forschungscamp, wie diese Veranstaltung in Askov.

Eröffnet wurde das Camp von Prof. Dr. Ilka Parchmann, die in ihren Worten an die Jugendlichen besonders auf den Reiz des grenzübergreifenden gemeinsamen Arbeitens einging. Die drei Tage boten



Abb. 1: Dänische und deutsche Schüler\*innen bei der Arbeit mit Ozobot-Minirobotern

Fig. 1: Danske og tyske elever, der arbejder med Ozobot mini-robotter

#### Elev-forskningslejr i PANaMa-projektet, marts 2018

Dr. Marc Wilken / IPN



Tre dage i den vinteragtige marts måned på Askov Højskole midt i det jyske landskab – det var tiden og stedet for en af projekt PANaMa igangsat forskningslejr for 84 danske og tyske elever fra 8. til 10. klassetrin. Tre dage, i hvilke de unge mennesker henover sprogbarrierer kunne beskæftige sig med robotter, droner og programméring af disse. Derefter en dag med besigtigelse af virksomheder i region Flensborg, hos hvilke robotteknologi og automatisering spiller en rolle.

Askov Højskole havde de rigtige lokalitetsmæssige rammer for projektet, idet der er tale om en normal efterskole, den kendte danske skoleform, hvor unge efter deres regulære skoletid kan få målrettet assistance indenfor forskellige områder. Askovs fokuspunkt er i den forbindelse det musiske og formgivningsmæssige. Den anden aften havde deltagerne i PANaMa campen mulighed for at lære højskolen at kende i forb. m. en campusrundvisning, som de fik af de danske studerende.

De 84 elever fra fire skoler – to danske klasser fra Ølgod-Skolen i Midtjylland samt hhv. en tysk klasse fra de såk. Gemeinschafts-skoler i Neumünster-Brachenfeld og Schönkirchen samt fra Lilli-Martius-Schule i Kiel - var taget til Askov for at beskæftige sig med arbejdslivets digitale fremtid i form af robotter, droner og automatiserede produktionsprocesser. Digitalisering og automatisering i erhvervslivet spiller i stigende grad en central rolle for dimitterende skoleelevers jobperspektiver, idet der nærmest ikke findes nogen fag længere, hvor de digitale media ikke forandrer arbejdsprocesserne. Derfor er det så vigtigt at være i besiddelse af basal viden og forståelse omkring brugen af disse medier - og at denne viden allerede formidles i skoleundervisningen. Med reference til dette sætter projektet PANaMa ligeledes en særlig fokus på spørgsmålet, hvordan digitale kompetencer kan knyttes sammen med erhvervs- og jobvejledende aspekter i den naturvidenskabelige og matematiske skoleundervisning. Og én mulighed er f. eks. en forskningslejr som den, der foregik i Askov.

Prof. dr. Ilka Parchmann åbnede lejren. I sin tale til de unge kom hun også ind på, hvor charmerende hun syntes, det grænseoverskridende aspekt var i det fælles arbejde, som de unge skulle i gang med. Eleverne havde i disse tre dage lejlighed til at gå et skridt videre end deres basale kundskaber, idet de v. h. a. praktiske eksempler kunne lære noget om, hvordan digitale processer foregår og hvilke muligheder de giver. Man

den Jugendlichen die Gelegenheit über die bestehenden Basiskenntnisse hinaus, Möglichkeiten und Abläufe digitaler Prozesse an praktischen Beispielen kennenzulernen. Ein Einstieg bildete hier die Programmierung von Robotern und die praktische Anwendung von Drohnen. Erste Schritte in der Programmierung konnten mithilfe von Lego®-Mindstorm- und Ozobot-Robotern gemacht werden. Während viele Schüler\*innen bereits mit dem Mindstorm-Programm gearbeitet hatten, boten die kugelförmigen Miniroboter der amerikanischen Firma Ozobot einen ganz anderen und neuen Einstieg. Die Programmierung dieser kleinen Roboter ist dabei nicht nur über Computer, Tablets oder Handys möglich, sondern auch ohne jeglichen Bildschirm mittels Strichcodes, die auf weißes Papier gezeichnet werden. Über unterschiedliche Aufgabenstellungen erforschten die Jugendlichen die vielfältigen Einsatzmöglichkeiten und gewannen so erste Eindrücke, wie Robotertechnologie auch in Industrie oder Verkehr praktisch genutzt werden kann. Als drittes Arbeitsfeld wurde den Jugendlichen ein Forschungsprojekt des Drohnencenters an der Syddansk-Universität, Odense, vorgestellt, bei dem Drohnen Landwirten bei der Erfassung der zu erwartenden Kürbisernte helfen. Hierbei werden die Felder von den Drohnen gescannt und über eine spezielle Software dann die Anzahl der Kürbisse gezählt. Dies konnten die Schüler\*innen in einem Versuch simulieren, bei dem die Kürbisse durch Kunststoffbälle ersetzt wurden. Die Daten wurden nach der Erfassung durch die Drohne von den Jugendlichen ausgewertet. Mit diesem Beispiel aus der Landwirtschaft gab es einen guten Einblick, wie auch in traditionellen Berufsfeldern die Digitalisierung zu Veränderungen der Tätigkeiten führt.

Mit den Eindrücken und Erfahrungen der zwei Tage mit Roboterund Drohnentechnik starteten alle Teilnehmenden des PANaMa Camps am letzten Tag zu Unternehmensbesichtigungen Richtung Süden in den Raum Flensburg. Hier standen neben der Flensburger Schiffbau-Gesellschaft Industriebetriebe zur Auswahl, in denen automatisierte Produktionsprozesse oder eine robotergestützte Fertigung zu sehen war. Es konnten die Mitsubishi HiTech Paper Europe GmbH, die im Bereich Schleuderguss arbeitende M. Jürgensen GmbH und die Firma Anthon besucht werden. Letztere baut Produktionsstraßen zur Möbelherstellung u. a. für namhafte Unternehmen wie IKEA, wobei am Ende der Produktionsstraße zum Beispiel ein komplett verpackter Kleiderschrank steht. Was die Schüler\*innen mitnehmen konnten, waren konkrete Eindrücke einer zunehmend digitalisierten Arbeitswelt und den Blick in spannende Berufsfelder. Zudem konnten die Besuche Perspektiven eines grenzübergreifenden Arbeitsmarktes eröffnen, der insbesondere im High-Tech-Bereich auf Fachkräfte angewiesen ist.

Zum Abschluss des PANaMa Camps ging es für die teilnehmenden Jugendlichen vor allem um ausprobieren, entdecken, staunen und



Abb. 2: Schüler\*innen arbeiten mit Lego-Mindstorm-Robotern

Fig. 2: Studerende arbejder med Lego Mindstorm-robotter



startede med at programmére robotter og gøre praktisk brug af droner og de indledende skridt var i form af at programmere nogle robotter fra Lego®-Mindstorm og Ozobot. En hel del elever havde allerede erfaring med Mindstorm-programmet, hvorimod de kugleformede miniaturerobotter fra det amerikaniske firma Ozobot var noget helt nyt og ukendt. Programmering af disse små robotter er i den forbindelse ikke kun mulig v. h. a. computere, tablets eller mobiltelefoner, men også – helt uden brug af nogen form for skærm – gennem stregkoder, som tegnes på et hvidt stykke papir. I kraft af forskellige opgaver, som de unge fik, dykkede de ned i en masse alsidige anvendelsesmuligheder og fik på den måde nogle første indtryk af, hvorledes teknologi med robotter kan anvendes i praksis i industrien eller til at styre trafikken. Et tredje aktivitetsområde fik de unge præsenteret i forb. m. et forskningsprojekt ved Dronecentret hos Syddansk-Universitet i Odense – et projekt, hvor droner hjælper landmænd med at forudsé og registrere, hvordan den forestående græskarhøst bliver. I den forbindelse gennemfører dronerne en scanning af markerne, og v. h. a. noget en speciel software kan man tælle græskarene. Dette havde eleverne mulighed for at simulere i et forsøg, hvor der blev brugt bolde af plast i stedet for græskar. Efter at dronen havde registreret dataene, blev disse evalueret af eleverne. Dette eksempel fra landbrugsverdenen gav eleverne en god idé om, hvordan arbejdsgangene også indenfor et ganske traditionelt erhverv forandres af digitaliseringen.

Med disse indtryk og erfaringer fra to dage med robot- og droneteknologi tog alle PANaMa camp-deltagerne så hul på den tredje dag, hvor man tog sydpå til Flensborg-området for at besigtige nogle virksomheder. Udover skibsværftet Flensburger Schiffbau-Gesellschaft kunne deltagerne vælge imellem forskellige industrivirksomheder, hvor der anvendes automatiserede produktionsprocesser eller robotsupporteret produktion f. eks. papirfirmaet Mitsubishi HiTech Paper Europe GmbH eller M. Jürgensen GmbH, som arbejder med centrifugalstøbning, eller også firmaet Anthon. Sidstnævnte fremstiller produktionslinjer til møbelindustrien for kendte kunder, såsom IKEA, hvor der henne for enden af produktionslinjen eksempelvis kommer et komplet emballeret klædeskab ud som resultat. Eleverne fik her nogle helt konkrete indtryk af, hvordan arbejdslivet bliver mere og mere bliver digitaliseret, og hvilke spændende jobmuligheder der ligger i dette. Desuden viste virksomhedsbesøgene perspektiverne på et grænseoverskridende arbejdsmarked, hvor kvalificeret arbejdskraft især efterspørges indenfor de højteknologiorienterede fag.

Afslutningsvis i forb. m. PANaMa campen handlede det for elevernes vedkommende først og fremmest om at afprøve, opdage, blive forbavset og at kunne gennemskue. Eleverne kunne så møde fænomener, forsøg og love fra naturvidenskabens verden i forb. m. deres afsluttende besøg på Phänomenta-eksperimentariet i Flensborg.

begreifen. Naturwissenschaftliche Phänomene, Versuche und Gesetze konnten die Schüler\*innen bei dem abschließenden Besuch der Phänomenta in Flensburg kennenlernen.

Mit dem aus dem PANaMa-Projekt entstandenen Konzept für ein Forschungscamp, wie es die drei Tage in Askov und Flensburg stattgefunden hat, konnten wir zeigen, wie eine didaktische Vermittlung von berufsorientierten Wissensbereichen aussehen kann. Insbesondere die einzelnen naturwissenschaftlich-mathematischen Themenfelder, mit denen das PANaMa-Projekt arbeitet, erschließen sich den Lernenden eindringlicher durch die intensive Beschäftigung v. a. in Verbindung mit praktischen Elementen. Das Thema Robotik bot sich für das PANa-Ma-Camp an, da es als zukunftweisende Technologie sowohl Perspektiven in der Arbeitswelt bietet als auch einen Bezug zu Unterrichtseinheiten an Schulen mit Inhalten um Computertechnologie und -software, Programmierung oder innovative Techniken hat. Zudem ist das Interesse der Jugendlichen in den Jahrgangsstufen 8 bis 10 an Themen um digitale Medien vergleichsweise hoch. So bildet auch für das PANaMa-Projekt insgesamt die Arbeit mit Robotern ein zentrales Thema im letzten Jahr der Projektlaufzeit, das exemplarisch sowohl in die Lehrkonzepte als auch in die Fortbildungen für Lehrkräfte integriert werden kann. Dies soll in der Zusammenarbeit mit den Schulen weiter ausgebaut werden und als Angebot für weitere Projekte zur Verfügung stehen.

Das Resümee der Schüler\*innen war im Hinblick auf die Arbeit mit den Robotern und der Drohne überwiegend positiv, bei der Zusammenarbeit mit ihren dänischen Pendants hielt sich allerdings die bereits im Vorwege erkennbare Skepsis zu möglichen Sprachbarrieren. Eine wichtige Rolle spielten hierbei sicher auch andere Lerngewohnheiten und -praktiken im Nachbarland, die sich in der selbstständigeren Herangehensweise an die gestellten Aufgaben zeigte. Für das PANaMa-Projekt lieferten die drei Tage in Askov wichtige Anhaltspunkte für die Weiterentwicklung der berufsbezogenen Lehrkonzepte, die sowohl an Schulen sowie außerschulischen Lernorten eingesetzt werden können.

Gennem det af PANaMa-projektet resulterede koncept for en forskningslejr som disse tre dage i Askov og Flensborg har vi kunnet vise, hvad didaktisk formidling af joborienteret viden kan handle om. Netop de enkelte naturvidenskablig-matematiske emneområder, som PANa-Ma-projektet arbejder med, får eleverne en mere dybdegående forståelse af ved at beskæftige sig intensivt med emnerne, især, når dette sammenholdes med de praktiske dele af forløbet. Emnet robotteknologi var oplagt til at blive taget op i PANaMa campen, fordi der er tale om en fremadrettet teknologi både med perspektiver for arbejdslivet og med relation til undervisningen ude på skolerne – med indhold som edb-teknologi og software, programmering eller innovativ teknologi. Dertil kommer, at unge mennesker på klassetrin 8 til 10 i forvejen er meget interesserede i emneområder vedr. digitale medier. På den måde gælder det også for PANaMa-projekt samlet set, at arbejdet med robotter vil være et centralt emne i det sidste år af projektets løbetid og desuden noget, der som et godt eksempel kan indgå i læringskoncepter, men også i forb. m. videreuddannelsestiltag for lærere. Dette skal fortsættes og udvides i samarbejde med skolerne og være til rådighed som et tilbud også til andre projekter.

Arbejdet med robotter og droner har eleverne overvejende oplevet dette som noget positivt, men omkring samarbejdet med deres danske modparter fortsatte den fra starten udviste skepsis med at gøre sig gældende under forløbet. En væsentlig medvirkende årsag til dette har sikkert været, at man ikke talte samme sprog, men også, at der m. h. t. at (skulle) lære noget findes andre sædvaner og og praktiske fremgangsmåder i Danmark end i Tyskland, hvilket også gjorde udslaget i måden, man hhv. greb tingene an på. For PANaMa-projektet indeholdt de tre dage imidlertid vigtig læring og information m. h. t. den videre udvikling af erhvervs- og jobrelaterede lærekoncepter.

# III. PANaMa Camps in Bildern



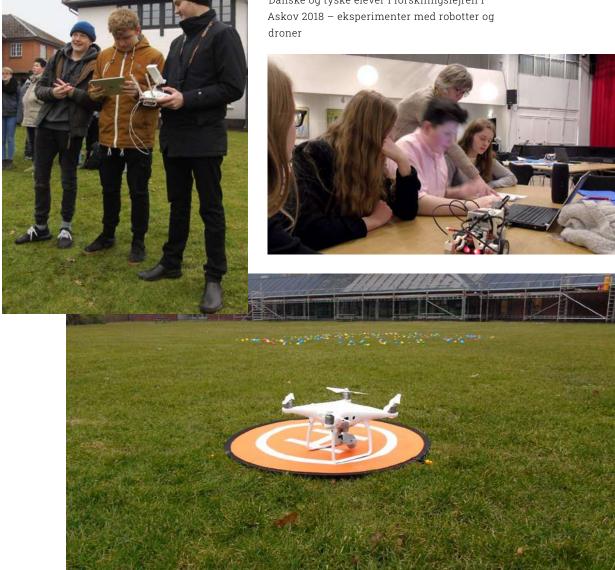
# III. PANaMa camps i billeder



### »Jeg kunne godt lide, at vi havde en seddel, hvor der stod alt det på, som vi skulle lave.«

Dänische und deutsche Schüler\*innen im Forschungscamp in Askov 2018 – Experimente mit Robotern und Drohnen

Danske og tyske elever i forskningslejren i droner









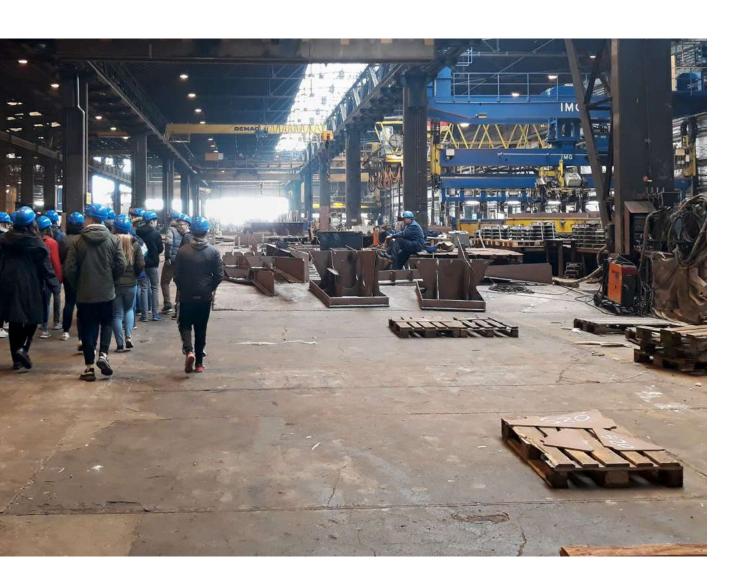
Unternehmensbesuche in der Region Flensburg im Rahmen des PANaMa Camps 2018

Virksomhedsbesøg i Flensborg-regionen som en del af PANaMa Campen 2018





»Roboter sind komplexe Geräte, man muss ihnen genaue Befehle geben, es kann manchmal sehr lange dauern und man muss viel ausprobieren.«

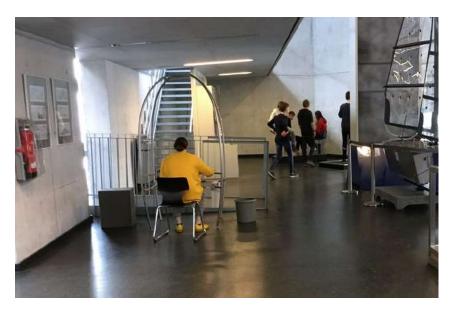


»Hvor meget arbejde det kræver at programmere en robot, at det er forskelligt fra robot til robot, hvordan man programmerer dem.«



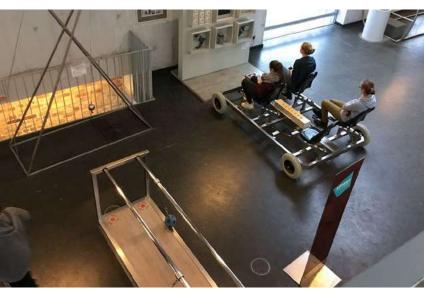








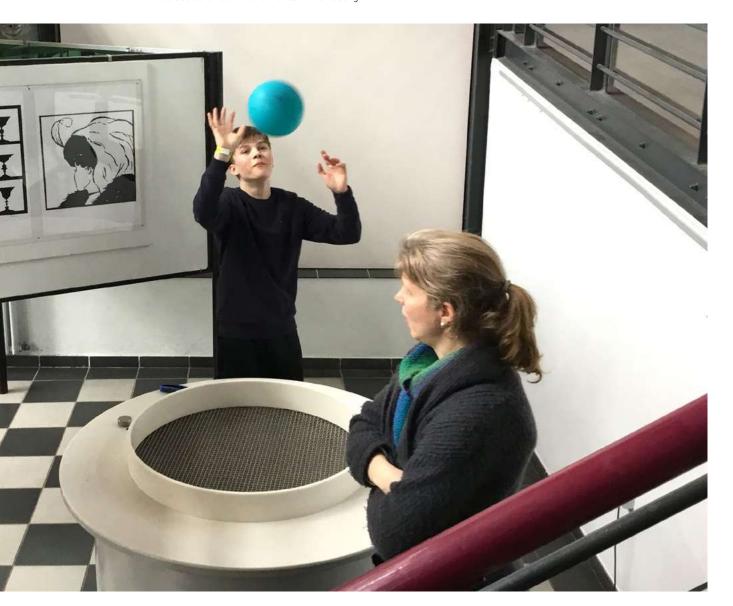




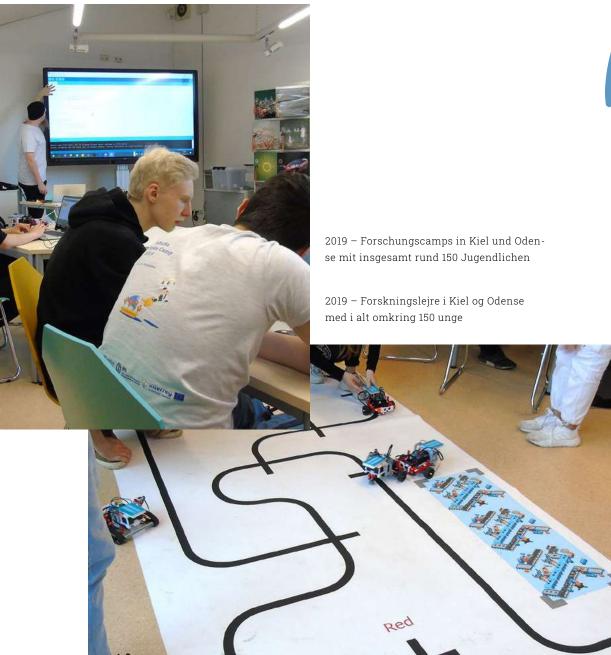
### »Ich werde mich im Anschluss noch mit dem Thema Roboter beschäftigen.«

Zum Abschluss des Forschungscamps 2018 besuchten die Schüler\*innen die Phänomenta in Flensburg

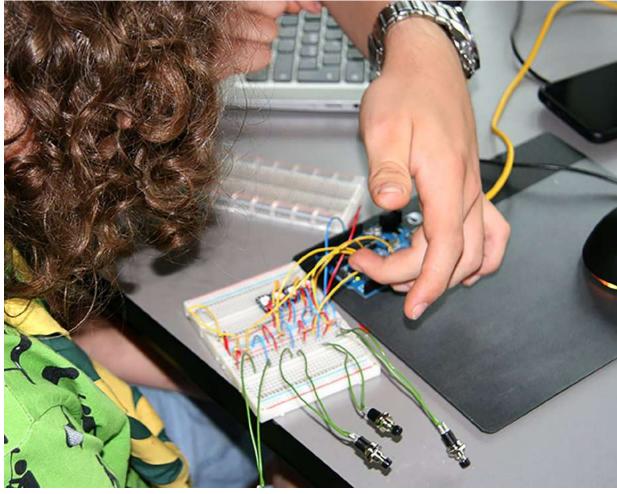
I slutningen af forskningslejren 2018 besøgte de studerende Phänomenta i Flensborg



## »Jeg vil også beskæftige mig med robotter efterfølgende. «







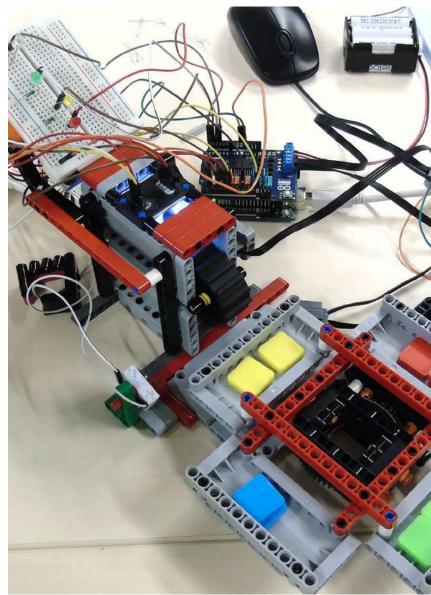


## »Mir hat gefallen, dass ich mich mit den Dänen verstanden hab. Dass wir mit den Ozobots, Pumpkins und Drohnen gearbeitet haben.«

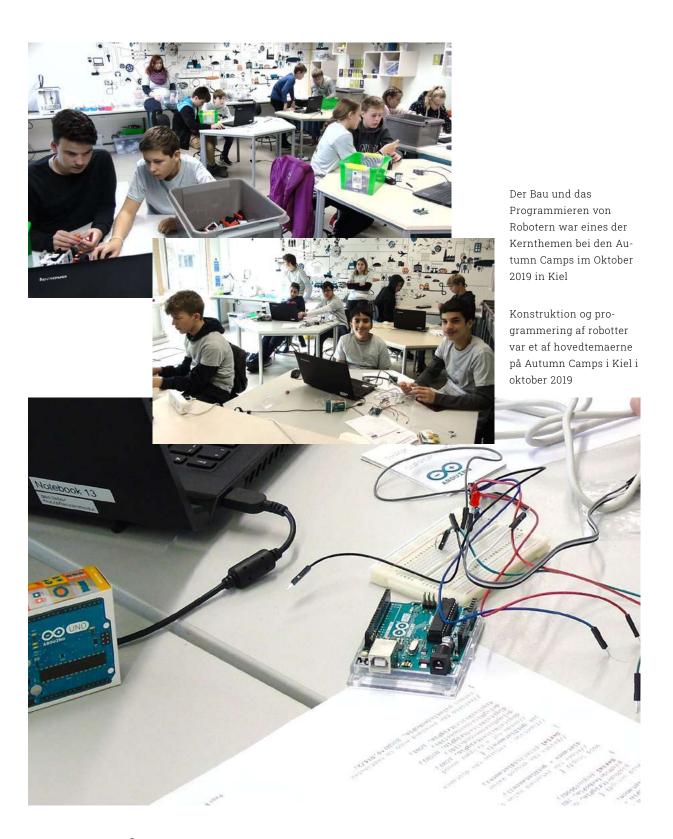
Autumn Camps 2019 in der Kieler Forschungswerkstatt zu den Themen Kryptographie und Robotik

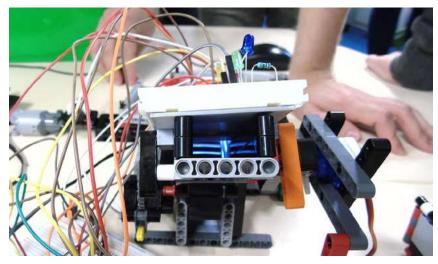
Autumn Camps 2019 i Kieler Forschungswerkstatt om emnerne kryptografi og robotik

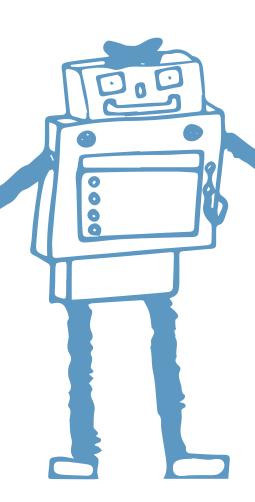




## »I learned a little bit more English.«









# IV. Autumn Camps in der Kieler Forschungswerkstatt



## IV. Autumn Camps i Kiels forskningsværksted



#### Außerschulische Lernangebote zur Kryptographie und Robotik

Dr. Stefanie Herzog / IPN

Da das jährliche Summer Camp der Teknologiskolen in Odense (http://www.teknologiskolen.dk/) einen stetig wachsenden Zuspruch erfährt, war die Überlegung, dieses Angebot auch Schüler\*innen im deutschen Projektgebiet nahezubringen. Realisiert werden konnte dies als ein zweiwöchiges Angebot in den Herbstferien 2019 in der im Botanischen Garten der Universität Kiel gelegenen Kieler Forschungswerkstatt (https://www.forschungs-werkstatt.de/). Die Einladung zur Teilnahme wurde an alle Kieler sowie zusätzlich Projekt-Schulen versendet, wodurch sich 49 Schüler\*innen verschiedener 7. – 9. Klassen angemeldet haben. Neben einem vergleichbaren Angebot zu dem der Teknologiskolen, in dem es darum ging, erste Erfahrungen beim Bauen und Programmieren eines Roboters zu sammeln, wurde den Jugendlichen auch ein Workshop zum Thema Kryptographie angeboten.

Im sogenannten "Robo-Camp" haben die Lernenden zunächst eine Einführung bekommen, was ein Roboter ist und aus welchen Teilen er aufgebaut ist. Auch wenn es dazu keine allgemeingültige Definition gibt, so lernten die Jugendlichen, dass Roboter i. d. R. als programmierbare Maschinen verstanden werden, die autonom funktionieren und ihre Aktionen auf Input aus der Umgebung abstimmen. Dazu besteht ein Roboter aus drei Hauptkomponenten, nämlich aus Sensoren, die bestimmte Aspekte der Umgebung wahrnehmen sollen, aus einem Mikrocontroller, der bestimmte vorher festgelegte Algorithmen abarbeitet und die Rechenleistung des Roboters übernimmt, und aus Aktoren, die bestimmte Handlungen umsetzen. Mögliche Sensoren sind Schalter, Licht-, Ultraschall-, Temperatur-, Farboder Bewegungssensoren sowie Mikrofone. Aktoren können z. B. Motoren, LEDs oder Lautsprecher sein. Der Mikrocontroller erhält ein Signal vom Sensor, z.B. "grüner Baustein vor dem Farbsensor", prüft in seinem Programm, was bei diesem Sensor-Input zu tun ist, und weist dann mit einem elektronischen Signal den Aktor an, eine bestimmte Antwort (Output) zu liefern, z. B. "Motor einschalten". Ebenso stellten die Jugendlichen Vermutungen auf, was ein Roboter nicht ist: Er ist sich seiner selbst nicht bewusst, ist nicht emotional, denkt nicht im eigentlichen Sinn, überlegt sich keine Konsequenzen zu seinen Aktionen und ist nicht kreativ – Roboter machen nur das, was ihnen einprogrammiert wurde und nicht mehr.

Um diese einzelnen Aufgaben zu verstehen, ging es am ersten Nachmittag darum, die Funktion der einzelnen Komponenten nachzuvollziehen.



Weißt Du eigentlich, wie ein Passwort deine Daten sichert? Ved du egentlig, hvordan en adgangskode sikrer dine data?



Weißt Du eigentlich, was das Schloss in deinem Browser bedeutet? Ved du egentlig, hvad låset i din browser betyder?



Weißt Du eigentlich, wie Roboter navigieren können? Ved du egentlig, hvordan robotter kan navigere?

Abb. 3: Die Fragen auf den Karten konnten die Jugendlichen, die an den Camps teilgenommen haben, nunmehr beantworten

Fig. 3: De unge, som deltog i lejrene, ved nu dette!

Weißt Du eigentlich, wie deine Handy-Kommunikation geschützt wird? Ved du egentlig, hvordan din kommunikation på mobilen bliver beskyttet?



Weißt Du eigentlich, wie Roboter-Systeme automatisiert werden können? Ved du egentlig, hvordan robottersystemer kan automatiseres?



Weißt Du eigentlich, wie Roboter Dinge sortieren und aussuchen können? Ved du egentlig, hvordan robotter kan sortere og udvælge ting?

#### Læringsmuligheder uden for skoleundervisningen for kryptografi og robotik

Dr. Stefanie Herzog / IPN

Den årlige sommerlejr, der gennemføres af Teknologiskolen (http://www.teknologiskolen.dk/), bliver mere og mere populær, og det frembragte idéen om også at muliggøre dette tilbud for skoleelever i det tyske programområde. Det skete i form af et to ugers forløb i efterårsferien 2019 på Forskningsværkstedet (https://www.forschungs-werkstatt.de/), der ligger i Kiels Universitets botaniske have. Invitationen til at deltage blev sendt til alle skoler i Kiel samt yderligere projektskoler, hvorefter i alt 49 elever fra forskellige 7. til 9. klasser tilmeldte sig. Ud over et tilbud, som kunne sammenlignes med Teknologiskolens, og hvor det handlede om at få indledende erfaringer med at bygge og programmere robotter, blev de unge også tilbudt en workshop om kryptografi.

I den såkaldte "Robo-Camp" fik eleverne oprindeligt en introduktion til, hvad en robot er og hvilke dele den består af. Selvom der ikke findes nogen generelt anvendelig definition, lærte de unge, at robotter normalt kan forstås som programmerbare maskiner, der fungerer autonomt og koordinerer deres handlinger i forhold til input fra deres omgivelser. Til dette formål består en robot af tre hovedkomponenter, nemlig sensorer, der skal opfatte visse signaler eller forhold fra omgivelserne, en mikrocontroller, der arbejder med visse foruddefinerede algoritmer og står for robottens computerydelse, og aktuatorer, der gennemfører bestemte handlinger. Mulige sensortyper er afbrydere, lys-, ultralyd-, temperatur-, farve- eller bevægelsessensorer samt mikrofoner. Aktuatorer kan f. eks. være motorer, LED'er eller højttalere. Mikrocontrolleren modtager et signal fra sensoren, f. eks. "Grøn byggeklods foran farvesensoren", checker i sit program, hvad den skal gøre i tilfælde af dette input til sensoren, og giver derefter gennem et elektronisk signal aktuatoren besked på at give et specifikt svar (output), f. eks. "Start motoren". Tilsvarende udviklede de unge også antagelser om, hvad en robot ikke er: Den er ikke bevidst om sig selv, har ingen følelser, har ingen tankevirksomhed i ordets normale forstand, overvejer ikke konsekvenserne af sine handlinger og er ikke kreativ. Det vil sige, at robotter kun gør det, hvad de er blevet programmeret til at gøre, og ikke andet eller mere.

For at forstå de enkelte opgaver, handlede den første eftermiddag om at blive klar over de enkelte komponenters funktion. Til dette formål dannede de unge grupper på tre, hvor én fik "rollen" som sensor, én anden som mikrocontroller og den tredje som aktuator. Målet var at indsamle tre krus hurtigst muligt pr. gruppe. Og det viste sig faktisk at være

Dazu wurden Dreiergruppen gebildet, wobei ein Jugendlicher den Sensor, einer den Mikrocontroller und einer den Aktor "spielen" sollte. Ziel war es, pro Gruppe möglichst schnell drei Becher einzusammeln. Durchaus herausfordernd war es dabei für die Jugendlichen, nur entsprechend ihrer Rolle zu agieren: Der "Sensor" musste dem "Mikrocontroller" berichten, was er sah, durfte aber keine Anweisungen geben, wie sich der "Aktor" bewegen sollte. Der "Mikrocontroller" selbst musste nach den Informationen des "Sensors" (Rückfragen zur Präzisierung waren erlaubt) Anweisungen an den "Aktor" geben, sah aber selbst nichts, da ihm die Augen verbunden waren. Der "Aktor" musste sich an die Anweisungen des "Mikrocontrollers" halten, auch wenn diese ihn weiter vom Becher wegführten.

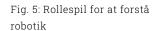
Bevor am nächsten Tag das eigentliche Wochenziel verkündet und dieses annähernd selbstständig erarbeitet werden konnte, mussten jedoch noch ein paar Vorarbeiten geleistet werden. Zunächst haben sich die Jugendlichen mit den möglichen Bestandteilen vertraut gemacht und herausgefunden, was Elektrizität überhaupt ist und wie die dabei genutzten Konzepte Spannung, Stromstärke und Widerstand zusammenhängen. Etwas praktischer ging es dann auch darum, wie z. B. eine LED in einen Stromkreis eingebaut wird, welche Arten von Widerständen es gibt und welche Funktion diese haben und wie man in einem Programm bestimmte Anweisungen an den Mikrocontroller überträgt. Ebenso beschäftigten sich die Schüler\*innen mit Programmieranweisungen, mit Variablen, Schleifen und Bedingungen sowie mit der logischen Abfolge von Schritten, um ein bestimmtes Problem zu lösen. Die erste mit einem selbst geschriebenen Programm zum Blinken gebrachte LED erzeugte dann auch bei allen Jugendlichen große Freude. Weiter ging es mit etwas komplexeren Programmierungen: Über einen Schalter wurde z. B. einem Motor kommuniziert, wann dieser sich drehen sollte. Über eine Bedingung wurde einer LED kommuniziert, dass diese z.B. nur leuchten sollte, wenn sich ein Gegenstand vor dem Sensor befand.

Am 2. Tag wurde das Wochenziel verkündet: In Partnerarbeit sollte entweder eine Farbsortiermaschine oder ein Transportroboter, der auf dem Boden aufgezeichneten Linien folgt, gebaut werden, die am Ende gemeinsam in einer hypothetischen Fabrik die farbigen Steine sortieren sollten. Durch Unterstützung von fünf studentischen Hilfskräften konnte in den darauf folgenden Tagen jede Gruppe eine Version einer Farbsortiermaschine oder eines Transportfahrzeugs konstruieren (vgl. auch Anleitungen im Kapitel "Technologieverständnis"), die dann am letzten Tag den Eltern in einer Abschlusspräsentation vorgestellt wurden. Jede\*r Jugendliche erhielt außerdem eine Teilnahmeurkunde.

Die Jugendlichen haben am Ende der Woche nicht nur einen funktionierenden Roboter gebaut, der mit denen der anderen Gruppen



Abb. 5: Rollenspiel zum Verständnis von Robotik





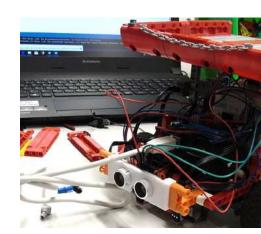




Abb. 4: Bau der Farbsortiermaschine

Fig. 4: Konstruktion af farvesorteringsmaskinen

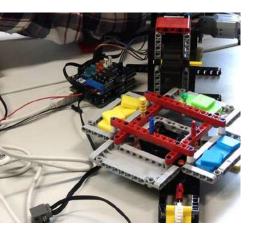




Abb. 6: Nach dem Zusammenbau des Transportroboters musste dieser noch programmiert werden

Fig. 6: Efter at transportrobotten var bygget, skulle den stadig programmeres

ganske udfordrende for de unge kun at måtte handle i henhold til deres rolle. "Sensoren" skulle rapportére til "mikrocontrolleren", hvad den så, men måtte ikke sige noget om, hvordan "aktuatoren" skulle bevæge sig. "Mikrocontrolleren" skulle efter "sensorens" oplysninger (som den havde lov til stille uddybende spørgsmål til) instruere "aktuatoren", men kunne ikke selv se noget, da den havde bind for øjnene. "Aktuatoren" skulle følge "mikrocontrollerens" instrukser, selvom det i givet fald betød, at den ikke nærmede sig kruset, men fjernede sig fra det.

Inden ugens egentlige mål blev annonceret den næste dag og herefter skulle udvikles nogenlunde selvstændigt, var der dog noget indledende arbejde, der skulle udføres af eleverne. Først gjorde de unge sig fortrolige med de mulige komponenter, prøvede at nå frem til, hvad elektricitet egentlig er for noget, og hvordan begreberne spænding, strøm og modstand hænger sammen med hinanden. På den mere praktiske plan handlede det så om, hvordan f. eks. en LED monteres i et el-kredsløb, hvilke typer modstande der findes, hvilken funktion de har, og hvordan man ved hjælp af et program overfører visse kommandoer til mikrocontrolleren. På samme måde beskæftigede eleverne sig med programmeringsinstrukser, variabler, sløjfer og betingelser samt den logiske rækkefølge af trin for at løse et specifikt problem. Den første LED, der kom til at blinke ved hjælp af et selvskrevet program, vakte da også stor glæde hos alle de unge. Som det næste drejede det sig så om mere komplekse programmeringer. Via en afbryder fik en motor f. eks. besked på, hvornår den skulle gå i omdrejninger. Via en betingelse blev det kommunikeret til en LED, at den f. eks. kun skulle lyse op, når der var et objekt foran sensoren.

Næste dag blev ugens mål forkyndt. I forbindelse med et makkerskab skulle der enten fremstilles en farvesorteringsmaskine eller en transportrobot. Sidstnævnte skulle følge nogle linjer, som var trukket på gulvet. I slutningen skulle enhederne kunne sortere kulørte klodser på en hypotetisk fabrik. Med støtte fra fem studentermedhjælpere havde hver gruppe i de næste dage mulighed for at konstruere en version af en farvesorteringsmaskine eller et transportkøretøj (jf. også vejledningen i kapitel "Teknologiforståelse"), som så blev præsenteret for forældrene på afslutningsdagen. Hver elev fik desuden udleveret et deltagerbevis.

Da ugen var omme, endte de unge ikke kun med at have lavet en funktionsdygtig robot, som var i stand til at sortere og transportere kulørte klodser, men mange deltagere kom også af med deres utryghed med at tale engelsk: I og med, at forløbet gennemførtes sammen med en dansk person, var lejrsproget vedrørende alle forhold omkring indhold og opfølgningsspørgsmål engelsk!

Som et biprodukt har de unge også erhvervet forskellige kompetencer, de kan drage nytte af både i skolen og senere i livet (både privat og professionelt). Udover evnen til at arbejde sammen med andre som gemeinsam farbige Steine sortieren und befördern konnte – viele von ihnen haben im Laufe der Woche auch Hemmungen gegenüber der englischen Sprache abgebaut: Da der Kurs von einem dänischen Kollegen durchgeführt worden ist, war die Sprache des Camps zu allen inhaltlichen Erklärungen und Nachfragen Englisch!

Quasi "ganz nebenbei" haben die Jugendlichen auch verschiedene Kompetenzen erworben, die ihnen sowohl in der Schule als auch im späteren Leben (privat und beruflich) nützlich sein werden: Neben der Fähigkeit zur Partner-/Teamarbeit und dem damit verbundenen Postulieren von Ideen, Argumentieren von Lösungsansätzen und Aushandeln von Kompromissen, stand die Problemlösekompetenz ganz oben auf der Liste. Dazu mussten die Schüler\*innen ein Problem zunächst verstehen und verbalisieren, dieses dann in kleinere Teile zerlegen und für die einzelnen Teile eine chronologisch abzuarbeitende Abfolge an Programmierschritten aufstellen. Diese Abfolge von Aktivitäten stellt dabei genau das algorithmische Denken dar, welches auch in allen Schulfächern genutzt wird: vom Lösen mathematischer Gleichungen über das Erstellen eines bestimmten Kunstwerkes bis hin zum Auswerten eines chemischen Elements müssen immer kleine Schritte in einer bestimmten Reihenfolge ausgeführt werden, damit eine Aufgabe erfüllt werden kann. Kommt man an einer bestimmten Stelle nicht weiter, muss sich der bzw. die Lernende überlegen, wo im Prozess man hängt und wie man das dann auftretende Problem lösen kann. Dieses lässt sich am Beispiel des Programmierens gut trainieren, da dort das Problem relativ niedrigschwellig ist, spielerisch angegangen wird und im Lösungsfall z. B. durch Blinken einer LED zu sofort sichtbaren Ergebnissen führt. Diese erworbenen Kompetenzen, die ausführlich auch im Kapitel "Technologieverständnis" dieses Bandes beschrieben sind, sind natürlich auch in jedem beruflichen Kontext von Vorteil - so müssen z. B. KfZ-Mechatroniker\*innen erkennen, warum der Motor im Auto nicht rund läuft, und entsprechend eine Lösungsstrategie entwickeln und testen, Landwirt\*innen überlegen, welche Kombination aus Wachstumsfaktoren eine ideale Ernte hervorbringen oder Zerspanungsmechaniker\*innen entscheiden, wieviel von welchem Material für die Erstellung eines bestimmten Präzisions-Bauteils eingesetzt werden soll.

Im sogenannten "Krypto-Camp" musste natürlich auch zunächst geklärt werden, was denn Kryptographie überhaupt ist. Die Schüler\*innen erfuhren, dass es in erster Näherung darum geht, dass ein Sender eine Nachricht an einen Empfänger schickt, die ein Unbefugter zwar lesen kann, aber nicht verstehen soll. Die Jugendlichen kamen sehr schnell darauf, dass dazu der Sender den Text einer Nachricht verschlüsseln müsste und diese verschlüsselte Nachricht, Chiffre genannt, dann verschickt wird. Der Empfänger muss dann natürlich den richtigen Schlüssel besitzen, um aus der Chiffre wieder den eigentlichen Text der Nachricht

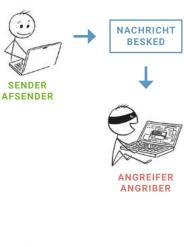
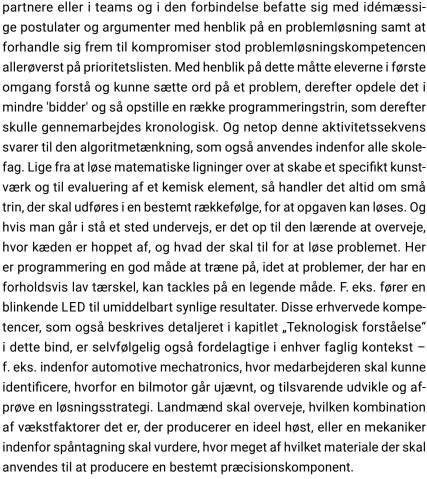
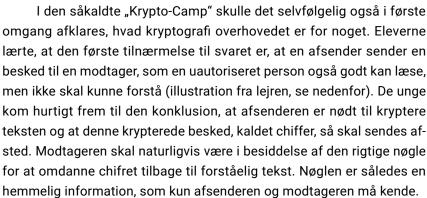




Abb. 7: Das Chiffrieren macht die Nachrichtenübermittlung sicherer

Fig.7: Kryptering gør transmission af meddelelser mere sikker





For at kunne vurdere, hvor sikre forskellige krypteringsmetoder er, påtog de unge sig i løbet af ugen gentagne gange rollen som skurk, der skulle forsøge at knække forskellige typer kryptering. Den første metode, der skulle gennemskues, var et princip, som allerede blev anvendt for over 2000 år siden, nemlig Cæsar-krypteringen. Hvert enkelt bogstav i beskeden erstattes her af et bestemt andet bogstav. Nøglen er et tal, der angiver antallet af karakterer, som alfabetet forskydes med. Hvis bogstav "a" forskydes med nøgleværdien 3, fremkommer bogstav "d". Hvis et bogstav flyttes ud over "z", starter alfabetet forfra. De unge startede





zu machen. Der Schlüssel ist somit eine geheime Information, die nur Sender und Empfänger bekannt sein darf.

Um beurteilen zu können, wie sicher verschiedene Verschlüsselungsverfahren sind, haben die Jugendlichen im Laufe der Woche immer wieder die Rolle eines Bösewichts eingenommen und versucht, diverse Verschlüsselungen zu knacken. Bei dem ersten Verfahren, was es zu knacken galt, handelte es sich um ein bereits vor über 2000 Jahren eingesetztes Prinzip, die Cäsar-Verschlüsselung. Dabei wird jeder einzelne Buchstabe der Nachricht durch einen bestimmten anderen Buchstaben ersetzt. Der Schlüssel ist eine Zahl, die angibt, um wie viele Stellen das Alphabet verschoben wird: Bei Verschiebung des Buchstabens "a" um den Schlüsselwert 3 erhält man Buchstaben "d". Bei einer Verschiebung über das "z" hinaus, schließt sich wieder das Alphabet an. Zunächst haben die Jugendlichen die Verschlüsselung mit der Cäsar-Scheibe ausprobiert: dabei stellten sie zuerst den Schlüssel ein. Dann konnten die Schüler\*innen für jeden Klartext-Buchstaben (außen) den verschlüsselten Buchstaben (innen) ablesen. Jetzt wurde natürlich ausprobiert, sich eigene kleine Nachrichten derart verschlüsselt hin und her zu schicken. Um wieder in die bereits erwähnte Bösewicht-Rolle zu schlüpfen, wurden diese geschriebenen Nachrichten abgefangen und versucht zu entschlüsseln. Dabei kamen die Schüler\*innen schnell auf eine Möglichkeit, sich den Schlüssel zu erschließen: alle 25 Verschiebungen im Alphabet ausprobieren und schauen, ob eine sinnvolle Nachricht dabei herauskommt. Die Jugendlichen erfuhren dann, dass das genauso auch in der damaligen Zeit passiert ist. Etwas systematischer kam man damals auf den Schlüssel, als man sich überlegt hat, dass ja in jeder Sprache bestimmte Buchstaben unterschiedlich häufig vorkommen - so ist in der deutschen Sprache das "e" der häufigste Buchstabe. Die Jugendlichen ermittelten nun in einer weiteren abgefangenen Nachricht den häufigsten Buchstaben in der Chiffre, welcher dann wahrscheinlich dem "e" entspricht - und schon war diese Verschlüsselung geknackt!

Am zweiten Tag des Camps konnten die Schüler\*innen das Verschlüsseln und anschließende Knacken der Cäsar-Verschlüsselung mit dem Computer ausprobieren. Dies wurde als authentischer für heutige Bösewichte empfunden, da diese sich vermutlich ja kaum die Mühe machen, alles per Hand auszuprobieren. Da aber bei der Nutzung des gewählten Programms von der "Alltags"-Sprache in Computersprache übersetzt werden musste, haben sich die Jugendlichen zunächst damit beschäftigt, wie sie das Problem so formulieren können, dass das Programm sie versteht. Dabei haben die Schüler\*innen jedem Buchstaben der Nachricht (1. Zeile in einer aus Übersichtszwecken kurz notierten Tabelle a) zunächst einen Wert zugeordnet von A=0 bis Z=25 (2. Zeile). Dann wird der Schlüssel ebenfalls in eine Zahl umgewandelt (3. Zeile).





Verschlüsselungstechniken

Fig. 8: Med Skytale og Cæsar-skiven lærte deltagerne gamle krypteringsteknikker med krypteringen med Cæsar-skiven. Først skulle nøglen defineres. Derefter kunne eleverne for hvert klartekstbogstav (ydre) aflæse det krypterede bogstav (indre). Derefter forsøgte eleverne selvfølgelig at sende egne små beskeder frem og tilbage på en sådan krypteret måde. For igen at smutte ind i den tidligere nævnte skurkerolle blev disse skriftlige beskeder opsnappet og naturligvis forsøgt at dekryptere. I den forbindelse fandt eleverne hurtigt ud af muligheden for at åbne systemet, idet de prøvede alle 25 forskydninger i alfabetet og fandt ud af, om der kom en meningsfuld besked ud af det. De unge fik så at vide, at man netop også gjorde dette dengang for 2000 år siden. Desuden kunne man – lidt mere systematisk - også finde frem til nøglen ved at tage højde for, at bestemte bogstaver forekommer med forskellig hyppighed i forskellige sprog. F. eks. er "e" det mest brugte bogstav i det tyske sprog. Så i forbindelse med en yderligere opsnappet meddelelse identificerede de (tyske) unge nu det mest brugte bogstav i chifferet, hvilket jo så sandsynligvis måtte svare til et "e" – og så var dén kryptering knækket!

Den næste dag var der lejlighed til at prøve at kryptere og efterfølgende knække Cæsar-krypteringen ved hjælp af en computer. Det syntes nutidens 'skurke' var mere autentisk, da nogen næppe ville gøre sig umage med at prøve sig frem til løsningen nu til dags. Men da det i forbindelse med brugen af det anvendte program var nødvendigt at oversætte "hverdagssprog" til computersprog, beskæftigede de unge sig først med, hvordan de kunne formulere problemet, så programmet forstod det. Eleverne tildelte først en værdi fra A=o til Z=25 (2. linje) til hvert bogstav i beskeden (1. linje i en tabel, som blev skrevet ned for at bevare overblikket). Derefter blev nøglen ligeledes konverteret til et tal (3. linje). Efter at disse to værdier bliver adderet kan der fremkomme tal, som er større end 25, det vil sige tal, der ikke er tildelt noget bogstav – svarende til flytning ud over "z", som nævnt ovenfor. Idéen at starte forfra på alfabetet klares så med metoden "division med rest". Hvis vi for eksempel deler 28 med antallet af bogstaver i alfabetet 26, fremkommer resten 2, 28 : 26 = 1 rest 2. Nu kan man bruge Cæsar-skiven til at kontrollere, at 'c', der er tildelt 2, faktisk er resultatet af forskydningen.

1.	Bogstav i beskeden	Н	E	U	Т	Е
2.	Bogstav svarer til tal	7	4	20	19	4
3.	Nøglebogstavets tal	8	8	8	8	8
		1	l	l		
4.	Sammenlagt, i alt	15	12	28	27	12
<b>4.</b> 5.	Sammenlagt, i alt  Rest ved division med 26	<b>15</b>	<b>12</b>	28	<b>27</b>	<b>12</b>

Tabelle a: Eksempel på en Cæsar-kryptering ved anvendelse af division med rest

Nach Addition dieser beiden Werte können nun Zahlen auftauchen, die größer als 25, also keinem Buchstabe zugeordnet sind. Dies entspricht der oben erwähnten "Verschiebung über das "z" hinaus". Das "Alphabet wieder anschließen" übernimmt nun die Division mit Rest. Teilen wir beispielsweise die 28 durch die Anzahl der Buchstaben im Alphabet 26, so erhalten wir den Rest 2, denn 28: 26 = 1 Rest 2. Mit der Cäsar-Scheibe kann nun überprüft werden, dass das der 2 zugeordnete "c" tatsächlich das Resultat der Verschiebung ist.

1.	Buchstabe der Nachricht	Н	E	U	Т	E
2.	Buchstabe entspricht Zahl	7	4	20	19	4
3.	Zahl des Schlüsselbuchstabens	8	8	8	8	8
4.	Summe	15	12	28	27	12
<b>4.</b> 5.	Summe  Rest bei Division durch 26	<b>15</b>	<b>12</b>	<b>28</b>	<b>27</b>	<b>12</b>

Tabelle a: Beispiel einer Cäsar-Verschlüsselung mit Hilfe der Division mit Rest

Mit dem Programm, mit dem die Verschlüsselung am PC durchgeführt wurde, haben sich die Schüler\*innen also nebenbei das mathematische Prinzip der Division mit Rest bzw. Modulo-Rechnung erarbeitet bzw. wiederholt. Auch die Schreibweise oder Verbalisierung selbst stellten hier für die Jugendlichen schon kleine Rätsel dar, denn längst nicht jeder (Erwachsene) versteht diese Ausdrucksweise – "28 modulo26 ist gleich 2" oder noch besser "28 (mod26) = 2" klingt doch tatsächlich ziemlich kryptisch, während "28 geteilt durch 26 hat den Rest 2" ja schon deutlich allgemeinverständlicher ist!

Die Cäsar-Verschlüsselung war mit und ohne Computereinsatz relativ leicht zu entschlüsseln, wenn die Jugendlichen eine Botschaft abfangen konnten. Daher haben sie sich am dritten Tag ein aus der Cäsar-Verschlüsselung weiterentwickeltes Verfahren erarbeitet, dass einfach dadurch sicherer wird, dass nicht mehr alle Buchstaben der Nachricht mit dem gleichen Schlüsselwert verschoben werden sondern mit unterschiedlichen. Um sich diesen Schlüssel dann besser merken zu können, wurde dieser in ein Wort umgewandelt. Unter die Nachricht haben die Schüler\*innen zum Verschlüsseln das Schlüsselwort dann so oft hintereinander geschrieben, bis jedem Nachrichten-Buchstaben ein Schlüsselwortbuchstabe zugeordnet war. Dann musste wieder jeder Buchstabe des Schlüsselworts in eine Zahl umgewandelt und mit dieser Zahl der Nachrichten-Buchstabe vergleichbar der Cäsar-Verschlüsselung codiert werden. Die Jugendlichen haben festgestellt, dass bei dieser neuen Methode, Vigenère-Verschlüsselung genannt, die Sicherheit von der Länge des Schlüsselworts und der Länge der Nachricht abhängt: Insbesondere bei kurzen Nachrichten und langen Schlüsselwörtern ist ein reines Ausprobieren aller möglicher Schlüssel zum Knacken der Chiffre nicht hilfreich - so müssten bei einem hypothetischen 7-stelligen Schlüsselwort ganze 8.031.810.176 (= 26 × 26 × 26 × 26 × 26 × 26) mögliche Buchstabenkombinationen getestet werden. Auch die bei der Cäsar-Verschlüsselung noch mögliche Häufigkeitsanalyse funktionierte jetzt nicht mehr so einfach, da ja gleiche Buchstaben nicht immer mit dem gleichen Schlüsselwert verschoben

Ved hjælp af programmet, som krypteringen blev udført med på pc'en, fik eleverne samtidig lært eller gentaget det matematiske princip med division med rest eller såk. modulo-regning. Selve skrivemåden eller verbaliseringen var såmænd også lidt af en gåde for de unge, fordi langt fra alle (voksne) forstår dette udtryk: "28 modulo26 er lig med 2" eller endnu bedre "28 (mod26) = 2". Udtrykket lyder jo faktisk ganske kryptisk, mens "28 divideret med 26 har resten 2" er meget mere alment forståeligt!

Cæsar-krypteringen var forholdsvis nem at dekryptere med og uden brug af computer, hvis de unge kunne opsnappe en besked. Derfor (videre) udviklede de på tredjedagen en fremgangsmåde med udgangspunkt i Cæsar-krypteringen, der simpelthen gøres mere sikker ved ikke at flytte alle bogstaver i beskeden med den samme nøgleværdi, men med forskellige. For bedre at kunne huske denne nøgle, blev den konverteret til et ord. Neden under beskeden skrev eleverne derefter krypteringsnøgleordet så mange gange efter hinanden, indtil der var tildelt et nøgleordsbogstav til hvert beskedbogstav. Derefter måtte hvert bogstav i nøgleordet igen konverteres til et tal, som beskedbogstavet blev kodet til på en i forhold til Cæsar-krypteringen tilsvarende måde. De unge fandt frem til, at sikkerheden med denne nye metode, kaldet Vigenère-kryptering, afhænger af længden af nøgleordet og længden af selve meddelelsen. lsær ved korte beskeder og lange nøgleord er der simpelthen ikke noget håb i at prøve sig frem til at knække chifferet med alle mulige nøgler. For et hypotetisk antaget 7-cifret nøgleord ville man skulle prøve sig igennem hele 8.031.810.176 (= 26 × 26 × 26 × 26 × 26 × 26 × 26) mulige bogstavkombinationer. Hyppighedsanalysen, der fungerede godt nok i forbindelse med Cæsar-krypteringen, fungerede heller ikke længere så godt, da de samme bogstaver ikke altid blev forskudt med den samme nøgleværdi. Denne type kryptering blev naturligvis bagefter ligeledes afprøvet i praksis. De unge gav hinanden små hovedbrudsopgaver, hvor enten beskeden og nøglen var kendte og beskeden skulle krypteres, eller chiffer og nøgle var kendte og beskeden skulle dekrypteres, eller besked og chiffer var kendte og nøglen skulle findes.

På fjerdedagen gik turen så igen langt tilbage i fortiden – til en metode, som allerede blev brugt før Kristi fødsel, nemlig brugen af en skytale (udtales dansk, ikke engelsk). Navnet er græsk og betyder pind eller stav. En blank papir- eller læderstrimmel blev viklet spiralformet omkring en sådan stav med en bestemt diameter, hvorefter man skrev vandret på strimlen. Den blev derefter viklet af staven, hvorefter bogstaverne tilsyneladende stod i en tilfældig rækkefølge. De unge prøvede forskellige cylindre til at kryptere små beskeder og fandt ud af, at forskellige cylinderdiametre førte til forskellige chifre. Det vil sige, at man skal kende den rigtige afkodningsdiameter for at kunne se, hvilke bogstaver der skal læses i hvilken rækkefølge.

Som afslutning og opsamling fremstillede de unge nogle plakater om eftermiddagen, som kunne præsenteres for deres forældre dagen efter. Ud fra en samlet vurdering af plakaterne kunne det konkluderes, at de tre gennemgåede metoder kunne opdeles i to områder: Cæsar & Vigenère leverede den krypterede tekst ved at skifte bogstaverne ud og hører således ind under substitutionsmetoderne (latin: "substiturere" = "at udskifte"), hvorimod skytalen decideret bytter rundt på bogstaverne og dermed hører til permutationsmetoden (latin: "permutare" = "at bytte rundt på"). Men for alle tre metoder gælder, at den hemmelige nøgle stadig skal formidles fra afsenderen til modtageren. Jamen, hvis man har en sikker måde at sende nøglen på, kunne man jo lige så godt bruge denne metode til med det samme at sende selve beskeden, uden

worden sind. Auch diese Art der Verschlüsselung wurde im Anschluss natürlich praktisch geübt: Die Jugendlichen stellten sich gegenseitig kleine Knobel-Aufgaben, indem entweder Nachricht und Schlüssel vorgegeben wurden und die Nachricht verschlüsselt werden sollte, oder Chiffre und Schlüssel vorgeben waren und die Chiffre entschlüsselt werden musste oder auch bei vorgegebener Nachricht und Chiffre der Schlüssel gefunden werden sollte.

Am vierten Tag ging es noch einmal ganz tief in die Vergangenheit, zu einem Verfahren, dass schon vor Christi Geburt genutzt worden ist: die Skytale. Der Name kommt aus dem Griechischen und bedeutet Stock bzw. Stab. Um eben einen solchen Stab mit einem bestimmten Durchmesser wurde schraubenförmig ein leerer Papier- oder Lederstreifen gewickelt und dann horizontal beschrieben. Anschließend wurde der Streifen abgewickelt und die Buchstaben waren dann in vertauschter Reihenfolge notiert. Die Jugendlichen probierten verschiedene Zylinder aus, um so kleine Botschaften zu verschlüsseln und fanden heraus, dass die unterschiedlichen Durchmesser der Zylinder zu unterschiedlichen Chiffren führten: man muss also beim Entschlüsseln den Durchmesser kennen, um zu wissen, welche Buchstaben in welcher Reihenfolge gelesen werden müssen.

Als eine Zusammenfassung erstellten die Jugendlichen am Nachmittag Poster, die am folgenden Tag ihren Eltern präsentiert werden konnten. So ließ sich aus dem Gesamtbild der Poster ableiten, dass die drei durchgenommenen Verfahren in zwei Bereiche eingeteilt werden konnten: Cäsar & Vigenère lieferten durch ein Austauschen von Buchstaben den verschlüsselten Text, gehören daher zu den Substitutionsverfahren (lat. "substiturere" = "austauschen"); während die Skytale ein Vertauschen von Buchstaben vornimmt und damit zu den Permutations-Verfahren gehört (lat. "permutare" = "vertauschen"). Bei allen drei Verfahren muss aber immer noch der geheime Schlüssel vom Sender an den Empfänger übermittelt werden – aber wenn man mit einem bestimmten Verfahren den Schlüssel sicher tauschen kann, dann könnte man mit diesem Verfahren doch auch gleich die ganze Nachricht sicher tauschen und bräuchte gar keine Verschlüsselung! Von den drei Verfahren ist die Vigenère-Verschlüsselung das sicherste Verfahren, da insbesondere bei langen Schlüsselworten ein Knacken per Häufigkeitsanalyse kaum möglich ist. Gleichzeitig hat es immer noch das Problem, dass für jede neue Verschlüsselung zur Sicherheit natürlich ein neuer Schlüssel erstellt und übermittelt werden muss. Eigentlich wünscht man sich ja, dass man den gleichen Schlüssel (bzw. das gleiche Passwort) mehrmals verwenden kann. An die Lösung dazu haben sich die Jugendlichen am fünften Tag angenähert, bevor sie final ihr erarbeitetes Wissen präsentieren konnten. Durch die Einteilung einer Nachricht in Blöcke und die Anwendung verschiedener Verschlüsselungsansätze, auch mehrerer nacheinander, auf

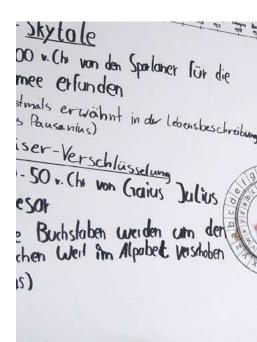
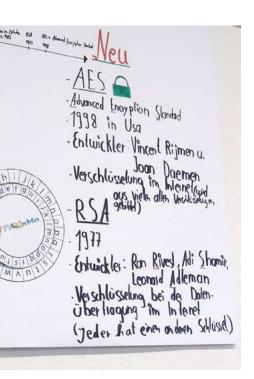


Abb. 9: Jugendliche gestalteten Poster mit den erlernten Verschlüsselungstechniken

Fig. 9: Deltagerne designede plakater med de krypteringsteknikker, de havde lært



at have brug for nogen kryptering! Den sikreste af de tre metoder er Vigenère-krypteringen, da det næppe er muligt at knække den ved hjælp af hyppighedsanalysen, især med lange nøgleord. Samtidig findes der stadig problemet, at der for hver ny kryptering for en sikkerheds skyld selvfølgelig skal oprettes og sendes (!) en ny nøgle. Men rent faktisk ville man jo ønske at kunne bruge den samme nøgle (eller det samme kodeord) flere gange. De unge tilnærmede sig løsningen på dette på femtedagen, inden de til sidst kunne præsentere deres opnåede viden. Ved at opdele en meddelelse i blokke og anvende forskellige krypteringsmetoder, endda flere efter hinanden til de forskellige blokke, krypteres delene af meddelelsen langt mere sikkert. De krypteringsmetoder, der blev introduceret i løbet af ugen, er substitution og permutation, men der findes også andre metoder, som er markant mere komplekse at forklare og forstå. Men det centrale ved dem alle sammen er, at der altid bliver skiftet informationer ud eller at der bliver byttet rundt på dem. Sagt på en lidt forenklet måde, er moderne kryptering en smart kombination eller rækkefølge af Cæsar, Vigenère og brugen af en skytale. Denne type blok-efter-blok-kombination af forskellige krypteringsmetoder, også kendt som Advanced Encryption Standard, AES, altså avanceret kryptering, forekommer i næsten alle sikkerhedssystemer for digital kommunikation nu til dags, f. eks. WiFi, Messenger eller netbanking, og er i modsætning til de historiske processer, som de unge lærte at kende, også hele tiden vokset i takt med computernes stadigt stigende regnekapacitet.

Den i løbet af lejropholdet opnåede forståelse vedrørende sikker overførsel af information, men også oplevelserne i forbindelse med den gennemførte Robo-Camp, kan være nyttige for de unge både i privatlivet og i det senere arbejdsliv. Autonome maskiner bliver anvendt indenfor en lang række fag. F. eks. indlæser mælkerobotter selvstændigt stregkoder fra køernes øremærker, måler derefter både mælkemængden og fedtindholdet og gemmer disse data sikkert i en database. Ved hjælp af sammenligninger med eksisterende data kan computeren identificere afvigelser i mælkemængden eller mælkesammensætningen for hver ko og således optimere fodersammensætningen eller identificere sygdomme på et tidligt tidspunkt. Lignende autonome processer kan findes i forbindelse med vindmøller, lagerlogistik, overvågning af bilmotorer o. s. v. Og basis for denne velfungerende kommunikation fra maskine til maskine er en pålidelig og ikke manipulerbar overførsel af information, både til dataindsamling ved hjælp af scannere og dataoverførsel via radionetværk. I lejren fik de unge en fornemmelse af, hvordan man tidligere forsøgte at videregive visse oplysninger (ofte af militær karakter), så de ikke kunne opsnappes, aflyttes eller ændres. Imidlertid kunne de unge også ved selvsyn konstatere, at denne type kryptering ikke længere er tilstrækkelig, eftersom der i dag findes højtydende computere. I dag bruger man die verschiedenen Blöcke werden die Teile der Nachricht deutlich sicherer verschlüsselt. Zwei dieser Verschlüsselungsansätze sind die bereits in dieser Woche kennengelernten Substitution und Permutation; es gibt aber auch noch weitere, die in der Erklärung deutlich komplexer sind. Der Kern dieser Verschlüsselungsansätze ist allerdings immer das Austauschen, oder das Vertauschen von Informationen. Vereinfacht gesagt ist also eine modern Verschlüsselung eine geschickte Aneinanderreihung von Cäsar, Vigenère und Skytale. Diese Art der blockweisen Kombination verschiedener Verschlüsselungsverfahren, auch Advanced Encryption Standard bzw. AES oder auf Deutsch etwa "fortschrittlicher Verschlüsselungsstandard" genannt, kommt heute in fast allen Sicherungsverfahren digitaler Kommunikation vor, bspw. WLAN, Messenger, Banking und ist im Gegensatz zu den kennengelernten historischen Verfahren auch stets mit der immer größer werdenden Rechenleistung von Computern gewachsen.

Das in diesem Camp erworbene Verständnis für die Übermittlung sicherer Informationen kann den Jugendlichen ebenso wie die Erfahrungen zum Robo-Camp sowohl im Privatleben als auch im späteren Berufsumfeld hilfreich sein: in diversen Berufsfeldern sind viele autonome Maschinen im Einsatz. So lesen z. B. Melkroboter die Strichcodes auf den Ohrmarken der Kühe selbstständig ein und messen anschließend sowohl Milchmenge als auch Fettgehalt und legen diese sicher Daten in einer Datenbank ab. Über Abgleiche mit vorhandenen Daten können Computer für jede Kuh Abweichungen bei der Milchmenge oder -zusammensetzung identifizieren und so die Futterzusammensetzung optimieren oder frühzeitig Krankheiten erkennen. Vergleichbare autonome Prozesse finden sich bei Windkraftanlagen, in der Lagerlogistik, beim Fahrzeugmotorenmonitoring, etc. Grundlage dieser funktionierenden Maschine-Maschine-Kommunikation ist eine zuverlässige und von außen nicht manipulierbare Informationsübermittlung sowohl bei der Datenerfassung über Scanner als auch bei der Datenweitergabe über Funknetze. Im Camp erhielten die Jugendlichen ein Gespür dafür, wie in der Vergangenheit versucht worden ist, bestimmte Information (oft militärischer Art) abhör- und veränderungssicher übermitteln zu können. Ebenso konnten die Jugendlichen aber selbst ausprobieren, dass mit Einführung rechenstarker Computer diese Art der Verschlüsselung nicht mehr ausreicht und heute viele verschiedene Kombinationen eingesetzt werden, um unbefugte Leute am Auslesen oder Manipulieren von Daten zu hindern. Auch in diesem Camp konnten somit wichtige Kompetenzen vermittelt werden, die die Schüler\*innen sowohl für den privaten (Sicherheit von Textnachrichten mit einem Messenger) als auch den späteren beruflichen Kontext (Sicherheit von Arbeitsmails oder Datenübertragung zwischen Maschinen) sensibilisieren konnten.

Einige Jugendliche haben das Camp in beiden Wochen besucht, wobei sie sich in der einen Woche mit Robotern und in der anderen mit Verschlüsselung beschäftigt haben. Neben dem Inhalt standen in beiden Wochen durch kleine Spiele, u. a. zum Kennenlernen des Botanischen Gartens, auch das Gemeinschaftsgefühl als Gruppe und durch den täglichen Mensabesuch zum Mittagessen auch der Aspekt des "ungewohnten Umfelds Universität" auf dem Programm.

derfor mange forskellige kombinationer til at forhindre uautoriserede personer i at læse eller manipulere data. I denne camp kunne der derfor også formidles vigtige kompetencer, som sensibiliserer eleverne for emnet både indenfor privatlivet (sikkerhed for tekstbeskeder i et messenger-system) og senere i en jobmæssig kontekst (e-mail-sikkerhed på arbejdet eller i forbindelse med dataoverførsel mellem maskiner).

Nogle af de unge var med i lejren i begge uger – og beskæftigede sig med robotter i den ene uge og kryptering i den anden. Udover selve læringsindholdet havde begge uger også små lege og spil på programmet, f. eks. i forbindelse med at lære den botaniske have at kende. Her drejede det sig især om følelsen af at være sammen om noget i en gruppe. Også det daglige besøg i universitetskantinen (mensa'en) til frokost gav deltagerne en fornemmelse for det "uvante universitetsmiljø".

## V. Aus der Sicht von Lehrkräften und Schüler\*innen



## V. Fra lærernes og elevernes perspektiv



#### Das meinten die Lehrerinnen und Lehrer im Nachhinein zum PANaMa Camp 2018

#### Was war Ihre Motivation, mit Ihrer Klasse am Camp teilzunehmen? (Erwartungen)

Meine Motivation, mit meiner Klasse am Camp teilzunehmen lag darin, dass dadurch die Schülerinnen und Schüler einen Einblick in das Thema Robotik und damit verbundene Berufsperspektiven erhalten konnten und gleichzeitig den Jugendlichen die Möglichkeit gegeben werden konnte, sich einfach und unkompliziert mit dänischen Jugendlichen auszutauschen.

Ich wollte mit meinen Schüler\*innen am Camp teilnehmen, weil es sich spannend anhörte und zudem kostenfrei war, was für viele bedürftige Schüler\*innen ein Geschenk war. Ich selbst spreche Dänisch und reise seit vielen Jahren in das Land, das ich sehr mag.

Den Schüler\*innen mit Hilfe der Ozobots das Programmieren näher zu bringen, war eine große Motivation. Die Möglichkeit, dabei auf andere Schüler\*innen aus dem In- und Ausland zu treffen und mit denen zusammenarbeiten zu dürfen/müssen, war ein weiterer Anreiz. Der Einblick in Betriebe war ebenfalls interessant. Natürlich war auch die Möglichkeit, der Klasse einen Ausflug zu bieten, eine große Motivation.

Ich fand es spannend zusammen mit den Schüler\*innen neue Wege in naturwissenschaftlichen Fächern zu finden.

#### Haben sich die Erwartungen erfüllt?

Für den kurzen Zeitrahmen der zur Verfügung stand, haben die Schüler\*innen viele Erfahrungen machen dürfen und hatten vor allem in den Workshops gute Gelegenheiten, sich sowohl mit dem Thema Coding zu beschäftigen als auch nette Gespräche mit den dänischen Schüler\*innen führen zu können.

Ich hätte mir mehr Unterstützung gewünscht und weniger Einbindung der begleitenden Lehrkräfte. Gefühlt hatte man während der Fahrt keine ruhige Minute, das war extrem anstrengend.



Abb. 10: Arbeiten mit den Ozobot-Minirobotern in Askov 2018

Fig. 10: Arbejde med Ozobot-minirobotterne i Askov 2018

#### Det mente lærerne efterfølgende om PANaMa Campen 2018

## Hvad var din motivation for at deltage i campen med din klasse? (Forventninger)

Min motivation for at deltage i campen med min klasse lå i, at eleverne derigennem kunne få et indblik i robotteknik og de karrieremæssige perspektiver, der er forbundet med det, men også at de unge samtidig fik mulighed for at udveksle ideer med unge fra Danmark på en enkel og ukompliceret måde.

Jeg ville gerne deltage i campen med mine elever, fordi det lød spændende og desuden var gratis, hvilket var en gave for mange elever, som ikke ellers ville have haft råd til det. Jeg taler selv dansk og har rejst til Danmark i mange år. Det kan jeg godt lide.

Det var en stor motivation for mig at kunne gøre eleverne mere fortrolige med at programmere ved hjælp af ozobots. Det var også en motivation for mig at få mulighed for at møde andre elever fra ind- og udland dér og at måtte/skulle arbejde sammen med dem. Det var også interessant at få et indblik i virksomhederne. Det var naturligvis også en stor motivation at kunne tilbyde klassen en udflugt.

Jeg synes, det var spændende at opdage nye veje i de naturvidenskabelige fag sammen med eleverne.

#### Blev dine forventninger opfyldt?

I forhold til hvor kort tid der stod til rådighed, fik eleverne lov til få mange forskellige oplevelser, og især til workshopperne havde de god mulighed for både at beskæftige sig med kodning og have gode snakke med de danske elever.

Jeg kunne godt have ønsket mig noget mere hjælp og mindre inddragelse af de lærere, der var med. Man følte ikke, at man havde et eneste minuts ro på turen derhen. Det var meget anstrengende.



## Was kann Ihrer Meinung nach speziell ein solches Camp leisten/vermitteln?

Ein solches Camp bietet die Möglichkeit sich mit Themenfeldern zu beschäftigen, die üblicherweise im Regelunterricht zu kurz kommen oder nicht angesprochen werden.

Neben naturwissenschaftlichen und technischen Kompetenzen ist so ein Camp ideal, um Kommunikations- und Sprachkompetenzen zu fördern. Für meine Schüler\*innen bot es zudem die Möglichkeit, das Nachbarland Dänemark mal aus einer anderen Perspektive als die des Urlaubers kennen zu lernen. Die Jugendlichen haben viele Gemeinsamkeiten und auch Unterschiede zu ihrem Alltag in Schleswig-Holstein kennengelernt. Solche Erfahrungen, die z. B. bei den gemeinsamen Mahlzeiten oder in den gemischten Arbeitsgruppen entstanden, sind sehr nachhaltig und tragen sicherlich dazu bei, den "europäischen Gedanken" nicht nur theoretisch zu leben. Dadurch, dass es sich nicht um einen klassischen Schüleraustausch handelte, sondern die Teilnehmer\*innen gemeinsam konkrete Arbeitsaufträge zu innovativen Themen bekamen, stand für die Jugendlichen die Lösung eines konkreten technischen Problems im Vordergrund. Der interkulturelle Austausch sowie die sprachliche Verständigung über die Aufgabenstellung waren notwendige Werkzeuge, um dieses Problem zu lösen.

Ein solches Camp kann die Möglichkeit bieten, in fremder Umgebung mit fremden Menschen in Kontakt zu treten und auf diese Weise neue Erfahrungen bieten. Gleichzeitig bietet es auch etwas für das Klassengefüge, ähnlich wie bei einer Klassenfahrt.

Ein solches Camp bietet den Jugendlichen Gelegenheit, andere Leute kennen zu lernen. Zum anderen kann die große Herausforderung, das eigene Wissen in einer Fremdsprache weiter zu geben das Selbstbewusstsein der jungen Leute stärken.

## Was ist ihrer Meinung nach am Konzept eines solchen Camps wichtig?

Wichtig am Konzept eines solchen Camps ist eine gute Information der beteiligten Klassenlehrer über das Programm, damit diese wiederum ihre Klassen gut informieren können, so dass jederzeit klar ist, was demnächst anliegt.

Wichtig am Konzept eines solchen Camps ist es, dass es auch Zeit genug gibt, damit die Schüler\*innen untereinander Ideen austauschen und neue Freundschaften knüpfen können.



Abb. 11: Jugendliche und Lehrkräfte aus 2 dänischen und 3 deutschen Schulen im Forschungscamp in Askov, März 2018

Fig. 11: Unge og lærere fra 2 danske og 3 tyske skoler i forskningslejren i Askov, marts 2018



## Hvad kan specielt sådan en camp efter din mening udrette/give videre?

Sådan en camp giver mulighed for at beskæftige sig med emneområder, der normalt bliver overset eller slet ikke bliver taget op i den normale undervisning.

Ud over de naturvidenskabelige og tekniske kompetencer er sådan en camp også ideel til at fremme kommunikations- og sprogkompetencer. Campen gav desuden mine elever mulighed for at lære nabolandet Danmark at kende ud fra et andet perspektiv, end hvis man bare er på ferie i landet. De unge lærte en masse ligheder og også forskelle at kende i forhold til deres hverdag i Slesvig-Holsten. Sådanne erfaringer, som fx opstod til de fælles måltider eller i de blandede arbejdsgrupper, er meget langtidsholdbare og bidrager sikkert også til ikke kun at leve i teorien efter den "europæiske tanke" i teorien. Det var løsningen af et konkret teknisk problem, der var i fokus for de unge, fordi der ikke var tale om en klassisk skoleudveksling, men derimod at deltagerne sammen fik konkrete arbejdsopgaver om innovative emner sammen. Den interkulturelle udveksling og den sproglige kommunikation om opgaveformuleringen var nødvendige værktøjer for at kunne løse dette problem.

Sådan en camp kan give mulighed for at få kontakt til udenlandske personer i et fremmed miljø og på denne måde give nye erfaringer. Samtidig giver det også noget til klassestrukturen, ligesom på en skoleudflugt.

Sådan en camp giver de unge mulighed for at lære andre mennesker at kende. For det andet kan den store udfordring, det er at give ens egen viden videre på et fremmedsprog, styrke de unge menneskers selvtillid.

## Hvad er efter din mening vigtigt ved selve konceptet for sådan en camp?

Det vigtige ved konceptet for sådan en camp er, at de deltagende klasselærere får ordentlig information om programmet, så de til gengæld kan orientere deres klasser på en sådan måde, at det hele tiden står klart, hvad der skal ske næst.

Det vigtige ved konceptet for sådan en camp er, at der også er nok tid, så eleverne kan udveksle ideer med hinanden og knytte nye venskaber.

## Welche Hinweise/Wünsche würden Sie den Organisatoren mitgeben, wenn diese einen weiteren Camp-Durchgang planen? (organisatorisch, inhaltlich, etc.)

Bei einer weiteren Durchführung wäre es wünschenswert, einen noch direkteren Bezug der Workshops zu den anschließenden Betriebsbesichtigungen herzustellen. Vielleicht wäre es auch gut schon in der Workshop-Phase Ansprechpartner aus der betrieblichen Wirklichkeit miteinzubeziehen, um den Schüler\*innen zu zeigen, dass ihr Tun tatsächlich Relevanz für spätere Berufsperspektiven hat.

Insgesamt war das Programm ein wenig zu straff geplant. Die Schüler\*innen hätten sich mehr Freizeit gewünscht. Die einzelnen Workshops müssten noch einmal evaluiert werden und es sollte dabei darauf geachtet werden, dass eine hohe Schüleraktivität gewährleistet ist. Zum Teil kam es zum "Leerlauf", da nur einer von vier Teilnehmern einer Gruppe wirklich aktiv etwas machen konnte.

Die Idee mit dem Camp ist richtig gut, auch die Idee mit dem Austausch zwischen Dänemark und Deutschland. Man muss aber auch auf die pädagogische Seite bei der Schülerbetreuung achten, damit ihnen nicht alles fremd erscheint und sie sich ggf. unsicher fühlen. Man sollte darauf achten, dass auch Aktivitäten eingeplant werden, die nichts mit den eigentlichen Inhalten zu tun haben, sondern bei denen sich die Schüler in Kleingruppen unterhalten und austauschen.

## Wie haben Sie die Zusammenarbeit zwischen deutschen und dänischen Schülern wahrgenommen? Zeigten sich kulturelle Unterschiede?

Die Zusammenarbeit war zu Beginn von Zurückhaltung geprägt. In der ersten Zeit sprachen die Jugendlichen in den Workshop-Gruppen vor allem mit gleichsprachigen Jugendlichen. Es wäre sicherlich sinnvoll zu Beginn des Camps den Fokus noch stärker auf gruppenbildende Maßnahmen oder Spiele, die die Sprach- und Kulturbarrieren knacken, zu legen. Dieses sollte sowohl in der großen Gruppe als dann auch kurz in der Kleingruppe passieren.

Wenn die Gruppen wirklich am Problem arbeiteten, verschwanden diese Barrieren später aber fast vollständig.

Mein Eindruck war, dass zunächst eine große Distanz da war, aber nicht nur zwischen deutschen und dänischen Schüler\*innen, sondern generell zwischen fremden Schülern. In den Arbeitsgruppen war diese Distanz aber nicht störend, bzw. wurde sie auch etwas abgebaut, allerdings



Abb. 12: Schüler\*innen beim Programmieren eines Lego-Mindstorm-Roboters

Fig. 12: Elever, der programmerer en Lego Mindstorm-robot



#### Hvilke henvisninger/ønsker ville du give arrangørerne med, hvis de nu planlægger endnu en camp? (organisatorisk, indholdsmæssigt, osv.)

Hvis det skal gennemføres igen, ville jeg ønske mig at få skabt en endnu mere direkte forbindelse mellem workshopperne og de efterfølgende virksomhedsbesøg. Måske kunne det også være godt at inddrage kontaktpersoner fra den virksomhedsmæssige virkelighed allerede i workshop-fasen for at vise eleverne, at deres indsats faktisk er relevant i forhold til senere jobudsigter.

Samlet set var programmet en smule for stramt planlagt. Eleverne kunne godt have tænkt sig at have mere fritid. De enkelte workshops skal nok evalueres igen, og der bør i den sammenhæng lægges vægt på at sikre en høj aktivitet fra elevernes side. Der gik nogle gange lidt "tomgang" i den, fordi der kun var én ud af de fire deltagere i gruppen, der virkelig kunne lave noget aktivt.

Ideen med campen er rigtig god, også ideen med udveksling mellem Danmark og Tyskland. Man er dog også nødt til at være opmærksom på den pædagogiske del af at have ansvaret for eleverne, så det hele ikke bare føles fremmed, og de i givet fald føler sig usikre. Man bør tage hensyn til, at der også lægges aktiviteter ind, der ikke har noget at gøre med det egentlige indhold, men hvor eleverne kan snakke sammen i mindre grupper og udveksle ideer.

#### Hvordan oplevede du samarbejdet mellem de danske og de tyske elever? Viste der sig at være kulturforskelle?

Samarbejdet var præget af tilbageholdenhed i starten. I det første stykke tid snakkede de unge frem for alt sammen med andre unge i deres workshopgruppe, der talte det samme sprog som dem selv. Det ville sikkert give mening at lægge et større fokus på tiltag eller spil i starten af campen til at danne grupper, så de sproglige og kulturelle barrierer kan nedbrydes. Det burde man både gøre i den store gruppe og kort i den lille gruppe.

Når grupperne virkelig arbejdede med problemet, så forsvandt disse barrierer næsten fuldstændig senere.

Mit indtryk var, at der i starten var en stor distance til stede, dog ikke kun mellem de danske og de tyske elever, men derimod generelt mellem elever, som ikke kendte hinanden. I arbejdsgrupperne var denne distance dog ikke forstyrrende, eller den blev rettere sagt også nedbrudt lidt, men dog ikke så meget, at det resulterede i fælles aktiviteter i pauserne.

wohl nicht so weit, dass daraus gemeinsame Aktivitäten in der freien Zeit resultierten.

Manche Gruppen haben gut zusammen gearbeitet, andere nicht ganz so gut, weil bei denen die Sprache wohl eine zu große Hürde darstellte. Auch der relativ große Altersunterschied zwischen den Jugendlichen (8.–10.Klasse) hat manche Gruppen an einem noch produktiveren Austausch gehindert.

## Wie haben Sie den Austausch mit den Kollegen aus dem anderen Land wahrgenommen?

Der Austausch mit den Lehrerkollegen aus Dänemark war für mich das wohl gewinnbringendste Erlebnis am Camp. Sehr gut gefallen hat mir die gemeinsame Betreuung eines Workshops zusammen mit einem dänischen Kollegen. Hier haben wir viel voneinander gelernt und profitiert. Wichtig waren auch die gemeinsamen Abende mit den dänischen Kollegen bei einem Glas Bier. Hier fanden hoch interessante Gespräche über das dänische und deutsche Bildungssystem sowie über Unterschiede und Gemeinsamkeiten des Lehreralltags in beiden Ländern statt. Mit einem dänischen Kollegen bin ich seitdem im regelmäßigen Austausch.



Abb. 13: Teilnehmer\*innen des Forschungscamps 2018 in Askov beim abschließenden Besuch der Phänomenta in Flensburg

Fig. 13:.Deltagere af forskningslejren 2018 i Askov under det sidste besøg på Phänomenta i Flensborg



Nogle grupper arbejdede godt sammen, andre ikke så godt, fordi sproget nok udgjorde en for stor hurdle for dem. Den relativt store aldersforskel på de unge (8.–10. Klasse) forhindrede nok også nogle grupper i at få en endnu mere produktiv udveksling af ideer.

### Hvordan oplevede du udvekslingen med kollegaerne fra det andet land?

Udvekslingen med lærerkollegaerne fra Danmark var nok den mest udbytterige oplevelse for mig ved campen. Jeg kunne rigtig godt lide, at vi skulle stå for en workshop sammen med en dansk kollega. Der lærte vi meget af hinanden og havde glæde af hinanden. Fællesaftenerne med de danske kollegaer over et glas øl var også vigtige. Der var en masse interessante snakke om det danske og det tyske uddannelsessystem og om forskelle og ligheder mellem hverdagen som lærer i begge lande. Siden da har jeg regelmæssigt udvekslet ideer med en dansk kollega.

#### Exemplarische Schüleraussagen

#### Was war gut?

- Freizeit, andere Leute kennenlernen, mit anderen zusammenarbeiten, Essen
- Gutes Internet, mit anderen Schülern aus anderen Schulen/aus Dänemark reden und arbeiten; Schüler aus Dänemark haben getanzt; nettes Personal; man wurde nicht gedrängt (keine strengen Regeln)
- Leckeres Essen, aber kalt; Englisch sprechen; gutes WLAN, nette Leute; vieles gelernt; Handys benutzen dürfen; erst um 23 Uhr schlafen müssen
- Mir hat gefallen, dass wir einen Zettel hatten, wo alles drauf stand, was wir so machen. Dass wir WLAN hatten, dass die Dänen so gut getanzt haben. Dass ich mich mit den Dänen verstanden hab. Dass wir mit den Ozobots, Pumpkins und Drohnen gearbeitet haben.

#### Was habe ich gelernt?

- Dass Ozobots dumm sind und man ihnen jeden kleinen Schritt erzählen muss, ein bisschen Dänisch
- Roboter sind komplexe Geräte, man muss ihnen genaue Befehle geben, es kann manchmal sehr lange dauern und man muss viel ausprobieren
- Wieviel Arbeit es benötigt um einen Roboter zu programmieren, dass es auch immer anders ist bei jedem Roboter, sie zu programmieren
- I learned a little bit more English

#### Verbesserungsvorschläge

- · Längere und mehr Pausen, man kann sich die Workshops aussuchen
- · Mehr auf Englisch, besseres Essen
- Mehr Freizeit, nicht so lange Vorträge, ein paar Vorträge auf Deutsch

#### **Fazit**

Ich werde mich im Anschluss noch mit dem Thema Roboter beschäftigen



Abb. 14: Deutsche und dänische Schüler\*innen arbeiten gemeinsamen an Aufgabenstellungen bei der Programmierung von Robotern, Askov 2018

Fig. 14: Tyske og danske elever arbejder sammen om programmering af opgaver til robotter, Askov 2018

#### Eksempler på elevudtalelser



#### Hvad var godt?

- Fritiden, at lære andre at kende, at arbejde sammen med andre, maden
- Hurtigt internet, at snakke og arbejde sammen med andre elever fra andre skoler/fra Danmark; eleverne fra Danmark dansede; sødt personale; man blev ikke påtvunget noget (ingen strenge regler)
- Lækker mad, men koldt; at tale engelsk; hurtigt trådløst internet, søde mennesker, har lært meget; man måtte bruge sin mobil; først at skulle sove kl. 23.
- Jeg kunne godt lide, at vi havde en seddel, hvor der stod alt det på, som vi skulle lave. At vi havde trådløst internet, at danskerne dansede så godt. At jeg kom godt ud af det med danskerne. At vi arbejdede med både ozobots, pumpkins og droner

#### Hvad har jeg lært?

- At ozobots er dumme, og man er nødt til at fortælle dem, hvert lille skridt de skal tage, en lille smule dansk
- Robotter er komplekse apparater, man skal give dem helt præcise instrukser, nogle gange kan det tage meget lang tid, og man er nødt til at prøve en masse af
- Hvor meget arbejde det kræver at programmere en robot, at det er forskelligt fra robot til robot, hvordan man programmerer dem
- I learned a little bit more English

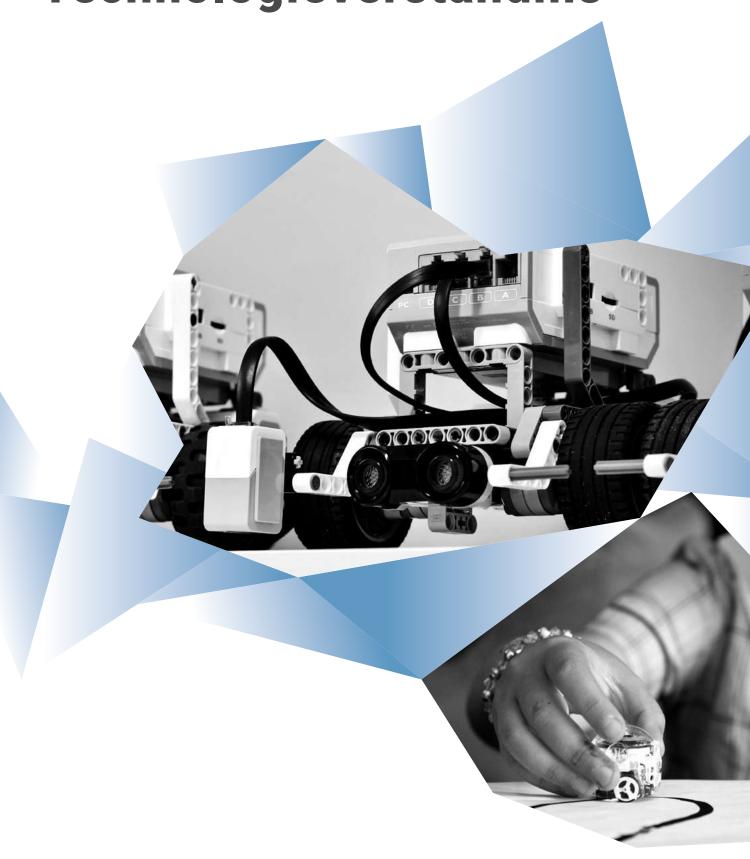
#### **Forbedringsforslag**

- · Længere og flere pauser, at man selv kan vælge workshopperne
- · Mere på engelsk, bedre mad
- Mere fritid, ikke så lange oplæg, et par oplæg på tysk

#### **Konklusion:**

Jeg vil også beskæftige mig med robotter efterfølgende

## VI. Schwerpunkt Technologieverständnis



## VI. Fokus på teknologisk forståelse



#### Technologische Grundbildung sichert den Zugang zur modernen digitalen Arbeitswelt – Robotik und Kryptographie im Schulunterricht

Prof. Claus Michelsen, Linda Ahrenkiel, Kaj Nedergård Jepsen (UC Syd), Annette Jäpelt (UCL) / Laboratorium for Sammenhængende Uddannelse og Læring (LSUL), SDU Odense; Prof. Jørgen Christian Larsen, Prof. Jacob Nielsen, Bjarke Kristian Maigaard Kjær Pedersen / Teknologiskolen, SDU Odense; Benedikt Karrasch, Stefanie Herzog / IPN Kiel

In der modernen Welt sind die Entwicklung und Nutzung von Technologien Teil des fundamentalen menschlichen Handelns, um Anforderungen im Privat- und Arbeitsleben effizienter bewältigen zu können. Da insbesondere die digitalen Technologien zunehmend Einfluss auf unser Leben haben, steigt der Bedarf an Kompetenzen, diese zu verstehen und mit ihnen zu arbeiten. Im Bildungskontext ist hierbei die Rede von technologischer Grundbildung (technological literacy) oder häufiger Technologieverständnis. Technologieverständnis bezieht sich konkret auf den Einfluss von Technologie auf soziale und kulturelle Umstände in der Lebens- und Arbeitswelt. Für eine adäquate Vorbereitung ist es wichtig, dass Jugendliche durch das Bildungssystem Gelegenheit bekommen, die Kompetenzen zu erlernen, die von der Gesellschaft und dem Arbeitsmarkt der Zukunft von ihnen erwartet werden. Für das Verständnis digitaler Technologie spielen hier verschiedene Aspekte eine Rolle, von denen im Folgenden exemplarisch die Programmierung von Robotern und die sichere Verschlüsselung von Informationen thematisiert werden.

Wir hoffen, dass das vorgestellte Material und die Hintergrundinformationen als Inspirationsquelle für Lehrkräfte dienen, um Aspekte zur Robotik, Programmierung und Kryptographie kennenzulernen, um sie gemeinsam mit ihren Schüler\*innen zu behandeln und um Kompetenzzuwächse bei diesen zu erleben. Die Projekte in diesem Bereich sollten möglichst fächerübergreifend angelegt werden, um unterschiedliche Wissensbereiche, Interessen und Stärken der Lernenden zu kombinieren.

In diesem Kapitel finden Lehrkräfte Anregungen für unterschiedliche Unterrichtsideen, in die neben der Robotertechnik (vgl. auch Band 1, Kapitel V) sowohl die Programmierung als auch die Verschlüsselungstechniken einfließen können. Alle hier vorgestellten Ideen sind mit Schulklassen oder in den PANaMa-Forschungs Camps mit Schüler\*innen erprobt worden. Gemäß der Ziele des PANaMa-Projekts werden die Vorschläge mit Informationen zum Arbeitsmarkt der Zukunft verknüpft sowie um eine Präsentation der Projektergebnisse z. B. über schülerkuratierte Ausstellungen (vgl. Band 1 und 2) ergänzt.



Abb. 15: Mit dem Verständnis technologischer Prozesse erlangen Jugendliche grundlegende Kompetenzen

Fig. 15: Med en forståelse af teknologiske processer erhverver de unge grundlæggende færdigheder

#### Grundlæggende teknologisk uddannelse sikrer adgang til den moderne digitale arbejdsverden – robotik og kryptografi i skoletimer

Prof. Claus Michelsen, Linda Ahrenkiel, Kaj Nedergård Jepsen (UC Syd), Annette Jäpelt (UCL) / Laboratorium for Sammenhængende Uddannelse og Læring (LSUL), SDU Odense; Prof. Jørgen Christian Larsen, Prof. Jacob Nielsen, Bjarke Kristian Maigaard Kjær Pedersen / Teknologiskolen, SDU Odense; Benedikt Karrasch, Stefanie Herzog / IPN Kiel



I den moderne verden er udvikling og anvendelse af teknologier en del af grundlæggende menneskelig handling for at kunne leve mere effektivt op til krav og forventninger i privat- og arbejdslivet. I og med, at især digitale teknologier i stigende grad påvirker vores liv, øges behovet for kompetencer med hensyn til at forstå og arbejde med dem. I uddannelsesmæssig henséende er der tale om basal teknologisk dannelse (technological literacy) eller mere almindeligt: Teknologisk forståelse. At forstå teknologi refererer specifikt til teknologiens indflydelse på sociale og kulturelle forhold i vores liv og arbejdsunivers. Med henblik på en fyldestgørende forberedelse er det her vigtigt, at uddannelsessystemet giver unge mennesker mulighed for at tilegne sig de kompetencer, som samfundet og fremtidens arbejdsmarked forventer af dem. I den forbindelse spiller forskellige forhold en rolle vedrørende forståelsen af digital teknologi, hvor programmering af robotter og sikker kryptering af information skal være de eksempler, som diskuteres i det følgende.

Vi håber, at det præsenterede materiale og baggrundsinformationen fungerer som en kilde til inspiration for lærere til at stifte bekendtskab med aspekter omkring robotteknologi, programmering og kryptografi med henblik på at behandle dem sammen med deres elever og opleve forbedringer af deres kompetencer. Projekterne indenfor dette område bør være så tværfaglige som muligt med henblik på at kombinere forskellige læringsområder, viden, interesser og styrkepositioner.

I nærværende kapitel finder lærere forslag til forskellige undervisningsidéer, hvor man ud over robotteknologi (se også bind 1, kapitel V) både kan lade programmering og krypteringsteknik indgå. Alle idéer, der præsenteres her, er blevet afprøvet med skoleklasser eller med elever i PANaMa-forskningscamps. I overénsstemmelse med PANaMa-projektets målsætninger er forslagene knyttet sammen med information om fremtidens arbejdsmarked og suppleret med en præsentation af projektresultater, f. eks. elevgennemførte udstillinger (se bind 1 og 2).

Die beschriebenen Aktivitäten sind für dänische und deutsche Schüler\*innen der Klassenstufen 7–10 geeignet.

#### 1 HINTERGRUNDINFORMATIONEN

## 1.1 Konzept sowie pädagogische und didaktische Betrachtungen

Wie auch die anderen Unterrichtsvorschläge im PANaMa-Projekt folgen die Unterrichtsideen zum Technologieverständnis einem offenen, handlungsorientierten oder forschenden Unterrichtsansatz.

Im Mittelpunkt des forschenden Lernansatzes stehen die eigenen Hypothesen und Untersuchungen der Schüler\*innen. Insbesondere geht es darum, von den Schüler\*innen Ideen entwickeln zu lassen und mit diesen weitere Untersuchungen anstellen zu lassen. Dabei werden das Vorwissen und die Erfahrung der Schüler\*innen als Ausgangspunkt betrachtet. Für die Einstellung der Lehrkraft ist es wichtig, gemeinsam mit den Schüler\*innen neugierig zu sein. Es geht nicht darum, den Lernenden Fragen in den Mund zu legen, auf die sie selbst nicht gekommen wären. Stattdessen ist das Ziel, die Begriffe, Erklärungen und das Verständnis der Schüler\*innen durch ihre neuen Erfahrungen, Handlungen und Gedanken weiterzuentwickeln.

Im vorliegenden Material werden exemplarisch offene Aufgaben vorgestellt, mit denen sich die Schüler\*innen selbstständig beschäftigen können. Grundsätzlich sollten offene Aufgaben oder Fragen immer so formuliert sein, dass sie an Vorwissen der Lernenden anknüpfen und gleichzeitig ein konkretes Ergebnis einfordern, das ein paar Schritte jenseits dessen liegt, was Schüler\*innen ohne Anstrengung beantworten können. Sie sollten gleichzeitig so komplex sein, dass sie in kleinere Teile zerlegt werden müssen, um erfolgreich bearbeitet werden zu können. Bei allen Fragen oder Aufgaben in diesem Kapitel steht immer das Ziel im Vordergrund, über die Entwicklung einer funktionierenden Programmierung von Robotern oder Computer zur Lösung eines Problems zu kommen. Wenn möglich sollte eine erste Problemlösung dann noch optimiert werden.

Aufgrund der Art der Aufgabenstellung ist es wichtig, dass die Schüler\*innen ausreichend Zeit haben, um ihre Ansätze und Ideen zu diskutieren. Offene Aufgabenstellungen fördern Kreativität und Relevanzempfinden der Lernenden – die damit verbundenen Arbeitsweisen müssen allerdings erfahren bzw. erlernt worden sein. Offene Ansätze gehen immer auch mit der Gefahr der Überforderung einher, weshalb es gerade hierbei immer sinnvoll ist, Hilfestellungen bereit zu halten. Diese sollen



Abb. 16: Moderne Medien sind ein unerlässliches Werkzeug geworden

Fig. 16: Moderne medier er blevet et vigtigt værktøj

De beskrevne aktiviteter er velegnede til danske og tyske skoleelever på 7. til 10. klassetrin.



# 1.1 Koncept samt pædagogiske og didaktiske betragtninger

I lighed med de andre undervisningsrelaterede forslag i forbindelse med PANaMa-projektet har undervisningsidéerne vedr. forståelse af teknologi en åben, handlingsorienteret eller forskningsrelateret undervisningstilgang.

Det er elevernes egne hypoteser og undersøgelser, som er i centrum for den forskningsbaserede læringsmetode. Især handler det om at få eleverne til at udvikle egne idéer og ved hjælp af disse gennemføre yderligere undersøgelser. I den forbindelse checkes der forinden op på elevernes erfaringer og eksisterende viden. For tilgangen på lærersiden er det vigtigt at være nysgerrig sammen med eleverne. Det handler ikke om at stille spørgsmål til eleverne, som de ikke selv ville være kommet i tanker om. Det er snarere målsætningen at videreudvikle elevernes termer/begreber, forklaringer og forståelse gennem de nye oplevelser, handlinger og tanker, de når frem til.

I det foreliggende materiale præsenteres eksempler på åbne opgaver, som eleverne kan håndtere og beskæftige sig med på egen hånd. I princippet bør åbne opgaver eller spørgsmål altid formuleres på en sådan måde, at de bygger videre på elevernes eksisterende viden og samtidig stiller krav om et konkret resultat, som ligger en smule ud over, hvad eleverne ellers uden videre ville kunne svare på. Samtidig bør opgaverne være så komplekse, at man er nødt til at skille dem ad i mindre dele for at kunne løse dem. For alle spørgsmåls eller opgavers vedkommende i dette kapitel er den overordnede målsætning altid at løse et problem ved at udvikle en fungerende programmering til robotter eller computere. Og hvis det er muligt, bør en indledende problemløsning sidenhen optimeres længere henne i forløbet.

Grundet opgavens art er det vigtigt, at eleverne har tid nok til at diskutere deres tilgange og idéer. Åbne opgaveformuleringer fremmer de lærende personers kreativitet og fornemmelse for relevans – dog således, at de resulterende arbejdsmetoder skal være oplevelses- eller læringsbaserede. Åbne tilgange går altid hånd i hånd med en vis risiko for at ville eller skulle for meget (overchallenged), hvorfor det altid giver mening at kunne hjælpe, hvis det bliver nødvendigt. Sådan en hjælp bør



dabei keinesfalls die inhaltliche Lösung oder Teile davon umfassen, sondern strategischer Natur sein. Die Hilfen sollten eher Impulse liefern, sich die Aufgabe zunächst in Teile zu untergliedern und dann diese Teile zu bearbeiten. Dabei empfiehlt es sich nicht nur aufgrund einer möglichen begrenzten Ausstattung der Schule auf Gruppenarbeit zu setzen; diese kooperative Sozialform fördert im Idealfall auch das Aufgreifen und kritische Bewerten von Ideen anderer und das Aushandeln von möglichen Lösungsideen. Sollten Lernende es wünschen, kann ihnen phasenweise aber auch Einzelarbeit ermöglicht werden, um sich selbst auszuprobieren und ggf. mit Frustrationsphasen umgehen zu lernen.

Für die Dokumentation empfiehlt es sich, die Forschungsfrage oder offene Aufgabe sowie die dazu bereits erarbeiteten Lösungschritte visuell festzuhalten (z. B. über (digitale) Mindmaps oder Wissenslandkarten), so dass sich Lernende jederzeit vergewissern können, warum sie etwas machen und wozu das Ergebnis letztendlich beitragen soll. So kann etwa zur Darstellung der Programmierungsideen ein von den Schüler\*innen erstelltes Flussdiagramm dienen, bevor die Lernenden zur eigentlichen Programmiersprache übergehen. Dies erleichtert auch die Diskussion in der Gruppe, da man schneller einen Eindruck von den Ideen der anderen Schüler\*innen erhält, eine gemeinsame Sprache findet und so lösungsorientierter vorgehen kann.

Ideal ist bei allen programmierten Problemlösungen, dass Schüler\*innen sofort Feedback zu ihren Lösungsideen erhalten: Einmal eingegeben, kann man sofort sehen, ob z. B. der Roboter das tut, was er machen soll, oder ob es irgendwo hakt. Vorteil dieser Art des Feedbacks ist auch, dass es nicht von einer bewertenden Instanz (wie z. B. einer Lehrkraft) kommt, sondern von einer neutralen. Auch wenn letzteres zunächst ein wenig frustrierend sein mag, so kann jederzeit der Frage nachgegangen werden, "Was passiert, wenn wir hier oder da etwas ändern?" So müssen sich Lernende nicht allein mit hypothetischen Fragen beschäftigen, sondern können konkret ausprobieren, welchen Effekt ein bestimmter Baustein im Programm hat. Das Resultat lässt sich dann am Roboter direkt ablesen.

#### 1.2 Programmiersprachen

Programmiersprachen sind Sprachen, die in der Programmierung angewendet werden und mit denen Computerprogramme bzw. Software erstellt werden. Wie bei allen anderen Sprachen gibt es auch hier Regeln für die sprachlichen Ausdrücke. Eine solche Definition oder Spezifikation kann in Syntax (Form/Grammatik) und Semantik (Bedeutung) unterteilt werden.

Es gibt derzeit keine branchenübergreifende oder allgemeingültige Programmiersprache für Roboter. Viele Programmiersprachen in der Robotik sind sich jedoch ähnlich, da sie über eine ähnliche Syntax verfügen wie die Programmiersprache C. Die Sprache C war eine der ersten Programmiersprachen, mit denen man versuchte, das Programmieren für den Menschen lesbarer zu gestalten und somit das Schreiben und das Verständnis zu erleichtern. Sobald man eine Programmiersprache erlernt hat, kann man sich relativ schnell in eine weitere Programmiersprache gleichen Typs hineindenken. Viele Informatiker\*innen lernen heute zunächst das Programmieren in den Sprachen Python, C, C++ oder Java. Die zahlreichen unterschiedlichen Programmiersprachen werden häufig in Flow-Programmierung, Funktions-Programmierung oder andere Arten der Programmiersprache unterteilt. Im Rahmen der in diesem Kapitel vorliegenden Ansätze ist

imidlertid under ingen omstændigheder være i form af selve den indholdsmæssige løsning eller dele af den, men snarere være af strategisk karaktér. Hjælpen skal hellere give impulser til at starte med at dele opgaven op i flere dele og sidenhen give sig til at arbejde med disse dele. I den forbindelse er det – og ikke kun, fordi skolen muligvis kun har begrænsede muligheder til rådighed – en god idé at satse på gruppearbejde, idet denne sociale samarbejdsform, hvis det kører godt, fremmer muligheden for at tage andres tanker og idéer op, evaluere dem kritisk samt 'forhandle' sig frem til opgaveløsninger. Hvis eleverne ønsker det, kan de imidlertid i nogle perioder også få lov til at være alene om noget arbejde for at afprøve sig selv og, om nødvendigt, også lære at håndtere perioder med frustration.

Vedr. dokumentation tilrådes det, at man visuelt registrerer forskningsspørgsmålet eller den åbne opgave samt de løsningstrin, som allerede er udarbejdet (f. eks. via (digitale) tankekort eller videnlandkort), så eleverne hele tiden kan blive mindet om, hvorfor de gør noget, og hvad resultatet i sidste ende skal bidrage til. F. eks. kan et flowdiagram, oprettet af eleverne, bruges til at illustrere programmeringsidéerne, inden eleverne skifter over til selve programmeringssproget. Dette letter også diskussionen i gruppen, idet man hurtigere kan få et indtryk af idéerne fra de andre elever, finde et fælles sprog og dermed være mere løsningsorienteret.

I forbindelse med programmerede problemløsninger er det altid bedst, hvis eleverne med det samme får en tilbagemelding på deres idéer: Man indtaster noget – og sér straks, om f. eks. robotten gør det, den skal, eller om der er noget galt et sted. Fordelen ved denne type feedback er, at den ikke kommer fra en evaluerende instans (f. eks. en lærer), men fra en neutral. Selv hvis sidstnævnte måske er lidt frustrerende i starten, kan man altid komme videre ved at spørge "Hvad sker der, hvis vi ændrer noget her eller der?" På den måde skal eleverne ikke kun beskæftige sig med hypotetiske spørgsmål, men kan konkret afprøve, hvilken effekt et bestemt modul i programmet har. Og resultatet kan konstateres direkte på robotten.

#### 1.2 Programmeringssprog

Programmeringssprog er sprog, der bruges til programmering og dermed fremstilling af computerprogrammer eller software. Og som med alle andre sprog er der også her regler for den sproglige udtryksmåde. Og sådan en definition eller specifikation kan igen opdeles i syntaks (form/grammatik) og semantik (betydning).

Der findes p. t. ikke noget tværsektorielt eller alméngyldigt programmeringssprog for robotter – dog således, at mange programmeringssprog indenfor robotteknologien alligevel ligner hinanden ganske meget, fordi de har en lignende syntaks som programmeringssproget C. Sproget C var ét af de første programmeringssprog, med hvilket man forsøgte at gøre programmeringen mere læselig for mennesker og dermed gøre det nemmere at skrive skrive og forstå. Såsnart man har lært et programmeringssprog, kan man rimelig hurtigt også sætte sig ind i et andet programmeringssprog af samme type. Mange dataloger starter nu om dage med at lære at programmere på sprogene Python, C, C++ eller Java. De mange forskellige programmeringssprog er ofte opdelt i flowprogrammering, funktionsprogrammering eller andre typer programmeringssprog. I forbindelse med de tilgange, der er præsenteret i dette kapitel, er det muligt at samle nogle første indtryk af de forskellige typer programmeringssprog. F. eks. kan

es möglich, erste Eindrücke der verschiedenen Typen der Programmiersprache zu sammeln. Beispielsweise können die Ozobot-Miniroboter, die in 1.3 vorgestellt werden, mittels Farbcodierung (Flow-Programmierung) und Blockprogrammierung (Funktions-Programmierung) programmiert werden. Bei der Blockprogrammierung werden bereits vorprogrammierte und visuell aufbereitete "Bausteine" aneinandergereiht, so dass eine Abfolge an Befehlen erfolgt. Die Blockprogrammierung ist eine häufig angewandte Anfängersprache des Programmierens und wird beispielsweise auch für die Sprache Scratch oder MakeCode verwendet.

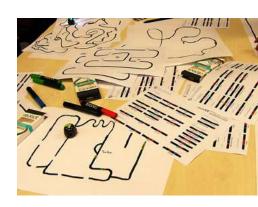
Im Zuge der Arbeit mit industriellen Robotern ist es notwendig, dass die Roboter die jeweilige Arbeitsschritte eindeutig und in korrekter Reihenfolge erledigen. Es werden dazu Codes verwendet, um dem Computer im Roboter mitzuteilen, was er tun soll. Damit der Roboter ein Problem oder eine Aufgabe lösen kann, muss zunächst ein Algorithmus entwickelt werden, also eine Liste von Arbeitsschritten und Regeln, die für die Problemlösung befolgt werden müssen. Der Algorithmus muss die einzelnen Schritte in der richtigen Reihenfolge enthalten und es muss sichergestellt sein, dass er sein Ziel erreicht.

Fast alle Programmiersprachen umfassen einige Basisbefehle. So wird beim Programmieren oft die Befehlskombination if (falls), then (dann), else (ansonsten) eingesetzt. Diese Befehlskombination wird verwendet, um in einer Entscheidungssituation anzugeben, unter welcher Bedingung welche Anweisung umgesetzt wird. Gäbe es z. B. einen Roboter, der sich anziehen könnte, müsste dieser sinngemäß folgenden Algorithmus erhalten: "Falls Strümpfe angezogen, dann Schuhe anziehen, ansonsten Strümpfe anziehen". Im Roboterkontext werden diese Befehle häufig verwendet, um aus einer Sensoreingabe eine Entscheidung zu treffen. Es kann aber auch um das Ausführen einer Bewegung (Abbiegen oder Vorwärtsbewegung) gehen: "Falls der Abstandssensorwert <50 cm beträgt, dann stoppen und um 180 Grad drehen, ansonsten vorwärts bewegen."

Lehrkräfte können weitere Basisinformationen und Anregungen u. a. auf dieser Webseite finden: http://www.taccle3.eu/english/category/understanding-algorithms/advanced/

## 1.3 Exemplarische Technologien

In diesem Abschnitt werden kurz die drei verwendeten Technologien der Roboter (Ozobot, LEGO MINDSTORMS® und Arduino) sowie die Verfahren zur Kryptographie vorgestellt.



Ozobot-minirobotterne, der præsenteres i 1.3, programmeres ved hjælp af farvekodning (flowprogrammering) og blokprogrammering (funktionsprogrammering). Når man blokprogrammerer, betyder det, at nogle forprogrammerede og visuelt forberedte "byggeklodser" sættes på række, så der opstår en sekvens af kommandoer. Blokprogrammering anvendes ofte til begyndere f. eks. i sprogene Scratch eller MakeCode.

Når man arbejder med industrirobotter, er det nødvendigt, at robotterne udfører de respektive arbejdstrin utvetydigt og i den rigtige rækkefølge. I den forbindelse anvendes koder til at fortælle computeren i robotten, hvad den skal gøre. For at robotten kan løse et problem eller en opgave, skal der først udvikles en algoritme, d. v. s. en liste over arbejdstrin og regler, som skal følges for at løse problemet. Algoritmen skal indeholde de enkelte trin i den rigtige rækkefølge, og det skal sikres, at den når sit mål.

Næsten alle programmeringssprog indeholder nogle grundlæggende kommandoer. Således anvendes under programmeringen ofte kommandokombinationen, if (hvis), then (så), else (ellers). Denne kombination af kommandoer bruges i en beslutningssituation til at indikere, hvilken instruktion der skal følges under hvilken betingelse. Fandtes der f. eks. en robot, der kunne klæde sig på, skulle den – sådan lidt firkantet udtrykt – forsynes med følgende algoritme: "Hvis strømper er på, så tag sko på, ellers tag strømper på". I en robotsammenhæng bruges disse kommandoer ofte til at tage en beslutning på basis af, hvad en føler (sensor) har registreret. Men det kan også handle om at udføre en bevægelse (dreje rundt om et hjørne eller bevæge sig ligeud fremad): "Hvis afstandssensorværdien er <50 cm, så stop og drej 180 grader, ellers gå fremad."

Som lærer kan man finde yderligere basal information og inspiration på denne hjemmeside: http://www.taccle3.eu/english/category/understanding-algorithms/advanced/.

# 1.3 Eksempler på teknologier

I dette afsnit redegøres der i korte træk for de tre anvendte teknologier for robotterne Ozobot, LEGO MINDSTORMS® og Arduino samt metoderne vedr. kryptografi.

#### Ozobot

Ozobot er en lille produktserie med små programmérbare robotter, som såvel kan programmeres v. h. a. flowprogrammering (se fig. 17) som også

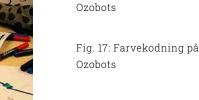
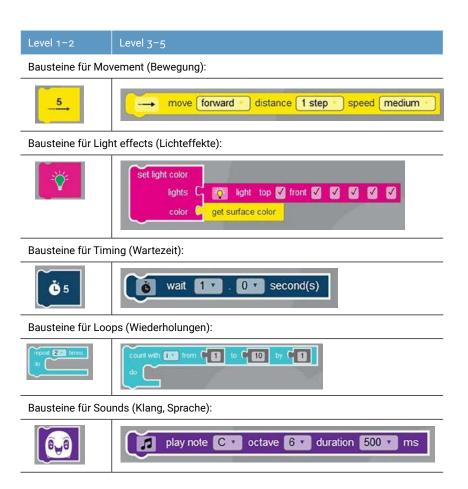


Abb. 17: Farbkodierung der

#### Ozobot

Ozobot ist eine Produktserie mit kleinen programmierbaren Robotern. Diese können sowohl über eine Flowprogrammierung (s. Abb. 17) als auch über eine webbasierte englischsprachige Blockprogrammierung (s. Abb. 18 und Tab. b) auf verschiedenen Levels (Anspruchniveaus, von pre-reader bis master) programmiert werden.

Die Ozobot-Software kann einfach aus dem Internet heruntergeladen werden: https://ozobot.com/. Es gibt eine Reihe von Internetseiten mit überwiegend englischsprachigen Anleitungen und Videos zur Arbeit mit den Robotern (siehe u. a. https://ozobot.com/stem-education). Hier finden sich u. a. Anregungen für Aufgabenstellungen und die Möglichkeiten zur Teilnahme an Webinaren. Die Optionen zur Programmierung in Ozoblockly sind für alle Level in folgende farbkodierten Blöcke unterteilt, wobei bei den fortgeschritteneren Levels auch komplexere Bausteine enthalten sind bzw. programmiert werden können (s. Tab. b):



Tab. b: Übersicht über einige Optionen zur Programmierung in Ozoblockly mit den entsprechenden Blöcken, pro Level



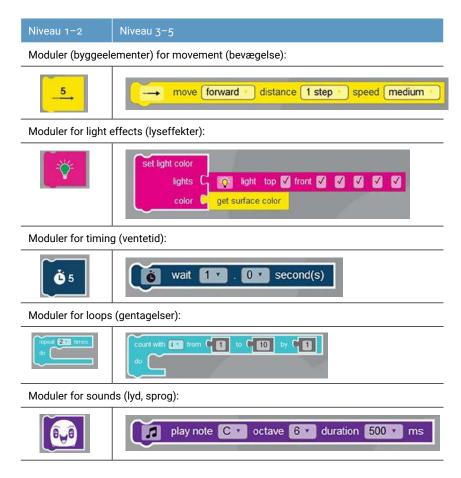
Abb. 18: Programmierfläche für Blockprogrammierung der Ozobots

Fig. 18: Programmeringsgrænseflade for blokprogrammering af Ozobots



en netbaseret engelsksproget blokprogrammering (se fig. 18 og tab. b) – og på forskellige niveauer (kravniveauer, fra pre-reader til master).

Ozobot-softwaren kan nemt downloades fra internettet: https://ozobot.com/. Der findes en række hjemmesider med vejledninger og videoer, for det meste på engelsk, om arbejdet med robotterne (se f. eks. https://ozobot.com/stem-education). Her kan man bl. a. finde inspiration til opgaveformuleringer samt mulighed for at deltage i webkurser (såk. webinarer). Optionerne med programmering i Ozoblockly er for alle niveauer opdelt i de følgende farvekodede blokke, hvor der for de lidt viderekomne niveauers vedkommende også er indeholdt eller kan programmeres mere komplekse moduler (byggeklodser) (se tab. b):



Tab. b: Skema over forskellige optioner for programmering i Ozoblockly med de tilsvarende blokke, pr. niveau

#### **LEGO MINDSTORMS®**

LEGO MINDSTORMS® ist die Version eines Roboter-Kits der Firma LEGO. Hiermit können viele unterschiedliche Arten von Robotern konstruiert und programmiert werden. Der Bausatz ist einfach zu verwenden und es gibt eine große LEGO MINDSTORMS®-Community, da es u.a. in sehr vielen Bildungseinrichtungen in Dänemark verwendet wird. Im Internet finden sich zahlreiche verschiedene Roboter aus der LEGO MIND-STORMS®-Serie, Anleitungen, Programmierungsbeispiele, Anregungen und Hilfestellungen.

LEGO MINDSTORMS® ist programmierbare Robotertechnik, mit der Schüler\*innen ihre Roboter – im Gegensatz zu den Ozobots – zunächst selbst konstruieren und dann erst programmieren können. LEGO MINDSTORMS® EV3 kann auch für die Arbeit mit elektronischer Datenerhebung verwendet werden.

Die Software kann bei LEGO Education heruntergeladen werden: https://education.lego.com/en-us/downloads. Wenn die Software installiert wird, empfiehlt sich die Installation als Lehrkräfte-Version (s. Abb. 19), da dies erweiterte Möglichkeiten inklusive einer Kurzbeschreibung für die schnelle Inbetriebnahme von LEGO MINDSTORMS® bietet. Ergänzend finden sich Internetseiten mit Anleitungen und kurzen Filmen zur Inbetriebnahme, siehe u. a.

https://www.lego.com/de-de/themes/mindstorms/learntoprogram https://www.lego.com/de-de/themes/mindstorms/downloads https://www.lego.com/cdn/cs/set/assets/bltd65492add9dd16ba/ User\_Guide\_LEGO\_MINDSTORMS\_EV3\_11\_All\_DE.pdf

Die Aufgaben in Abschnitt 2b sind auf das Automodell Chassis ausgelegt (s. Abb. 20). Die Bauanleitung dafür ist über die Software für LEGO MINDSTORMS® abrufbar.



Abb. 20: LEGO MIND-STORMS® Modell Chassis (Illustration aus der Software LEGO MINDSTORMS® education EV3)

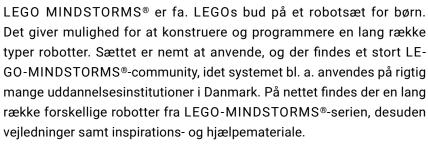
Fig. 20: LEGO-MIND-STORMS®-model Chassis (illustration fra software LEGO MINDSTORMS® education EV3)

#### LEGO MINDSTORMS®



Abb. 19: LEGO MIND-STORMS®-Software-Startseite in der Lehrer-Version. Hier ist eine schnelle Inbetriebnahme möglich und kann erfahren, wie LEGO MINDSTORMS® funktioniert

Fig. 19: LEGO-MIND-STORMS®-softwarens indgangsside i lærerversionen. Her er der mulighed for hurtigt at komme i gang, og man får at vide, hvordan LEGO MINDSTORMS® fungerer



LEGO MINDSTORMS® er programmérbar robotteknologi, v. h.a. hvilken eleverne – modsat Ozobots – i første omgang selv konstruerer og først derefter programmerer deres robotter. LEGO MINDSTORMS® EV3 kan også anvendes i forbindelse med elektronisk dataindsamling.

Softwaren kan downloades hos LEGO Education: https://education.lego.com/en-us/downloads. I forbindelse med installation af softwaren tilrådes det at vælge lærerversionen (se fig. 19), idet denne indeholder flere muligheder – inkl. en kort beskrivelse af, hvordan LEGO MINDSTORMS® kan igangsættes på en hurtig måde. Som supplement findes der yderligere hjemmesider med vejledninger og små film om, hvordan man kommer i gang, f. eks.:



https://www.lego.com/de-de/themes/mindstorms/learntoprogram https://www.lego.com/de-de/themes/mindstorms/downloads https://www.lego.com/cdn/cs/set/assets/bltd65492add9dd16ba/ User\_Guide\_LEGO\_MINDSTORMS\_EV3\_11\_All\_DE.pdf

Opgaverne i afsnit 2b er baseret på bilmodellen Chassis (se fig. 20). Byggevejledningen kan hentes via softwaren for LEGO MINDSTORMS®.

LEGO MINDSTORMS® wird mittels Blockprogrammierung programmiert, wobei die Blöcke folgendermaßen unterteilt sind (s. Tab. c):

#### Ablauf-/Flow-Blöcke (orange):

Ablauf-/Flow-Blöcke steuern den Fluss des Programms (Abwicklung). Alle Programme, die erstellt werden, beginnen mit dem Startblock.



# Aktions-Blöcke (grün):

Aktions-Blöcke steuern die Handlungen des Programms, wie Umdrehungen des Motors, Bild, Ton und Licht.



#### Sensor-Blöcke (gelb):

Sensor-Blöcke ermöglichen es dem Programm, den Input des Farbsensors, des Infrarot-Sensors, des Berührungssensors etc. zu erfassen.



#### Datenoperations-Blöcke (rot):

Datenoperations-Blöcke ermöglichen es, Variablen zu schreiben und zu lesen, Werte zu vergleichen etc. \*



#### Erweiterungs-Blöcke (blau):

Erweiterungs-Blöcke verwalten Dateien, Bluetooth-Verbindungen etc. \*



<sup>\*</sup> sind nur über die Programmierungssoftware für PC/Mac zugänglich, jedoch nicht über die EV3-Programm-App für Tablets

Tab. c: Übersicht über einige Optionen zur Programmierung in LEGO MINDSTORMS $^{\textcircled{\$}}$  mit den entsprechenden Blöcken

LEGO MINDSTORMS® programmeres v. h. a. blokprogrammering – og blokkene er opdelt på følgende måde (se tab. c):

#### Flowblokke (orange):

Flowblokke styrer programmets flow (kørsel/ afvikling). Alle programmer, der fremstilles, begynder med startblokken.



# Hændelsesblokke (grønne):

Hændelsesblokke styrer programmets handlinger, f. eks. motorens omdrejninger, billede, lyd og lys.



#### Sensorblokke (gule):

Sensorblokke gør, at programmet kan registrere input fra farvesensoren, infrarødsensoren, berøringssensoren osv.



#### Dataoperationsblokke (røde):

Dataoperationsblokke giver mulighed for at skrive og læse variabler, sammenligne værdier m. m. \*



#### Udvidelsesblokke (blå):

Udvidelsesblokke administrerer filer, bluetoothfo rbindelser m. m. \*



<sup>\*</sup> Kun tilgængelige via programmeringssoftwaren for PC/Mac, men ikke via EV3-program-appen for tablets.

 $\textit{Tab. c: Skema over nogle optioner for programmering af LEGO MINDSTORMS} ^{\textcircled{\$}} \ \textit{med de hertil indrettede blokke}$ 

#### **Arduino**

Das Arduino-System besteht aus Mikrocontroller-Boards mit analogen und digitalen Ein- und Ausgängen (s. Abb. 21) sowie zugehöriger Software, welche kostenlos vom Anbieter heruntergeladen werden kann https://www.arduino.cc/. Es handelt sich um ein Open-Source-System, d.h. die Komponenten sind quelloffen. Über ein USB-Kabel kann eine Verbindung zur Platine hergestellt werden. So kann deren Funktion geprüft werden, beispielsweise indem ein mitgeliefertes Programm, ein sogenannter "Sketch", geladen und mit einem Mausklick auf das Board übertragen wird. Die Programmierung selbst erfolgt in einer C-ähnlichen Programmiersprache (vgl. Abb. 22). Diese kann z. B. genutzt werden, um mit farbigen Dioden und ein paar Steckverbindungen eine Ampel nachzustellen. Auch die Verwendung als Licht-, Farb- oder Entfernungssensor sowie als Antrieb für Motor sind möglich.

Anleitungen für die Inbetriebnahme und die ersten Schritte können die Videos zu "Arduino Anfänger Turorials" unter https://www.youtube.com/user/MaxTechTV1 empfohlen werden. Viele Anleitungen zu verschiedensten Einsatzmöglichkeiten des Arduino-Systems gibt es neben der Anbieterseite z.B. auch auf http://mint-unt.de/arduino-1.html.

Jedes Programm besteht aus einem Setup-Teil sowie einem Loop-Teil, welche dann Bedingungen, Schleifen, Variablen etc. enthalten können:

#### Setup-Teil:

Der Setup-Teil steht an erster Stelle des Sketches und wird vom Arduino nach dem Einschalten einmalig ausgeführt. Dieser Teil enthält Anweisungen, ob PINs als Input oder Output verwendet werden sollen (s. Abb. 23). LEDs werden z.B. als OUTPUT, auszulesende Zustände von Schaltern als INPUT angesteuert.

Abb. 23: Beispiel eines Setup-Teils

## Loop-Teil:

Der Loop-Teil wird direkt nach Abschluss des Setup-Teils begonnen. Es handelt sich um eine Dauerschleife, die alle darin befindlichen Anweisungen ständig wiederholt (s. Abb. 24). Der Loop-Teil wird von oben nach unten abgearbeitet.

In den Loop-Teil werden Funktionen und Methoden geschrieben, also das, was dann ausgeführt werden soll. Dabei ist es wichtig, auf Großund Kleinschreibung zu achten. Jede Zeile, die nicht mit einer geschweiften Klammer endet, endet mit einem Semikolon. Ein paar Funktionen und



Abb. 21: Arduino Uno als eine Version aus der Arduino Mikrocontroller-Board-Serie

Fig. 21: Arduino Uno som en version fra Arduino-microcontroller-board-serien



Abb. 22: Screenshot eines Programms in der Arduino Programmierumgebung

Fig. 22.: Screenshot af et program indenfor Arduino-programmeringsmiljøet

#### Arduino

Arduino-systemet består af mikrocontrollerprints med analoge og digitale ind- og udgange (se fig. 21) samt tilhørende software, som kan downloades gratis hos udbyderen på <a href="https://www.arduino.cc/">https://www.arduino.cc/</a>. Der er tale om et open-source-system, d. v. s. komponenternes kilde er tilgængelig. Via en USB-ledning kan der etableres forbindelse til printet. På den måde kan man checke dets funktion, f. eks. ved at indlæse et medfølgende program, en såk. "Sketch", som så overføres til printet ved at klikke med musen. Selve programmeringen forgår på et C-lignende programmeringssprog (jf. fig. 22). Dette kan f. eks. bruges til at simulere en lysregulering v. h. a. kulørte dioder og nogle stikbøsninger. Anvendelse som lys-, farve- eller afstandssensor samt motordrev er ligeledes mulig.

Som vejledning til igangsætning og de indledende trin kan det anbefales at se videoer med "Arduino Beginner Turorials" på https://www.youtube.com/user/MaxTechTV1. En række vejledninger omkring diverse anvendelsesmuligheder for Arduino-systemet findes udover på selve udbydersiden f. eks. også på http://mint-unt.de/arduino-1.html.

Hvert program består af en opsætningsdel og en loopdel, som kan indeholde betingelser, sløjfer, variabler m. m.:

#### Opsætningsdelen:

Denne del kommer som det første i forb. m. sketchen og udføres een gang efter første indkobling. Den indeholder instrukser samt oplysninger, om PINs skal anvendes som in- eller output (se fig. 23). LEDer aktiveres f. eks. som OUTPUT, kontakt- eller koblingsstatus som INPUT.

Fig. 23: Eksempel på en opsætningsdel

## Loopdelen:

Denne går igang, såsnart opsætningsdelen er færdig. Der er tale om en permanentsløjfe, som hele tiden gentager alle instrukser, den indeholder (se fig. 24). Loopdelen gennemføres ovenfra og nedad.

Det, der skrives ind i loopdelen, er funktioner og metoder, d. v. s. det, som sidenhen skal udføres. I den forbindelse er det vigtigt at være opmærksom på skrivning med store og små bogstaver. Hver linje, der ikke slutter med en akkolade (en "tuborg" = {}), slutter med et semikolon (;). Nogle funktioner og metoder er vist i tabel c. Yderligere af slagsen findes i et detaljeret skema på https://www.arduino.cc/reference/en/.

Methoden sind in der Tabelle c dargestellt. Weitere sind in einer ausführlichen Übersicht auf https://www.arduino.cc/reference/en/ zu finden.

Weitere Ausführungen zum Arduino-System mit detaillierten Lehrbeispielen finden Sie in Kapitel VII. dieses Bandes.

Abb. 24: Beispiel eines Loop-Teils

Syntax	Erklärung
pinMode(13, OUTPUT);	PIN#13 wird als Output angesteuert
digitalWrite(13, HIGH);	Schaltet für PIN#13 die Stromspannung (Voltage) auf 5 Volt
delay(2000);	Wartet für 2000 Millisekunden, also 2 Sekunden
Serial.begin(9600);	Startet eine Datenübertragung mit einer seriellen Schnittstelle mit einer bestimmten Übertragungsgeschwindigkeit (Anzahl Symbole pro Sekunde)
digitalRead(11);	Zustand von PIN#11 auslesen (gibt Wert aus, 1=HIGH, 0=LOW)
int Name;	Definiert die Variable <i>Name</i> vom Datentyp Integer (ganze Zahl); in dieser können im Anschluss Werte zwischengespeichert werden
Name = digitalRead (11);	Variable erhält den ausgelesenen Wert von PIN#11 zugewiesen
Serial.println(Name, DEC);	Überträgt den Wert der Variable zur Ausgabe an den Computer, Format des Wertes ist hier eine Dezimalzahl
if (Bedingung) { CODE; }	Wenn eine bestimmte Bedingung erfüllt (wahr) ist, wird der Befehl CODE einmalig ausgeführt (→ Vergleichsoperatoren für Bedingung); ist die Bedingung nicht erfüllt, wird dieser Teil übersprungen
while (Bedingung) { CODE; }	Wie bei der if-Abfrage wird eine Bedingung geprüft, mit dem Unterschied, dass Arduino nach einmaliger Prüfung und Ausführung des CODEs die Bedin- gung erneut prüft und den CODE erneut ausführt, und zwar solange bis die Bedingung nicht mehr erfüllt ist
//	Kommentare werden hinter einen double slash geschrieben, damit man hinterher noch weiß, was eine bestimmte Anweisung machen soll. Das Programm ignoriert diese Angaben.

Tab. d: Übersicht über einige Befehle der Arduino-Umgebung

Yderligere information om Arduino-systemet med detaljerede undervisningseksempler findes i kapitel VII af dette bind.

Fig. 24: Eksempel på en loopdel.

Syntaks	Forklaring
pinMode(13, OUTPUT);	PIN#13 aktiveres om output.
digitalWrite(13, HIGH);	Sætter for PIN#13 el-spændingen (voltage) på 5 volt.
delay(2000);	Venter i 2000 millisekunder, d. v. s. 2 sekunder.
Serial.begin(9600);	lgangsætter en dataoverførsel med en seriel grænseflade og en bestemt overførselshastighed (antal symboler pr. sekund).
digitalRead(11);	Udlæsning af status på PIN#11 (udlæser værdi, 1=HIGH, 0=LOW).
int Name;	Definerer variablen <i>Name</i> af datatypen Integer (helt tal); hvorefter der her midlertidigt kan lagres ("dumpes") værdier.
Name = digitalRead (11);	Variablen får tilknyttet den udlæste værdi fra PIN#11.
Serial.println(Name, DEC);	Overfører variablens værdi til udlæsning til computeren, hvor værdiens format her er et decimaltal.
if (betingelse) { CODE; }	Hvis en bestemt betingelse er opfyldt (sand), udføres kommandoen CODE een gang (→ sammenligningsoperatorer for betingelsen); er betingelsen ikke opfyldt, skippes denne del.
while (betingelse) { CODE; }	Som ved if-forespørgslen checkes der op på en betingelse, dog således, at Arduino efter at have checket og udført CODEn een gang checker betingelsen og udfører CODEn igen – og bliver ved med det, indtil betingelsen ikke længere er opfyldt.
//	Kommentarer skrives bag en dobbelt skråstreg (double slash), så man bagefter stadig kan huske, hvad en bestemt instruks skal gøre. Programmet ignorerer disse oplysninger.

Tab. d: Skema over nogle kommandoer indenfor Arduino-miljøet

# Kryptographie

Bei der Kryptographie handelt es sich um das mathematische Teilgebiet, das sich mit der sicheren Verschlüsselung von Informationen beschäftigt. Immer, wenn irgendwo Informationen ver- oder entschlüsselt werden müssen, um diese sicher zu übertragen (z. B. Passwörter), werden Verfahren der Kryptographie genutzt. Im Schüleralltag stellen Nachrichtendienste, Browser-Adressen, Internet-Logins oder WLAN-Zugänge die größten Berührungspunkte dar. Häufig werden Verschlüsselungen genutzt, ohne dass Schüler\*innen die Verschlüsselung bewusst mitbekommen oder die Notwendigkeit zu hinterfragen. Stellt man Lernenden allerdings die Situation vor, dass z. B. ein nicht-befreundeter Mitschüler WhatsApp-Nachrichten mitlesen und darauf reagieren kann, so sind sie recht schnell für das Problem sensibilisiert.

Zum Einstieg in die Thematik eignet sich eine historische Perspektive, da so dem Vorwissen der Schüler\*innen entsprechend zunächst einfache Verschlüsselungskonzepte angewendet und nachvollzogen werden können. Ziel der Aktivitäten ist es immer, eine Nachricht so vom Sender zum Empfänger zu schicken, dass ein Unbefugter diese zwar lesen, aber nicht verstehen kann.

Die Verschlüsselung mittels Skytale ist wohl das älteste bekannte Verschlüsselungsverfahren (um 400 v. Chr). Hierbei wird ein leerer Papierstreifen (historisch gesehen war es wohl eher ein Lederstreifen) um einen Zylinder (die Skytale, also ein Stab mit bestimmtem Durchmesser) gewickelt, der Papierstreifen horizontal beschrieben (s. Abb. 25) und der Papierstreifen abgewickelt. Die Buchstaben sind auf dem Papierstreifen dann untereinander in einer Reihenfolge notiert, die keine sinnvollen Wörter ergeben. Der Empfänger der Nachricht wickelt den Streifen um einen Stab mit dem gleichen ihm und dem Absender bekannten Durchmesser. Der Nachteil dieser Verschlüsselung ist, dass man über ein Ausprobieren und Vermutungen zum Durchmesser des Zylinders recht schnell einen Weg zur Originalnachricht finden kann. Dies gelingt auch Schüler\*innen, die entsprechend ein erstes Gefühl für die Sicherheit eines Verschlüsselungsverfahrens erhalten.

Die Cäsar-Verschlüsselung wurde vom gleichnamigen römischen Feldherrn genutzt, um militärische Nachrichten sicher zu verschicken. Dabei wurde jeder einzelne Buchstabe einer Nachricht durch einen festgelegten anderen Buchstaben ersetzt. Die Substitution erfolgt dabei durch das Verschieben aller Buchstaben im Alphabet (s. Abb. 26). Die Verschlüsselung ist durch einen sowohl dem Sender als auch dem Empfänger bekannten Schlüssel (= der Anzahl der Buchstaben, um die im Alphabet verschoben wird) bestimmt (s. Abb. 27). Nachteil dieser Verschlüsselung ist, dass auch hier ein sog. brute force-Angriff (also das







Abb. 25: Beispiel einer Skytale (Quelle: Wikipedia, Commons)

Fig. 25: Eksempel på en skytale (kilde: Wikipedia, Commons)

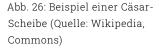


Fig. 26: Eksempel på en Cæsar-skive (kilde: Wikipedia, Commons)

# Kryptografi

Kryptografi er det matematiske underområde, der beskæftiger sig med sikker kryptering (kodning) af information. Hver gang information skal krypteres eller dekrypteres ét eller andet sted for at overføre informationen (f. eks. en adgangskode) på en sikker måde, anvendes kryptografimetoder. For skoleelevers vedkommende er nyhedstjenester, browseradresser, internetlogin eller WiFi-adgang nu om stunder de mest fremtrædende kontaktpunkter. Og ofte anvendes i den forbindelse kryptering, uden at eleverne bevidst bemærker dette eller sætter spørgsmålstegn ved nødvendigheden heraf. Men hvis man opfordrer eleverne til at forestille sig en situation, hvor f. eks. en klassekammerat, som ikke er på vennelisten, alligevel kan læse ens WhatsApp-beskeder og også reagere på dem, så bliver de hurtigt opmærksomme på problemet.

Et historisk perspektiv er velegnet til at komme i gang med emnet, idet man da i første omgang kan starte med at anvende og gengive enkle krypteringskoncepter, som passer til den viden, eleverne har i forvejen. Formålet med aktiviteterne er altid at sende en besked fra afsenderen til modtageren på en sådan måde, at en uautoriseret person også godt kan læse den, men ikke forstå den.

Kryptering v. h. a. en skytale er nok den ældste kendte metode (ca. 400 f. Kr.). Der anvendes en blank papirstrimmel (som, historisk betragtet, nok snarere har været af læder), som vikles rundt om en cylinder (en skytale, d. v. s. en stav med en bestemt diameter). Så skriver man vandret på strimlen (se fig. 25) og vikler den af staven. På strimlen står bogstaverne så under hinanden i en rækkefølge, som der ikke kommer nogen mening (ord) ud af, når man læser dem. Modtageren af beskeden vikler strimlen rundt om en stav med samme diameter, som han og afsenderen kender. Ulempen ved denne krypteringsmetode er, at man ved at gætte og formode om cylinderens diameter og prøve sig frem ret hurtigt kan finde frem til beskedens egentlige ordlyd. Og det samme vil også lykkes for elever, som på den måde får en første fornemmelse af sikkerheden af en krypteringsmetode.

Cæsar-kryptering anvendtes af den store hærfører af samme navn for at fremsende militære beskeder på en sikker måde. Metoden gik ud på, at hvert enkelt af beskedens bogstaver erstattedes af et på forhånd fastlagt andet bogstav. Substitutionen (udskiftningen) foregår ved at skubbe/forskyde alle bogstaverne i alfabetetet (se fig. 26). Krypteringen består i, at både afsenderen og modtageren kender den anvendte nøgle (= antallet af bogstaver, som der forskydes med i alfabetet) (se fig. 27). Ulempen ved denne metode er, at det også her gennem et såk. brute-force-angreb (d. v. s. ved simpelthen at blive ved med at prøve sig frem) forholdsvis hurtigt vil lykkes at bryde krypteringen. Cæsar-skiven kan i alt



Knacken der Verschlüsselung durch reines Ausprobieren) relativ schnell zum Erfolg führt. Es sind nur 26 verschiedene Einstellungen der Cäsar-Scheibe möglich. Darüber hinaus führt auch eine Häufigkeitsanalyse der eingesetzten Buchstaben schnell zu einer Entschlüsselung des Codes, wenn man die Häufigkeit der Buchstabenverteilung in der jeweiligen Sprache kennt.

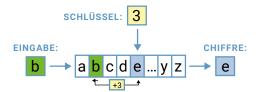


Abb. 27: Beispiel einer Cäsarverschlüsselung

Die Verschlüsselung durch Vigenère-Chiffre (benannt nach Blaise de Vigenère, 1523–1596) stellt eine Verbesserung der Cäsar-Verschlüsselung dar. Dabei wird jeder Buchstabe ähnlich der Cäsar-Verschlüsselung im Alphabet verschoben, allerdings nicht jeder Buchstabe um den gleichen Wert. Der Schlüssel besteht hier nicht nur aus einer Zahl, sondern aus mehreren Zahlen, die man als Schlüsselworte codiert. Möchte man eine Nachricht mit dem Schlüsselwort "hallo" verschlüsseln, geht man wie folgt vor:

- Das Schlüsselwort wird so oft unter die zu verschlüsselnde Nachricht geschrieben, dass jeder Buchstabe der Nachricht einen Buchstaben des Schlüsselworts zugewiesen bekommt.
- Das Schlüsselwort selbst besteht aus fünf Buchstaben.
- Der erste Buchstabe des Schlüsselworts "hallo" ist ein H. Dieses steht an der 7. Stelle des Alphabets, so dass der Buchstabe, der mit dem H verschlüsselt werden soll, um sechs Buchstaben im Alphabet (da der Buchstabe A eine Verschiebung um o Stellen bedeutet, vgl. Abb. 27) verschoben werden muss, so dass in dem Beispiel in Abbildung 28 beim ersten H des Schlüsselwortes ein I resultiert.
- Der zweite Buchstabe des Schlüsselworts "hallo" ist ein A. Dieses steht an der 1. Stelle des Alphabets, so dass der Buchstabe, der mit dem A verschlüsselt werden soll, gar nicht verschoben werden muss, so dass in dem Beispiel in Abbildung 28 beim ersten A des Schlüsselwortes ein E bleibt.
- Der dritte und vierte Buchstabe des Schlüsselworts "hallo" ist jeweils ein L. Dieses steht an der 12. Stelle des Alphabets, so dass der Buchstabe, der mit dem L verschlüsselt werden soll, um elf Buchstaben im Alphabet (da der Buchstabe A eine Verschiebung um o Stellen bedeutet) verschoben werden muss, so dass in dem Beispiel in Abbildung 28 beim ersten L des Schlüsselwortes ein H und beim zweiten L des Schlüsselwortes ein P resultieren.
- Der fünfte Buchstabe des Schlüsselworts "hallo" ist ein O. Dieses steht an der 15. Stelle des Alphabets, so dass der Buchstabe, der mit dem O verschlüsselt werden soll, um vierzehn Buchstaben im Alphabet (da der Buchstabe A eine Verschiebung um o Stellen bedeutet) verschoben werden muss, so dass in dem Beispiel in Abbildung 28 beim ersten O des Schlüsselwortes ein U resultiert.

kun indstilles på 26 forskellige måder. Dérudover kan man ved at analysere hyppigheden af de anvendte bogstaver hurtigt afkode krypteringen, såfremt man kender fordelingshyppigheden af bogstaverne på det pågældende sprog.

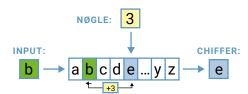


Fig. 27: Eksempel på en Cæsar-kryptering

Kryptering med Vigenère-chiffre (opkaldt efter Blaise de Vigenère, 1523–1596) er en forbedring af Cæsar-krypteringen. Også her flyttes hvert bogstav indenfor alfabetet, men ikke hvert bogstav med samme antal positioner. Nøglen til forståelse er her ikke kun eet, men flere tal, der er kodet som nøgleord. Hvis man vil kryptere en meddelelse v. h. a. nøgleordet "hallo", så foregår det på følgende måde:

- Nøgleordet skrives så mange gange neden under den besked, der skal krypteres, at hvert af beskedens bogstaver tilknyttes et bogstav i nøgleordet.
- · Selve nøgleordet består af fem bogstaver.
- Det første bogstav i nøgleordet "hallo" er et H. I alfabetet står dette i 7. position, således, at det bogstav, som skal krypteres med H, skal flyttes seks bogstavpositioner i alfabetet (idet bogstav A er lig med en forskydning på nul positioner, jf. fig. 27), hvilket i eksemplet i fig. 28 medfører, at det første H i nøgleordet resulterer i et I.
- Det andet bogstav i nøgleordet "hallo" er et A. I alfabetet står dette i 1. position, således, at det bogstav, som skal krypteres med A, slet ikke skal flyttes/forskydes, hvilket i eksemplet i fig. 28 medfører, at det første A i nøgleordet ender med at være et E.
- Det tredje og fjerde bogstav i nøgleordet "hallo" er hhv. et L. Det står i alfabetet i 12. position, således, at det bogstav, som skal krypteres med L, skal flyttes elleve bogstav-positioner i alfabetet (idet bogstav A er lig med en forskydning på nul positioner), hvilket i eksemplet i fig. 28 medfører, at det første L i nøgleordet resulterer i et H og det andet L i nøgleordet resulterer i et P.
- Det femte bogstav i nøgleordet "hallo" er et O. I alfabetet står dette i 15. position, således, at det bogstav, som skal krypteres med O, skal flyttes fjorten bogstavpositioner i alfabetet (idet bogstav A er lig med en forskydning på nul positioner), hvilket i eksemplet i fig. 28 medfører, at det første O i nøgleordet resulterer i et U.

Mathematisch kann hier die Division mit Rest (oder Modulo-Rechnung) herangezogen werden: Der x. Buchstabe einer Nachricht wird dabei durch die Länge des Schlüsselwortes geteilt. Das Ergebnis wird in der "Rest"-Schreibweise notiert. Bei gleichem Rest wird der jeweilige Buchstabe der Nachricht mit dem gleichen Schlüsselbuchstaben verschlüsselt (vgl. Tab. e):

Buchstabe der Nachricht	Länge de Schlüsselwortes	Formel zur Berechnung	Ergebnis	Buchstabe des Schlüsselwortes	Verschiebungszahl (vgl. Abb. 27)
1.	5	1:5=	o Rest 1	Н	6
2.	5	2:5=	o Rest 2	А	0
3.	5	3:5=	o Rest 3	L	11
4.	5	4:5=	o Rest 4	L	11
5.	5	5:5=	1 Rest 0	0	14
6.	5	6:5=	1 Rest 1	Н	6
7.	5	7:5=	1 Rest 2	А	0
8.	5	8:5=	1 Rest 3	L	11
9.	5	9:5=	1 Rest 4	L	11
10.	5	10:5=	2 Rest 0	0	14
11.	5	11:5=	2 Rest 1	Н	6
12.	5	12:5=	2 Rest 2	А	0
13.	5	13:5=	2 Rest 3	L	11
14.	5	14:5=	2 Rest 4	L	11
15.	5	15:5=	3 Rest 0	0	14

Tab. e: Schrittweises Verfahren zur Ermittlung der Verschlüsselung über die Division mit Rest

Diese Verschlüsselung eines jeden Buchstabens wird so fortgesetzt, so dass für die Nachricht in Abbildung 28 der 1., 6., 11., 16. etc. Buchstaben der Nachricht um sechs Buchstaben, der 2., 7., 12. etc. Buchstaben der Nachricht um null Buchstaben, der 3., 4., 8., 9., 13., 14. etc. Buchstaben der Nachricht um elf Buchstaben und den 5., 10., 15. etc. Buchstaben der Nachricht um vierzehn Buchstaben verschoben wird.

NACHRICHT: SCHLÜSSEL: CHIFFRE:



Abb. 28: Beispiel einer Vigenère-Verschlüsselung

Matematisk kan man her anvende division med rest (eller modulo-regning): Det x. bogstav i en besked divideres med nøgleordets længde. Resultatet noteres i form af "rest"-skrivning. Ved samme rest krypteres det respektive bogstav i en besked med det samme nøglebogstav (jf. tab. e):

Bogstav i beske- den	Kodeordets længde	Beregningsfor- mel	Resultat	Bogstav i nøgle- ordet	Forskydningstal (jf. fig. 27)
1.	5	1:5=	0 rest 1	Н	6
2.	5	2:5=	0 rest 2	А	0
3.	5	3:5=	o rest 3	L	11
4.	5	4:5=	o rest 4	L	11
5.	5	5:5=	1 rest 0	0	14
6.	5	6:5=	1 rest 1	Н	6
7.	5	7:5=	1 rest 2	А	0
8.	5	8:5=	1 rest 3	L	11
9.	5	9:5=	1 rest 4	L	11
10.	5	10:5=	2 rest 0	0	14
11.	5	11:5=	2 rest 1	Н	6
12.	5	12:5=	2 rest 2	А	0
13.	5	13:5=	2 rest 3	L	11
14.	5	14:5=	2 rest 4	L	11
15.	5	15:5=	3 rest 0	0	14

Tab. e: Trinvis metode til at afdække en kryptering gennem division med rest

Denne kryptering af hvert bogstav fortsættes på en sådan måde, at der for beskeden i figur 28 sker en forskydning af beskedens 1., 6., 11., 16. osv. bogstav med seks bogstaver, af beskedens 2., 7., 12. osv. bogstav med nul bogstaver, af beskedens 3., 4., 8., 9., 13., 14. osv. bogstav med elleve bogstaver og af beskedens 5., 10., 15. osv. bogstav med fjorten bogstaver.



Fig. 28: Eksempel på en Vigenère-kryptering

Das Entschlüsseln dieser Chiffren ist ohne Kenntnis des Schlüssels deutlich aufwendiger als die Skytale- oder Cäsar-Verschlüsselung, da die Länge des Schlüsselwortes und die Buchstabenfolge des Schlüsselwortes unzählige Kombinationsmöglichkeiten liefern, so dass ein Ausprobieren quasi unmöglich wird. Um allein alle möglichen 5-stelligen Schlüsselworte auszuprobieren, müsste man 26 × 26 × 26 × 26 × 26=11.881.376 Varianten durchprobieren. Auch die bei der Cäsar-Verschlüsselung noch mögliche Methode der Häufigkeitsanalyse bringt weniger schnell Erfolg, weil nicht jeder Buchstaben mit dem gleichen Schlüssel verschlüsselt wird und das "e" in obigem Beispiel mal mit "e", mal mit "p" und mal mit "s" verschlüsselt worden ist. Hier ist ein Ausprobieren über die Häufigkeit überhaupt nur dann sinnvoll, wenn z.B. die Schlüssellänge bekannt ist.

Heutzutage reichen die vorgestellten klassischen Verschlüsselungsmethoden aufgrund des Einsatzes fortgeschrittener effizienter Computertechnologie nicht mehr aus, um eine sichere Verschlüsselung zu garantieren. Dies kann man auch Schüler\*innen erfahren lassen, indem sie die klassischen, einstmals sicheren Verschlüsselungen mittels einfacher Computerprogramme in Sekunden knacken können. Deshalb werden heute Kombinationen aus den vorhergehenden Verfahren sowie neuere Ansätze genutzt. So besteht ein Teil des Advanced Encryption Standard (AES), was z. B. bei der Verschlüsselung über W-LAN, Skype oder WhatsApp eingesetzt wird, aus der Kombination der drei o.g. Chiffrierungsverfahren (s. Abb. 29):

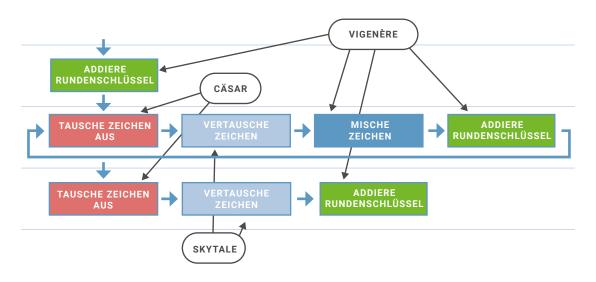


Abb. 29: Teil des Advanced Encryption Standard (AES)-Verschlüsselungsverfahrens

Ein weiterer Aspekt ist, dass bei allen vorangehenden Verfahren beim Ver- und Entschlüsseln der gleiche Schlüssel verwendet wird. Dieser Schlüssel muss einmal sicher zwischen Sender und Empfänger ausgetauscht werden. Gelingt aber eine sichere Übermittlung des Schlüssels, könnte man ja auf diese Art auch gleich die Nachricht selbst übermitteln und bräuchte sie nicht erst zu verschlüsseln. Hier setzt ein moderneres Verschlüsselungsverfahren an, welches die Nachricht mit einem Schlüssel verschlüsselt und mit einem anderen entschlüsselt. Es braucht also kein Schlüssel übertragen zu werden. Im sogenannten *Public-Key-*Verfahren hat jeder Empfänger zwei Schlüssel, einen öffentlichen und einen privaten. Der öffentliche Schlüssel ist

En afkodning af disse chiffre vil uden kendskab til nøglen være langt mere krævende end for en skytale- eller Cæsar-kryptering, idet længden af nøgleordet og bogstavsekvensen i nøgleordet medfører utallige kombinationsmuligheder, hvorfor prøve-sig-frem-metoden her tilnær-melsesvis vil være en umulighed. Alene for at afprøve alle mulige 5-cifrede nøgleord, ville man skulle prøve 26 × 26 × 26 × 26 × 26 = 11.881.376 varianter. Og også frekvensanalysemetoden (hyppighed), som endnu er mulig med Cæsar-krypteringen, vil ikke have store chancer for at lykkes her, fordi ikke hvert bogstav er krypteret med den samme nøgle, og f. eks. bogstav "e" i ovenstående eksempel nogle gange er krypteret med et "e", andre gange med et "p" og igen andre gange med et "s". I det hele taget vil prøve-sig-frem v. h. a. hyppighedsmetoden kun give mening, hvis man kender nøglens længde.

Nu om dage er de ovenfor viste klassiske krypteringsmetoder ikke længere gode nok til at garantere sikker kryptering p. g. a. brugen af avanceret, effektiv computerteknologi. Dette kan man også give eleverne mulighed for at opleve ved at lade dem knække den klassiske (og i sin tid sikre) kryptering på få sekunder v. h. a. enkle computerprogrammer. Derfor bruges nu om dage kombinationer af de tidligere metoder og nyere tilgange. Således består en del af Advanced Encryption Standard (AES), der f. eks. bruges i forbindelse med kryptering af WiFi, Skype® eller WhatsApp®, af en kombination af de tre ovennævnte chiffreringsmetoder (se fig. 29):

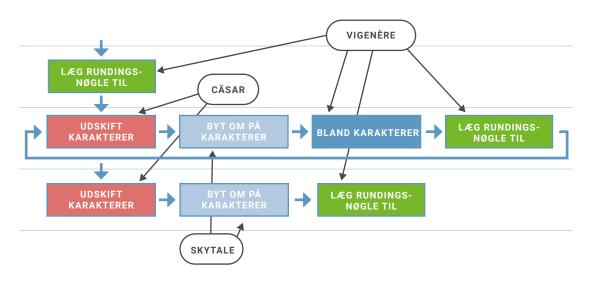


Fig. 29: Dele af krypteringsmetoden Advanced Encryption Standard (AES)

Et andet aspekt er, at der for alle tidligere brugte metoders vedkommende anvendes én og samme nøgle til kryptering og dekryptering. Denne nøgle skal afsender og modtager én gang være blevet gjort bekendt med på en sikker måde. Hvis det imidlertid lykkes at meddele/ fremsende nøglen på en sikker måde... jamen, så kan man jo lige så godt straks bruge denne metode til at fremsende selve beskeden – uden at kryptere den først. Det er her, en mere moderne krypteringsmetode kommer ind i billedet, som krypterer beskeden med én nøgle og dekrypterer den med en anden – hvorfor det ikke er nødvendigt at fremsende en nøgle. Med den såkaldte *public-key*-metode har hver modtager to nøgler, en offentlig og en privat. Den offentlige nøgle er tilgængelig for alle og enhver og anvendes af afsenderen til at kryptere data,

für jeden zugänglich und wird vom Sender zum Verschlüsseln von Daten für den Besitzer des privaten Schlüssels genutzt; der private ist nur dem Empfänger bekannt und wird von diesem zum Entschlüsseln verwendet. Genutzt wird diese Art der Verschlüsselung z. B. im E-Mail-Verkehr oder zur sicheren Kommunikation eines Web-Browsers mit einem Server. Grundlage für das Verfahren sind komplexe mathematische Berechnungsmethoden. Dazu werden Nachrichten in Zahlen übersetzt und die Zahlen mit dem öffentlichen Schlüssel verrechnet (Verschlüsseln). Das einfache Zurückrechnen (Entschlüsseln) geht nur mit dem privaten Schlüssel. Ohne den privaten Schlüssel ist der Rechenaufwand bei den derzeitigen Verfahren so groß, dass die schnellsten Computer der Welt mehrere Monate oder gar Jahre benötigen würden.

# 1.4 Bezug zum Unterrichtsfach

Durch das im Rahmen des PANaMa-Projekts entwickelte Material werden sowohl fachspezifische Kompetenzen zur Mathematik oder zum Technologieverständnis als auch fachübergreifende Kompetenzen zum Verständnis eines algorithmischen Denkens sowie zum Einsatz digitaler Prozesse im Alltag angesprochen.

Mathematische Konzepte, die insbesondere durch die Auseinandersetzung mit der Kryptographie adressiert werden können, sind in Tabelle f genannt.

Adressiertes mathematisches Konzept	Bsp. aus der Kryptographie
Zahlentheorie	Modulorechnung bei Cäsar & Vigenère bzw. Diffie-Hellman & RSA
Abbildungen	Insb. Skytale (Permutationen)
Wahrscheinlichkeitstheorie	Aufwandsberechnung bei Sicherheit von Verschlüsselungen / Häufigkeitsanalyse bei Cäsar & Vigenère
Exponetielles Wachstum	Anzahl der möglichen Schlüssel beim Vigenère-Verfahren
Exponentialfunktion	Diffie-Hellman & RSA

Tab. f: Mathematische Konzepte mit Bezug zur Kryptographie

Dabei werden in drei Bereichen Kompetenzen der Lernenden gefördert:

Schüler\*innen lernen, mathematisch zu argumentieren, d. h. sie erläutern oder entwickeln simple bis komplexe Argumentationen, beschreiben und begründen Lösungswege,
bewerten Ergebnisse im Kontext und erläutern Zusammenhänge und Strukturen.

som er bestemt til ejeren af den private nøgle. Den private nøgle kendes kun af modtageren og bruges af modtageren til dekryptering. Denne type kryptering bruges f. eks. indenfor e-mail-trafik eller til sikker kommunikation mellem ens webbrowser og en server. Det hele er baseret på komplekse matematiske beregningsmetoder, hvor beskeder oversættes til tal, og disse tal så regnes igennem (krypteres) v. h. a. den offentlige nøgle. Simpel tilbageregning (dekryptering) er kun mulig med den private nøgle. Uden den private nøgle ville regneindsatsen (comuting) med de nuværende metoder være så stor, at selv de hurtigste computere i verden ville være flere måneder eller endog år om det.

# 1.4 Relation til undervisningsfaget

Det materiale, der er udviklet i forbindelse med PANaMa-projektet, vedrører såvel fagspecifikke kompetencer indenfor matematik eller forståelse af teknologi som også tværfaglige kompetencer vedr. forståelse af algoritmisk tænkning samt anvendelsen af digitale processer i hverdagen.

Matematiske koncepter, som især kan adresseres ved at beskæftige sig med kryptografi, er anført i tabel f.

Adresseret matematisk koncept	Eks. fra kryptografien
Talteori	Modulo-regning med Cæsar & Vigenère resp. Diffie-Hellman & RSA
Figurer/billeder	Især skytale (permutationer)
Sandsynlighedsteori	Ressourceberegning i f. m. sikkerheden af krypteringer / hyppighedsanalyse for Cæsar & Vigenère
Eksponentiel vækst	Antal mulige nøgler i f. m. Vigenère-metoden
Eksponentiel funktion	Diffie-Hellman & RSA

Tab. f: Matematiske koncepter med relation til kryptografi

I den forbindelse fremmes elevernes kompetencer indenfor tre områder:

- Eleverne lærer at argumentere matematisk, d. v. s. de forklarer eller udvikler lige fra simple til komplekse argumentationer, beskriver og begrunder løsningsmuligheder, vurderer resultater i en kontekst og redegør for sammenhænge og strukturer.
- Eleverne lærer at løse problemer på en matematisk måde, d. v. s. de erkender og formulerer problemer, vælger problemløsningsstrategier og gennemfører dem samt vurderer løsninger og løsningsmuligheder.

- Schüler\*innen lernen, Probleme mathematisch zu lösen, d. h. sie erkennen und formulieren Probleme, wählen Problemlösestrategien aus und setzen diese ein und bewerten Lösungen und Lösungswege.
- Schüler\*innen lernen, mathematisch zu modellieren, d. h. sie konstruieren eigenständig passende mathematische Modelle, wenden vorgegebene Modelle an, verstehen und bewerten diese und sind in der Lage, beim Mathematisieren, Interpretieren und Validieren situationsgerecht zwischen Realsituation und mathematischer Beschreibung zu wechseln

Die fachspezifischen Kompetenzen zum Technologieverständnis können in verschiedene Bereiche gruppiert werden:

#### a. Im Bereich Fachwissen

- identifizieren und vergleichen von verschiedenen Robotertypen (Software und konkrete Roboter)
- erwerben Programmierkenntnisse für einen oder mehrere Robotertypen
- · unterteilen komplexe Aufgaben in Teilschritte
- formulieren Anweisungen unterschiedlich präzise, je nach intendiertem Ziel der Aufgabe
- formulieren Probleme so, dass sie von einem Computer/Roboter gelöst werden können
- sortieren Daten in einer logischen Anordnung und analysieren diese
- · repräsentieren Daten durch Abstraktion, wie z. B. Modelle und Simulationen
- formulieren Lösungsansätze durch algorithmisches Denken (Nutzung einer schlüssigen Abfolge einzelner Schritte)
- identifizieren und analysieren mögliche Lösungen mit dem Ziel, bezogen auf die Kombination aus Schritten und Resourcen den möglichst effizientisten und effektivsten Lösungsweg zu finden, und setzen diese um
- generalisieren und übertragen diesen Problemlöseprozess auf viele verschiedene Situationen

#### b. im Bereich Kommunikation über bzw. Bewertung von Technologieverständnis

- einschätzen der Möglichkeiten und Grenzen der Technologie für unterschiedliche Zwecke
- · erstellen Prognosen für die Entwicklung der Technologien auf bestimmten Gebieten
- einschätzen der Bedeutung der Technologie für die gesellschaftliche Entwicklung

Die oben beschriebenen Kompetenzen zum Technologieverständnis können durch eine fachspezifische Einbindung in verschiedenen Schulfächern erworben werden. So kann bei digitaler Datenerhebung in den Naturwissenschaften konkret auf das Programmieren eingegangen
werden. Ebenso lassen sich hier Zusammenhänge zwischen der technischen und der gesellschaftlichen Entwicklung aufzeigen sowie die Nachhaltigkeit und Grenzen einer Technologie
einordnen. In Mathematik lernen Schüler\*innen u.a. mithilfe digitaler Simulationen Modellierungsprozesse durchzuführen oder erwerben ein Verständnis von Kriterien zur Beurteilung von
mathematischen Modellen. Hinzu kommen drei zentrale überfachliche Kompetenzen, die im
Rahmen dieses Kapitels erworben werden können:

· Beschreibung von Beobachtungen,

 Eleverne lærer at modellére på en matematisk måde, d. v. s. de konstruerer på egen hånd passende matematiske modeller, anvender forhåndsvalgte modeller, forstår og vurderer disse og er i forbindelse med matematisering, fortolkning og validering i stand til – alt efter, hvad der aktuelt er behov for – at skifte mellem situationen i realiteten og den matematiske beskrivelse.

De fagspecifikke kompetencer vedr. teknologisk forståelse kan opdeles i forskellige grupper:

#### a. Indenfor faglig viden:

- Identificere og sammenligne forskellige typer robotter (software og selve robotten).
- Opnå programmeringsviden vedr. én eller flere typer robotter.
- · Dele komplekse opgaver op i underopgaver.
- Formulere instruktioner med forskellig grad af præcision, alt efter, hvilke(t) mål der tilstræbes med opgaven.
- Formulere problemer på en sådan måde, at de kan løses v. h. a. en computer eller robot.
- · Sortere data i en logisk rækkefølge og analysere denne.
- Præsentere data gennem abstraktion, f. eks. modeller og simulation.
- Formulere løsningsmuligheder v. h. a. algoritmisk tænkning (gå trinvist frem i en konkluderende rækkefølge).
- Identificere og analysere mulige l\u00f8sninger med en m\u00e5lsætning om s\u00e4vidt muligt at finde frem til – og gennemf\u00f8re – den mest ressourcebesparende og effektive l\u00f8sningsvej, set i lyset af kombinationen af trin og ressourcer.
- Generalisere og overføre denne problemløsningsproces til mange forskellige situationer.

## b. Indenfor kommunikation om resp. bedømmelse af teknologisk forståelse:

- · Estimere teknologiens muligheder og grænser i forhold til forskelige formål.
- Opstille prognoser for, hvordan teknologien vil udvikle sig på bestemte områder.
- · Vurdere teknologiens betydning for den samfundsmæssige udvikling.

De ovenfor beskrevne kompetencer m. h. t. at forstå teknologi kan tilegnes gennem fagspecifik integration i forskellige skolefag. På den måde kan man konkret komme ind på emnet programmering i forbindelse med digital dataindsamling i de naturvidenskabelige fag. På samme måde kan der i den forbindelse peges på sammenhænge mellem den tekniske og den samfundsmæssige udvikling og drøftes bæredygtigheden af og grænserne for en bestemt teknologi. I matematikfaget lærer eleverne bl. a. gennem digitale simuleringer at udføre modelleringsprocesser eller opnå en forståelse af kriterierne for vurdering af matematiske modeller. Desuden ligger der her tre centrale tværfaglige kompetencer, som kan tilegnes i forbindelse med dette kapitel:

- · Beskrivelse af observationer.
- Opstilling af en algoritme: Beskrivelse af (og begrundelse for) en konkret procesplanlægning (hvorfor gør man hvad?).
- · Fortolkning af resultater.

- Erstellung eines Algorithmus: Beschreibung (und Begründung) einer konkreten Prozessplanung (warum mache ich was?)
- · Interpretation eines Ergebnisses.

Diese sind als fachübergreifend zu verstehen, denn sie können in jedem Fach eingesetzt werden. So sind in den Naturwissenschaften Beobachtungen von natürlichen Phänomenen Ausgangspunkte jeder Untersuchung. Diese müssen im Anfangsunterricht beschrieben und im Laufe der Schulzeit verstärkt interpretiert werden, oft über die Durchführung von zu planenden Experimenten. Die reflexive Betrachtung des durchgeführten Experiments führt dann zu einer Aussage darüber, ob dieses z. B. die formulierte Hypothese bestätigt oder widerlegt. Im Sprach- oder Kunstunterricht sollen z.B. Bilder oder Texte beschrieben und interpretiert werden; eigene Werke wie Aufsätze oder kreative Produkte sollten im Vorfeld strukturiert geplant werden, damit sie ihre Intention nicht verfehlen (und so z. B. eine geforderte Argumentation nicht zu einer bloßen Zusammenfassung wird). Auch im Mathematikunterricht ist ein Beobachten unerlässlich, wenn z. B. aus Aufgaben zunächst die relevanten Informationen extrahiert werden müssen. Um eine bestimmte Aufgabe zu lösen, muss ggf. ebenfalls ein Algorithmus aufgestellt werden und das Ergebnis u. a. im Hinblick auf dessen Plausibilität geprüft werden.

Durch die Behandlung von Aufgaben zum computational thinking erwerben oder vertiefen Schüler\*innen insbesondere auch folgende Fähigkeiten, welche den Lernenden sowohl in der Schule als auch darüber hinaus nützlich sind:

- Zuversicht darin, sich auch mit komplexen Problemen zu beschäftigen
- Beharrlichkeit darin, sich auch über einen längeren Zeitraum und bei auftretenden Hindernissen mit schwierigen Problemen zu beschäftigen
- Auseinandersetzung mit offenen Problemen
- Toleranz darin, nicht immer einen eindeutig richtigen Weg gehen zu wollen/können
- Teamfähigkeit und Kommunikation, so dass ein Ziel oder eine Lösung gemeinsam erreicht werden kann

Eine Schwierigkeit der offenen und auf Forschung angelegten Unterrichtsansätze ist, dass Lehrkräfte nur schwer vorhersehen können, auf welche Details sich die Schüler\*innen im Rahmen einer Untersuchung konzentrieren oder welche Aspekte sie interessant finden. Hier ist es notwendig, eine Balance zwischen den fachlichen bzw. überfachlichen Unterrichtszielen und dem Interesse der Schüler\*innen herzustellen, indem beide Aspekte an verschiedenen Stellen des Unterrichts nachgegangen wird.

Disse skal forstås tværfagligt, fordi de kan bruges i ethvert fag. Således er observationer af naturfænomener udgangspunktet for enhver undersøgelse i de naturvidenskabelige fag. De skal i starten af undervisningen snarere beskrives og længere henne i skoleforløbet snarere fortolkes – ofte ved at gennemføre planlagte eksperimenter. Den reflekterende betragtning af det gennemførte eksperiment fører derefter til et udsagn om, hvorvidt eksperimentet f. eks. bekræfter eller modbeviser den formulerede hypotese. I sprog- eller kunstundervisningen skal der f. eks. beskrives og fortolkes billeder eller tekster; egne værker såsom fristile eller kreative produkter skal planlægges på en struktureret måde på forhånd, så de ikke skyder ved siden af det satte mål (og f. eks. en krævet argumentation ikke ender med blot at være et resumé). Også i matematikundervisningen er observation en bydende nødvendighed, f. eks. hvis man i forbindelse med en opgave i første omgang skal ekstrahere de relevante oplysninger. For at kunne løse en bestemt opgave kan det også være nødvendigt at opstille en algoritme og kontrollere resultatet bl. a. med hensyn til dens validitet.

Ved at arbejde med opgaver omkring emnet computational thinking opnår eller udvider eleverne i særdeleshed også følgende evner – som de kan drage nytte af både i skolen og senere i livet:

- Selvtillid m. h. t. også at (turde) beskæftige sig med komplekse problemer.
- Ihærdighed m. t. t. at (ville) beskæftige sig med vanskelige problemer, selvom det tager tid og der muligvis opstår forhindringer undervejs.
- · Tage (kampen med) åbenlyse problemer op.
- Tolerance m. h. t. ikke altid at kunne eller ville gå den ene og eneste rigtige vej.
- Evnen til at kommunikere og samarbejde med andre, så man løfter i flok m. h. t. at nå et mål eller en løsning i fællesskab.

En vanskelighed med åbne og forskningsbaserede undervisningsmetoder er, at lærere har svært ved at forudsige, hvilke detaljer eleverne vil fokusere på i forbindelse med en undersøgelse, eller hvilke aspekter de finder interessante. Her er det nødvendigt at skabe en balance mellem de fagspecifikke og tværfaglige undervisningsmål og elevernes interesser ved at undersøge begge aspekter i forskellige trin undervejs i undervisningsforløbet.

# 2. KONKRETE UNTERRICHTSIDEEN

Um dem oben angesprochenen Charakter forschender Aufgaben gerecht zu werden, sollten alle Arbeitsaufträge und Aufgaben zunächst recht offen gestellt werden, damit Schüler\*innen kognitiv aktiviert werden können. Natürlich ist immer Unterstützung in Form strategischer Hilfen bereit zu halten, falls Schüler\*innen alleine nicht weiterkommen.

Alle Unterrichtsideen im Kontext des PANaMa-Projekts streben an, Bezüge zu Berufen und regionalen Unternehmen aufzuweisen. Dazu eigenen sich beispielsweise neben der Thematisierung der verschiedenen Einsatzarten von Robotern (vgl. die verschiedenen Aufgaben zu den Simulationen) auch Unternehmensbesuche. So nutzen fast alle produzierenden Unternehmen eine Art Fördermechanismus, bei der Dinge innerhalb des Unternehmens transportiert werden – reine Logistikunternehmen bieten sich hier ebenso an wie Klein- und mittelständische Unternehmen bzw. Familienbetriebe (z. B. die in Kapitel II dieses Bands erwähnten M. JÜRGENSEN GmbH & Co KG oder Anthon GmbH).

Ebenso bietet es sich an, eine Unterrichtseinheit zum Thema *Robotik und Technologie* in der Alltags- und Arbeitswelt mit der in Band 2, Kapitel I bereits vorgestellten Methode der schülerkuratierten Ausstellung zu präsentieren.

Im Folgenden werden Anregungen für unterschiedliche Unterrichtsprojekte vorgestellt, bei denen mit Software und mit verschiedenen Technologien gearbeitet werden kann: Im Bereich der Robotik mit a) Ozobots sowie b) LEGO MINDSTORMS® und im Bereich der Kryptographie mit c) verschiedenen Verschlüsselungstechnologien. Ausführlichere Anregungen zur Arbeit mit Arduinos finden Sie in Kap. VII. Das Material ist sowohl im Rahmen der Forschercamps mit Jugendlichen ausprobiert worden, als auch z. T. für den Einsatz im Unterricht aufbereitet worden. Dabei richtet es sich mit einer Länge von ca. 10 Unterrichtsstunden mit zusätzlichem Zeitaufwand für Unternehmensbesuch und Ausstellungsbau an Schüler\*innen der Klassenstufen 7–10.

Vorgestellt wird immer zunächst eine Begegnungsstunde mit dem jeweiligen System sowie im Anschluss ein bis drei "Simulationen" mit offenen Aufgaben. Zu diesen ist immer eine weitere Untergliederung des Problems in mögliche Unterstützungsschritte angegeben, die natürlich von den Schüler\*innen selbst erarbeitet werden sollen, soweit dies möglich ist. Die können aber auch als Hilfestellung dienen.

Hierbei werden auch Möglichkeiten und eventuelle Bezüge zum Arbeitsmarkt der Zukunft aufgezeigt, wodurch die Schüler\*innen einen Eindruck über die Bedeutung von Technologie gewinnen können. Die Länge der Aufgaben kann variabel gestaltet werden, so dass sie optimal in den jeweiligen Unterricht hineinpassen





Abb. 30: Konstruktion und Programmierung von Robotern bietet eine Vielzahl von neuen Kompetenzen für Jugendliche

Fig. 30: Design og programmering af robotter tilbyder en række nye færdigheder for unge mennesker

# 2. KONKRETE UNDERVISNINGSIDÉER

For at leve op til den ovennævnte karakter af forskningsopgaver skal alle arbejdsordrer og opgaver i første omgang udformes ganske åbent, så eleverne kan kan aktivere deres kognitive evner. Selvfølgelig bør der altid være mulighed for at støtte i form af strategisk hjælp, ifald eleverne ikke kan komme videre på egen hånd.

Alle undervisningsidéer i forbindelse med PANaMa-projektet sigter mod at skabe referencer og henvisninger til erhvervsfag og regionale virksomheder. I den forbindelse kan det udover at tage de forskellige anvendelsesformer for robotter op (jf. de forskellige opgaver omkring emnet simulation) også være en god idé at gennemføre virksomhedsbesøg. Næsten alle produktionsvirksomheder anvender én eller anden form for transportsystemer for at flytte emner fra ét sted til et andet internt i virksomheden. Rene logistikfirmaer er her lige så velegnede som små og mellemstore virksomheder eller familievirksomheder (f. eks. det i kapitel II i dette bind nævnte M. JÜRGENSEN GmbH & Co. KG eller fa. Anthon GmbH).

Det er også en god ide at præsentere en undervisningsenhed om emnet *robotteknologi* og *teknologi* i hverdagen og arbejdslivet ved hjælp af metoden med en elevgennemført udstilling, der allerede blev præsenteret i bind 2, kapitel I.

I det følgende præsenteres forslag til forskellige undervisningsprojekter, hvor der kan arbejdes med software og forskellige teknologier: Inden for robotteknologi med a) Ozobots og b) LEGO MINDSTORMS® og indenfor kryptografi med c) forskellige krypteringsteknologier. Mere detaljerede forslag til arbejde med Arduinos findes i kap. VII. Materialet er såvel blevet afprøvet i forbindelse med forskningslejre med unge som også delvis behandlet/opgraderet til brug i undervisningen. Med en længde på ca. 10 undervisningstimer plus ekstra tid til firmabesøg og udstillingsklargøring henvender det sig til elever på 7. til 10. klassetrin.

Der indledes altid med en introduktionstime i forhold til det respektive system efterfulgt af én til tre "simuleringer" med åbne opgaver. Vedr. disse er der altid en yderligere opdeling af problemet i mulige støttetrin, som selvfølgelig skal udarbejdes af eleverne selv – så vidt muligt. Men de kan også anvendes som hjælp.

I den forbindelse vises også muligheder og eventuelle referencer i forhold til fremtidens arbejdsmarked, hvilket giver eleverne et indtryk af teknologiens betydning. Opgavernes længde kan varieres, så de passer optimalt ind i den respektive undervisning.

# Aktivierung von Vorwissen der Lernenden zum Thema Robotik

Vorschlag für eine Einstiegsstunde (90 Min)

Phase	Einführung in das Thema Roboter
Einführung in das Thema 15 Min.	Die Lehrkraft präsentiert das Thema Roboter über Fragen oder Bilder mit der Zielfrage: Wo begegnen euch im Alltag Roboter und was macht einen Roboter eigentlich aus?
Schülerarbeit 10-15 Min.	In Gruppen von 3–4 Schülern*innen werden Vorschläge für die Definition eines Roboters erarbeitet.  Die Vorschläge können auf einer Videowand angezeigt werden, z. B. Padlet, so dass die vielen verschiedenen Vorschläge sichtbar werden.
Schülerarbeit: Was sagen die anderen zu den Definitionen eines Roboters?	Die Schüler*innen untersuchen die Roboter-Definitionen der anderen im Hin- blick auf Unterschiede und Gemeinsamkeiten. Hier können beispielsweise folgende Links einbezogen werden:
	https://de.wikipedia.org/wiki/Roboter https://www.planet-wissen.de/technik/computer_und_roboter/roboter_ mechanische_helfer/index.html http://www.roboterwelt.de/magazin/wann-ist-ein-roboter-ein-roboter/ http://whatis.techtarget.com/definition/robot-insect-robot-autonomous-robot https://klexikon.zum.de/wiki/Roboter
Plenum 10-15 Min.	Erörterung der Definitionen. Kann die Klasse gemeinsam eine Definition erstellen, der alle zustimmen? Die gemeinschaftliche Definition wird dem Padlet hinzugefügt.
Roboter 15 Min.	Die Klasse diskutiert Ideen für die Gegenwart und die Zukunft mit Robotern. Hierzu zählen Problemstellungen, mit denen sich die Gesellschaft befassen muss. Die Klasse erstellt gemeinsam eine Liste bzw. Mindmap möglicher Themen. Auch hier können über Bilder Impulse gesetzt werden, z.B.:
	<ul> <li>Was ist Automatisierung bzw. Robotik? Wie funktionieren Roboter?</li> <li>Vor- und Nachteile der Automatisierung</li> <li>Arten der Automatisierung/Roboter</li> <li>Wo sehen wir sie heute?</li> <li>Wo befindet sie sich bereits heute, jedoch versteckt (z. B. in automatisierten Parkhäusern oder Suchrobotern im Internet)?</li> <li>Wo vermuten wir sie in der Zukunft?</li> <li>Roboter/Automatisierung im Berufsleben</li> </ul>
	<ul> <li>Jobmöglichkeiten und Ausbildung</li> <li>Roboter/Automation und Ethik</li> <li>Die Unterrichtsstunde wird mit einer Zusammenfassung beendet sowie mit einem Hinweis auf den Schwerpunkt der nächsten Stunde.</li> </ul>
104	

# Aktivering af elevernes eksisterende viden om emnet robotteknologi

Forslag til en introduktionstime (90 min.)

Fase	Introduktion til emnet robotter
Introduktion til emnet 15 min.	Læreren præsenterer emnet robotter v. h. a. spørgsmål eller billeder og spør- ger så: Hvor i jeres hverdag møder I robotter, og hvad er det, der gør en robot til en robot?
Elevopgave 10–15 min.	I grupper på 3–4 elever udarbejdes forslag til en definition af, hvad en robot er. Forslagene kan vises på en videovæg, f. eks. Padlet, så de forskellige forslag kan visualiseres.
Elevopgave: Hvad mener de andre om definitionerne af en robot? 20 min.	Eleverne undersøger hinandens robotdefinitioner m. h. p. at identificere forskelle og ligheder. I den forbindelse kan der eksempelvis inddrages følgende links:
	https://de.wikipedia.org/wiki/Roboter https://www.planet-wissen.de/technik/computer_und_roboter/roboter_ mechanische_helfer/index.html http://www.roboterwelt.de/magazin/wann-ist-ein-roboter-ein-roboter/ http://whatis.techtarget.com/definition/robot-insect-robot-autonomous-robot https://klexikon.zum.de/wiki/Roboter
Plenum 10−15 min.	Debat om definitionerne. Kan klassen komme frem til en fælles definition, som alle kan godkende? Den fælles definition skrives på Padletten.
Robotter 15 min.	Klassen drøfter idéer vedr. robotter for nutiden og fremtiden – dériblandt problemstillinger, som samfundet kommer til at (skulle) beskæftige sig med. Klassen udarbejder i fællesskab en liste resp. et tankekort (mindmap) med mulige emner. Også her kan der gives impulser v. h. a. billeder, f. eks.:
	<ul> <li>Hvad er automatisering og robotteknologi? Hvordan fungerer robotter?</li> <li>Automatiseringens fordele og ulemper.</li> <li>Typer automatisering/robotter.</li> <li>Hvor møder vi dem nu om dage?</li> <li>Hvor findes de allerede, men uden at vi sér dem (f. eks. i automatiserede parkeringsanlæg eller som søgerobotter på internettet)?</li> <li>Hvor forventer vi at finde dem i fremtiden?</li> <li>Robotter/automatisering i arbejdslivet.</li> <li>Jobmuligheder og uddannelse.</li> <li>Robotter/automation og etiske spørgsmål.</li> </ul>
	Undervisningstimen afsluttes med et resumé samt orientering om, hvad der vil være fokus på i den næste time.

#### Mögliche Ideen für Bildimpulse:

- Nutzung in Produktion (monotone Arbeit, Gefahr, Sicherheit, Einseitigkeit usw.)
- Nutzung im Transportbereich (autonome Autos, Flurförderfahrzeug)
- Zukünftige Nutzung (Science Fiction?, fliegende Autos?)
- Ethik (Arbeitskraft durch Roboter vs. Menschen; Verdrängung von Arbeitskraft vs. Generierung neuer Jobs; "Verantwortung" bei autonom fahrenden Autos, z. B. Dilemma bei einem unvermeidbaren Unfall: In wen soll das Auto hineinfahren? Wer soll sterben?)

Aus den verschiedenen aufgeworfenen Themen der Mindmap können je nach Lerngruppe unterschiedliche Schwerpunkte gewählt werden. Im Folgenden wird der Schwerpunkt "Wie funktionieren Roboter?" gewählt, an dem exemplarisch die Kompetenz des *computational thinking* vertieft werden kann.

#### Handeln wie ein Roboter

Um Schüler\*innen nachvollziehen zu lassen, was Roboter machen, bietet es sich an, sie zu Beginn der Einheit agieren zu lassen, als seien sie Roboter.

In Partnerarbeit gibt ein Teil der Klasse Anweisungen und der anderen Teil befolgt diese. Jeweils 10 Anweisungen sind zuvor von der Lehrkraft auf Kärtchen notiert, welche jedes Paar erhält. Die Anweisungen sollten sich in ihrer Komplexität unterscheiden, z. B.:

- a. "gehe 3 Schritte vor"
- b. "hebe den rechten Fuß bis auf Kniehöhe"
- c. "gehe 1 Meter rückwärts, dreh dich um 90 Grad nach rechts, geh in die Hocke und lege deine rechte Hand mit ausgestrecktem Arm auf dem Boden".

Jede\*r Schüler\*in gibt 5 Anweisungen und führt selbst 5 Anweisungen aus. Im Idealfall werden die identischen Anweisungen gleichzeitig ausgeführt, so dass die zuschauenden Schüler\*innen die Ausführungen vergleichen können. Anschließend wird im Plenum diskutiert, wie gelungen die Aufgaben umgesetzt worden sind.

Diese Aufgabe enthält mehrere Lernziele. So werden sich Schüler\*innen bewusst,

 dass Roboter nur das tun, was ihnen gesagt wird. Die Anweisung wird nicht interpretiert oder hinterfragt. Sie wird aber sehr wohl

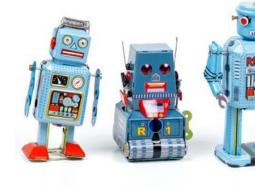


Fig. 31

#### Mulige idéer til billedimpulser:

- Anvendelse indenfor produktion (monotont arbejde, farligt, usikkert, énsidigt osv.)
- Anvendelse på transportområdet (selvkørende biler, transportvogne, gaffeltrucks o. l.)
- Anvendelse i fremtiden (science fiction?, flyvende biler?)
- Etiske spørgsmål (robotter vs. mennesker; fortrængning/nedlæggelse af jobs vs. skabelse af nye arbejdspladser; "ansvar/hæftelse" i forb. m. selvkørende biler (f. eks. dilemmaet, hvis ulykken er uundgåelig: Hvem skal bilen køre ind i? Pensionisten eller teenageren? Hvem skal 'dødsdømmes'?)

Afhængig af læringsgruppen kan der vælges forskellige prioriteter blandt de forskellige emner, der rejses på tankekortet. I det følgende er der valgt at sætte fokus på spørgsmålet: "Hvordan fungerer robotter?", hvor der eksemplarisk kan gås i dybden med kompetencen computational thinking.

#### At agére som en robot

For at få eleverne til at forstå, hvad robotter laver, er det i starten af forløbet en god idé at lade dem agére, som om de selv var robotter.

Der gennemføres partnerarbejde, hvor den ene del af klassen giver instruktioner, og den anden del følger dem. Hhv. 10 instruktioner har læreren forinden noteret på nogle kort, som hvert par modtager. Instruktionerne bør være forskellige m. h. t. deres kompleksitet, f. eks.:

- a. "Gå 3 skridt frem".
- b. "Løft højre fod op til knæhøjde".
- c. "Gå 1 meter baglæns, drej 90° til højre, sæt dig i hug og læg højre hånd på gulvet med udstrakt arm".

Hver elev giver 5 instruktioner og udfører 5 instruktioner. I bedste fald udføres identiske instruktioner på samme tid, så tilskuereleverne kan sammenligne præstationerne. Bagefter diskuteres i plenum, hvor dygtige man hhv. har været til at udføre opgaverne.

Denne opgave indeholder flere læringsmål. Eleverne bliver således bevidste om,

at robotter kun gør, hvad de får besked på. Instruktioner fortolkes ikke og drages ikke i tvivl – men udføres i givet fald meget forskelligt, hvis de ikke er formuleret præcist nok. For eksempel er det i b) ikke specificeret, på hvilken måde højre knæ skal vinkles (foran



unterschiedlich ausgeführt, wenn sie nicht präzise genug formuliert worden ist. So ist bei b) z. B. nicht vorgegeben, in welcher Art das rechte Knie abgewinkelt werden soll (vor den Körper, seitlich vom Körper oder parallel zum linken Knie, so dass der rechte Fuß nach hinten zeigt) oder bei c) die Information nicht vorhanden, in welche Richtung der Arm ausgestreckt werden soll. Je nach intendiertem Prozessziel muss die Anweisung unterschiedlich präzise formuliert werden.

 dass komplexere Anweisungen immer in kleinere Bereiche unterteilt werden müssen, die dann in einer bestimmten Reihenfolge ausgeführt werden, wobei unterschiedliche Reihenfolgen manchmal zum gleichen Ziel führen können.

Diese beiden Lernziele sind äußerst hilfreich für die weiteren Aufgaben, da dabei stehts präzise Anweisungen gegeben werden müssen bzw. aus Programmierungsblöcken ausgewählt werden soll und meist sequenzielle Prozessabläufe aufgebaut werden. Beides sind elementare Teile des computational thinking.

Während die Ideen bis hier unabhängig vom verwendeten Roboter-System eingesetzt werden konnten, werden im Folgenden exemplarische Aufgaben vorgestellt, welche natürlich auch auf andere übertragbar sind. In unseren Erprobungen der Unterrichtsvorschläge unterstützte es die Motivation der Schüler\*innen deutlich, wenn am Ende die Lösungen zu den Aufgaben per Video dokumentiert und/oder präsentiert wurde.

kroppen, til siden væk fra kroppen eller parallelt med venstre knæ, så højre fod peger bagud). Eller i c), hvor der mangler information om, hvilken retning armen skal strækkes ud i. Afhængigt af det tilsigtede procesmål skal instruktionerne formuleres med forskellige niveauer af præcision;

at mere komplekse instruktioner altid skal opdeles i mindre "bidder", som derefter udføres i en bestemt rækkefølge – og hvor forskellige rækkefølger undertiden alligevel godt kan føre til det samme mål.

Disse to læringsmål er yderst tjenlige i forbindelse med de videre opgaver, idet der altid skal gives nøjagtige instruktioner – eller vælges instruktioner fra programmeringsblokke – og det sædvanligvis er sekventielle procesforløb, der skal sættes op. Begge dele indgår nødvendigvis elementært i *computational thinking*.

Hvor idéerne hidtil (frem til dette punkt) har kunnet anvendes uafhængigt af det applicerede robotsystem, vil der nedenfor blive præsenteret eksempler på opgaver for de tre systemer, som imidlertid naturligvis også kan overføres til andre. I vore forsøg med undervisningsforslag viste det sig at være markant ekstra motiverende for eleverne, når opgaveløsningen afslutningsvis blev dokumenteret og/eller præsenteret v. h. a. video.

#### a) Ozobot

Exemplarische Stunde zur Begegnung mit Ozobot Evo mit Fokus Blockprogrammierung (90 Min)

Phase	Bsp. aus der Kryptographie	
Einführung in das Thema des Tages 10 Min.	Der Klasse wird der Ozobot Evo vorgestellt. https://ozobot.com/stem-education/education-getting-started	
	Die Lehrkraft präsentiert kurz die Möglichkeiten des Ozobot Evo:	
	<ul><li>Farbprogrammierung</li><li>Blockprogrammierung</li></ul>	
Schülerarbeit I 30 Min.	Die Schüler*innen erschließen sich selbst durch Anwendung und Modifizierung eines vorhandenen Programms mit Ozo- Blockly, wie das Programmieren funktioniert.	
Schülerarbeit II 30 Min.	Die Schüler*innen sollen ein kleines Programm erstellen, mit dessen Hilfe Evo einen einfachen Parcours mit bestimmten Vorgaben durchläuft. Das Programm wird von den Schüler*in- nen von Grund auf erstellt. Es werden somit keine Beispiele für Codes angeboten.	
	Den Schüler*innen wird ein Parcours vorgegeben. Die Lehr- kraft nennt Ziele, die mit dem von den Schüler*innen erstell- ten Programm absolviert werden müssen, wie beispielsweise der Roboter soll	
	<ul> <li>nach links und nach rechts abbiegen,</li> <li>die Geschwindigkeit ändern,</li> <li>sich rückwärts bewegen,</li> <li>usw.</li> </ul>	
Zusammenfassung 20 Min.	Einige Schüler*innen demonstrieren ihre Codes und die re- sultierenden Bewegungen des Evo. Die Ergebnisse werden diskutiert.	
	Vorausschau auf die nächsten Unterrichtsstunden: Wir simulieren, wie Roboter navigieren. Ein Rasenmähroboter fährt willkürlich auf einer Strecke. Industrieroboter (und Ro- boter in der Landwirtschaft) folgen bestimmten Linien. Und manche Roboter navigieren anhand von Sensoren.	



Abb. 32: Die Beschäftigung mit den Ozobot-Minirobotern bietet für Schüler\*innen unterschiedlicher Altersstufen einen Einstieg in die Robotik

Fig. 32: Beskæftigelsen med Ozobot-minirobotter tilbyder studerende i forskellige aldre en introduktion til robotik



#### a) Ozobot

Eksemplarisk time for at lære Ozobot Evo at kende – fokus på blokprogrammering (90 min.)

Fase	Introduktion til Ozobot Evo
Introduktion til dagens emne	Klassen introduceres til Ozobot Evo. https://ozobot.com/stem-education/education-getting-started
	Læreren præsenterer kort mulighederne med en Ozobot Evo:
	<ul><li>Farveprogrammering.</li><li>Blokprogrammering.</li></ul>
Elevopgave I 30 min.	Gennem anvendelse og modificering af et eksisterende pro- gram med OzoBlockly skyder eleverne sig ind på, hvordan programmeringen fungerer.
Elevopgave II 30 min.	Eleverne laver et lille program, v. h. a. hvilket Evoen fuldfører et simpelt forløb, en kørekurs med nogle fastlagte punkter. Eleverne fremstiller programmet helt fra bunden, d. v. s. de får ikke tilbudt eksempler på nogen koder.
	Eleverne får instruks om et bestemt forløb / en bestemt kurs. Læreren nævner nogle mål, som skal genne
	svinge til højre og venstre,
	<ul><li>ændre sin hastighed,</li><li>bevæge sig baglæns,</li></ul>
	• OSV.
Resumé 20 min.	Nogle elever demonstrerer deres koder og Evoens resulterende bevægelser. Resultaterne diskuteres.
	Orientering om de næste undervisningstimer: Simulation af, hvordan robotter navigerer. En robotplæneklip- per kører tilfældigt rundt indenfor et areal. Industrirobotter (og landbrugsrobotter) følger nogle bestemte linjeforløb. Og igen andre robotter navigerer v. h. a. følere (sensorer).

Anschließend sind folgende Aufgaben denkbar, die als kleine Projekte durchgeführt werden können:

#### Simulation eines Rasenmähroboters mit Ozobot Evo

#### Aufgabe 1:

Bringt den Evo-Roboter dazu, sich wie ein handelsüblicher Rasenmähroboter zu verhalten. Ein Beispiel zur Funktion eines solchen Roboters findet ihr u. a. hier: https://de.wikipedia.org/wiki/Rasenm%C3%A4hroboter. Die meisten Mähroboter fahren in geraden Linien über die Rasenfläche. Wenn sie auf eine Kante treffen, biegen sie in einem willkürlichen Winkel ab und fahren auf einer geraden Linie weiter.

#### Folgende Impulse können zur Unterstützung herangezogen werden:

- 1. Überlegt, ob eine Skizze der Rasenfläche sinnvoll ist.
- 2. Wie müsste der Evo fahren, damit er sich wie ein Rasenmähroboter verhält? Erstellt ein Programm dazu und verbessert dieses.
- Macht es einen Unterschied, ob der Evo nach rechts oder links abbiegt, wenn er auf ein Hindernis trifft? Passt das Programm weiterhin an bis ihr mit eurer Lösung zufrieden seid.

#### Aufgabe 2:

Befestigt einen Stift an dem Evo und lasst euer Programm erneut laufen, wodurch die Fahrstrecke des Evo auf das Papier gezeichnet wird (entspricht dem gemähten Rasen). Verändert jetzt euer Programm, so dass in einer Zeit von 2 Minuten möglichst viel "Rasen gemäht" wird.

#### Folgende Impulse können zur Unterstützung herangezogen werden:

- Welche Wirkung hat die Veränderung des Abbiegwinkel auf die Effizienz des Rasenmähens? Probiert z. B. aus, was passiert, wenn der zuvor willkürliche Winkel einen konkreten Zahlenwert erhält.
- 2. Findet so das effektivste Programm bzw. die effektivsten Parameter.



Efterfølgende kunne man forestille sig følgende opgaver – udført i form af små projekter:

#### Simulation af en robotplæneklipper med Ozobot Evo

#### Opgave 1:

Få Evo-robotten til at agére som en almindelig robotplæneklipper, man kan gå ud og købe. Et eksempel vedr. funktion af en sådan robot ses f. eks. her: https://de.wikipedia.org/wiki/Rasenm%C3%A4hroboter. De fleste robotklippere kører henover plænens areal i lige linjer. Hvis de bumper ind i en kant, drejer de til siden i én eller anden tilfældig vinkel og fortsætter kørslen ligeud ad en lige linje.

#### Følgende impulser kan inddrages som støtte:

- 1. Overvej, om det vil være hensigtsmæssigt at lave en skitse af plænen.
- 2. På hvilken måde vil en Evo skulle køre, for at dens adfærd svarer til en robotplæneklippers? Lav et program til det og foretag forbedringer.
- 3. Gør det en forskel, om Evoen drejer til højre eller til venstre, når den støder ind i en forhindring? Fortsæt med at justere programmet, indtil I er tilfredse med jeres løsning.

#### Opgave 2:

Montér en skrivepen på Evoen og kør jeres program igen – hvorved kørselsforløbet tegnes efter på papir (svarende til de steder, græsset er blevet klippet). Lav nu programmet om på en sådan måde, at der indenfor 2 minutter "klippes" så meget "græs" som overhovedet muligt.

#### Følgende impulser kan inddrages som støtte:

- 1. På hvilken måde påvirker ændringen af drejevinklen plæneklipperens ressourceoptimering (efficiency)? Prøv jer f. eks. frem til, hvad der sker, hvis den hidtil tilfældige vinkel forsynes med en talværdi.
- 2. Find på den måde frem til det mest effektive program resp. de mest effektive parametre.



Abb. 33

Fig. 33

### Simulation eines Industrieroboters (hier Flurförderfahrzeug) mit Ozobot Evo

#### Aufgabe:

Bingt den Evo-Roboter dazu, sich wie ein handelsüblicher Industrieroboter (Flurförderfahrzeug) zu verhalten! Ein Beispiel zur Funktion eines solchen Roboter findet ihr u.a. hier: https://www.youtube.com/watch?v=PiP8UrVuLUM. Die meisten dieser Flurförderfahrzeuge holen an einer bestimmmten Stelle Pakete ab und befördern sie zu einer anderen Stelle. Programmiert euren Evo so, dass er in möglichst kurzer Zeit insgesamt 3 Paktete von der "Produktion" ins "Lager" bringt, dabei aber immer nur eins transportieren darf.

#### Folgende Impulse können zur Unterstützung herangezogen werden:

- 1. Programmiert die Fahrstrecke (hin und zurück) zunächst ohne Pakete. Was muss bei dem Beginn der Rückfahrt beachtet werden?
- 2. Die Pakete werden dem Evo per Hand vor die Schiebevorrichtung gestellt. Wie muss das Programm geändert werden, damit der Evo nur losfährt, wenn ein Paket vor ihn gestellt wird?
- Testet verschiedene Geschwindigkeiten, ohne dass beim Transport Paktete verlorengehen.

#### Simulation einer Straßenverkehrssituation mit Ozobot Evo

#### Aufgabe:

Bei dieser Aufgabe ist es das Ziel, 4 Evo-Roboter so zu programmieren, dass sie ausgehend von den vier unterschiedlichen "Startboxen" den Fahrspuren nur in der jeweils angegebenen Richtung folgen, alle mind. zweimal die innere und äußere Spur nutzen und am Ende wieder in ihre "Startbox" zurückkehren. Dabei sollen sie untereinander die Vorfahrt beachten (rechts vor links).

#### Folgende Impulse können zur Unterstützung herangezogen werden:

- 1. Macht es Sinn, alle Evos gleichzeitig zu programmieren?
- 2. Wie erfolgen die Spurwechsel?
- 3. Wie muss das Programm geändert werden, damit der Evo an Kreuzungen immer (aber auch nur dann) anhält, wenn von rechts ein weiterer Evo kommt (und der von rechts kommende weiterfährt)?
- 4. Wie können Zusammenstöße verhindert werden, wenn ein langsamerer Ozobot vor einem schnelleren fährt?

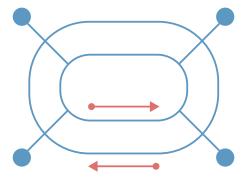


Abb. 35: Skizze zur Simulationsaufgabe mit den 4 Startboxen

Fig. 35: Skitse til simuleringsopgaven med de 4 startbokse

### Simulation af en industrirobot (her en transportvogn) med Ozobot Evo

#### Opgave:

Få Evo-robotten til at agére som en almindelig industrirobot (transportvogn), som man kan købe. Et eksempel vedr. funktion af sådan en robot ses bl. a. her: https://www.youtube.com/watch?v=PiP8UrVuLUM. De fleste af disse gulvbaserede transportvogne henter pakker et bestemt sted og transporterer dem et andet sted hen. Jeres Evo skal programmeres, så den indenfor kortest mulig tid flytter 3 pakker fra "produktionen" til "lageret" – men altid kun én pakke ad gangen.

#### Følgende impulser kan inddrages som støtte:

- 1. Start med at programmere kørestrækningen (frem og tilbage) uden pakker. Hvad er vigtigt, når tilbageturen skal til at starte?
- Pakkerne placeres manuelt foran Evoens skubbeanordning. På hvilken måde skal programmet nu ændres, for at Evoen kun(!) påbegynder tilbageturen, hvis den har fået en pakke stillet foran sig?
- 3. Afprøv forskellige hastigheder uden(!) at der går pakker tabt undervejs under transporten.



#### Opgaven:

Går her ud på at programmere 4 Evo-robotter på en sådan måde, at de med hver deres udgangspunkt fra fire forskellige "startbokse" følger kørebanerne i den hhv. tiltænkte retning (og kun den), at de alle minimum har kørt to gange på hhv. inder- og yderbanen – og tilsidst vender tilbage til deres "startboks igen". I den forbindelse skal de overholde den tyske forkørselsret "højre før venstre" (oversat til dansk vigepligt: Hold for dem, der kommer fra højre).

#### Følgende impulser kan inddrages som støtte:

- 1. Giver det mening at programmere alle Evoerne på én gang?
- 2. Hvordan foregår vognbaneskift?
- 3. På hvilken måde skal programmet ændres, for at Evoen altid standser op foran et kryds, hvis (men også kun hvis!) der kommer en anden Evo fra højre (og sidstnævnte samtidig fortsætter sin kørsel)?
- 4. Hvordan kan sammenstød undgås, hvis der kører en langsommere Evo foran en hurtigere, der kommer bagfra?



Abb. 34: Modernes Lagermanagement mit Einsatz von Transportrobotern

Fig. 34: Moderne lagerstyring med brug af transportrobotter

#### b) LEGO MINDSTORMS®

Exemplarische Stunde zum Kennenlernen mit LEGO MINDSTORMS® (90 Min)

Phase	Einführung zum LEGO MINDSTORMS®	
Einführung in das Thema des Tages 10 Min.	Der Klasse wird der LEGO MINDSTORMS®vorgestellt. https://education.lego.com/de-de/support/mindstorms-ev3/getting-started	
	Die Lehrkraft präsentiert kurz die Möglichkeiten des LEGO MINDSTORMS®: • Blockprogrammierung zu Sensoren & Motoren	
Schülerarbeit I 30 Min.	Die Schüler*innen bauen zunächst ein Auto mit LEGO MIND- STORMS®. Hierzu dient die Bauanleitung für das Modell Chassis:	
	LEGO MINDSTORMS® Modell Chassis (Illustration aus	
	der Software LEGO MINDSTORM® education EV3)	
	https://le-www-live-s.legocdn.com/sc/media/lessons/	
	mindstorms-ev3/building-instructions/ev3-rem-driving-base-79 bebfc16bd491186ea9c9o69842155e.pdf	
Schülerarbeit II 30 Min.	Die Schüler*innen erschließen sich selbst durch Anwendung und Modifizierung eines vorhandenen Programms, wie die verschiedenen Blöcke (Aktions-Blöcken, Sensor-Blöcken und Programmablauf-Blöcken) funktionieren.	
Zusammenfassung 20 Min.	Einige Schüler*innen demonstrieren modifizierten Programme und die resultierenden Bewegungen des LEGO MIND-	

STORMS®. Die Ergebnisse werden diskutiert.

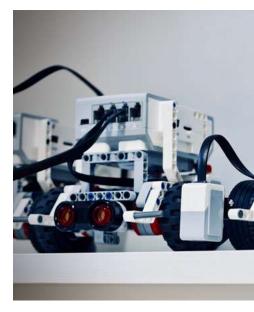


Abb. 36: LEGO MIND-STORMS®-Roboter bieten vielfältig Möglichkeiten in Konstruktion und Programmierung

Fig. 36: LEGO MINDSTORMS® robotter tilbyder en bred vifte af design- og programmeringsmuligheder

#### b) LEGO MINDSTORMS®

Eksemplarisk time for at lære LEGO MINDSTORMS® at kende (90 min.)

Fase	Introduktion til LEGO MINDSTORMS®
Introduktion til dagens emne 10 min.	Klassen introduceres til LEGO MINDSTORMS®. https://education.lego.com/de-de/support/mindstorms-ev3/ getting-started
	Læreren præsenterer kort mulighederne med LEGO MINDSTORMS®:  • Blokprogrammering til sensorer & motorer.
Elevopgave I 30 min.	Eleverne starter med at bygge en bil med LEGO MINDSTORMS®. Man anvender i den forbindelse byggevej- ledningen til model Chassis:



LEGO MINDSTORMS® model Chassis (illustration fra software LEGO MINDSTORM® education EV3).

https://le-www-live-s.legocdn.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-rem-driving-base-79bebfc16bd491186ea9c9069842155e.pdf

bebfc16bd491186eagcg069842155e.pdf

Elevopgave II Gennem anvendelse og modificering af et eksisterende program gør eleverne sig selv fortrolige med, hvordan de forskellige blokke fungerer (hændelsesblokke, sensorblokke og programkørselsblokke).

Resumé Nogle elever demonstrerer modificerede programmer og de

Resumé Nogle elever demonstrerer modificerede programmer og de 20 min. resulterende bevægelser, som LEGO MINDSTORMS® udfører. Resultaterne diskuteres.

Anschließend sind folgende Aufgaben denkbar, die als kleine Projekte durchgeführt werden können:

#### **Autonomes Fahren mit LEGO MINDSTORMS®**

#### Aufgabe:

Programmiert das Auto so, dass es in einem vorgegebenen Bereich selbstständig fährt und dabei kein Hindernis an-/umfährt. Hier könnt ihr mehr über selbstfahrende Autos erfahren: https://de.wikipedia.org/wiki/Selbstfahrendes\_Kraftfahrzeug

#### Untergliederung in Teilaufgaben

#### Teilaufgabe 1:

Programmiert das Auto so, dass es auf einer geraden Strecke einen Meter vor- und rückwärts zurückfahren kann.

#### Folgende Impulse können zur Unterstützung herangezogen werden:

- 1. Welchen Block braucht man hier? Probiert verschiedene Einstellungen aus.
- 2. Tipp: Damit das Auto geradeaus fahren kann, muss der Strom im rechten und linken Motor gleichwertig sein.

#### Teilaufgabe 2:

Programmiert das Auto so, dass es mit einer vorgegebenen Gradzahl abbiegen kann.

#### Folgende Impulse können zur Unterstützung herangezogen werden:

- 3. Welchen Block braucht man hier? Probiert verschiedene Einstellungen aus.
- 4. Der Block kann auf Sekunden, Grad oder Umdrehungen eingestellt werden. Experimentiert damit, wie ihr den Roboter dazu bekommt, um 90 Grad in Sekunden, Graden und Umdrehungen abzubiegen. Probiert auch den Roboter so zu programmieren, dass er um 120 Grad abbiegt. Besteht ein Zusammenhang zwischen Sekunden, Grad und der Umdrehung?

Meter	Sekunden	Grade	Umdrehungen

Efterfølgende kunne man forestille sig følgende opgaver, der kan gennemføres i form af små projekter:

#### Selvkørselsfunktion med LEGO MINDSTORMS®

#### Opgave:

Bilen programmeres på en sådan måde, at den indenfor et forudbestemt areal kører selvstændigt og ikke kører ind i eller vælter nogen forhindringer undervejs. Mere om emneområdet selvkørende biler kan findes her: https://de.wikipedia.org/wiki/Selbstfahrendes\_Kraftfahrzeug

#### Opdeling i delopgaver

#### Delopgave 1:

Programmér bilen på en sådan måde, at den på en lige strækning kan køre én meter frem og tilbage.

#### Følgende impulser kan inddrages som støtte:

- 1. Hvilken blok skal man bruge her? Prøv jer frem med forskellige indstillinger.
- Et tip: For at bilen kan køre ligeud, skal strømmen i højre og venstre motor have samme værdi.

#### Delopgave 2:

Programmér bilen på en sådan måde, at den kan svinge (køre om et hjørne) i en forudbestemt vinkel (grader).

#### Følgende impulser kan inddrages som støtte:

- 3. Hvilken blok skal man bruge her? Prøv jer frem med forskellige indstillinger.
- 4. Blokken kan indstilles til sekunder, grader eller omdrejninger. Prøv jer frem med, hvordan i får robotten til at svinge om hjørnet i en 90 graders vinkel i sekunder, grader og omdrejninger. Prøv også at programmere robotten til at udføre en 120 graders svingning. Er der en sammenhæng mellem sekunder, grader og omdrejninger her?

Meter	Sekunden	Grade	Umdrehungen

#### Teilaufgabe 3:

Programmiert das Auto so, dass es durch Verwendung von Loops/Schleifen anstatt wiederholten Programmblöcken beispielsweise in einem Quadrat fahren kann.

#### Folgende Impulse können zur Unterstützung herangezogen werden:

- 5. Welchen Block braucht man hier? Probiert verschiedene Einstellungen aus.
- 6. Kombiniert eure Ergebnisse aus den Teilaufgaben 1-3.

#### Teilaufgabe 4:

Programmiert das Auto unter Verwendung des Sensors zur Abstandsmessung so, dass es bremst bevor es gegen einen Gegenstand fährt.

#### Folgende Impulse können zur Unterstützung herangezogen werden:

- 7. Welche Blöcke braucht man hier? Probiert verschiedene Einstellungen aus.
- 8. In welcher Situation soll das Auto vorwärts, in welcher nach einem Signalton rückwärts fahren?

#### Teilaufgabe 5:

Programmiert das Auto mithilfe des Lichtsensors so, dass es frei auf einem großen Stück bunten Papiers fahren kann, nicht außerhalb des Bereichs fahren darf, aber rückwärtsfahren/abbiegen soll, wenn es auf eine Kante trifft, und anschließend weiterfährt.

#### Folgende Impulse können zur Unterstützung herangezogen werden:

9. Welche Blöcke braucht man hier? Probiert verschiedene Einstellungen aus.

#### Mögliche weitere Aufgaben für Fortgeschrittene oder zur Differenzierung:

#### Aufgabe X:

Das Auto muss einen Strich auf der Straße entlangfahren können. (Tipp: Ausrichtung des Farbsensors)

#### Aufgabe Y:

Das Auto muss an einer roten Ampel anhalten und an einer grünen Ampel weiterfahren können. (Tipp: Ausrichtung des Farbsensors)

#### Aufgabe Z:

Das Auto muss gut in eine Parklücke einparken können, ohne dabei etwas anzufahren. (Tipp: Bau einer Garage z. B. mit drei senkrecht stehenden Büchern etc., die insgesamt etwa doppelt so breit ist wie das Auto)

#### Delopgave 3:

Programmér bilen på en sådan måde, at den v. h. a. loops / sløjfer f. eks. kan køre i et kvadrat i stedet for gentagne programblokke.

#### Følgende impulser kan inddrages som støtte:

- 5. Hvilken blok skal man bruge her? Prøv jer frem med forskellige indstillinger.
- 6. Kombinér jeres resultater fra delopgaverne 1-3.

#### Delopgave 4:

Programmér v. h. a. afstandsmålingssensoren bilen på en sådan måde, at den bremser, inden den kører ind i en forhindring.

#### Følgende impulser kan inddrages som støtte:

- 7. Hvilke blokke skal man bruge her? Prøv jer frem med forskellige indstillinger.
- 8. I hvilken situation skal bilen køre fremad og i hvilken baglæns efter et lydsignal?

#### Delopgave 5:

Programmér v. h. a. lyssensoren bilen på en sådan måde, at den kan køre frit på et stort stykke kulørt papir, ikke må køre udenfor arealet, men skal bakke resp. svinge, når den møder en kant, og efterfølgende fortsætte sin kørsel.

#### Følgende impulser kan inddrages som støtte:

9. Hvilke blokke skal man bruge her? Prøv jer frem med forskellige indstillinger

#### Mulige yderligere opgaver for viderekomne eller m. h. p. differentiering:

#### Opgave X:

Bilen skal kunne køre langs med en streg på kørebanen. (Et tip: Farvesensorens orientering/retning.)

#### Opgave Y:

Bilen skal kunne standse foran rødt lys og fortsætte når/hvis der er grønt. (Et tip: Farvesensorens orientering/retning.)

#### Opgave Z:

Bilen skal kunne køre korrekt ind og parkere på en ledig plads uden at kollidere med noget. (Et tip: Byg en "garage" v. h. a. tre lodret stående bøger el. lign., som er ca. dobbelt så bred som bilen.)

#### Mögliche Lösungen:

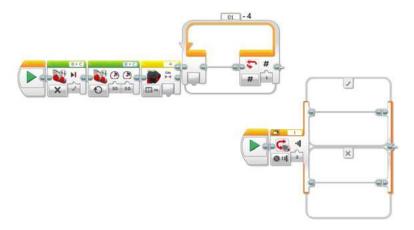
Das Auto muss einer Linie auf der Straße folgen können.

Verwendbare Blöcke: Standardsteuerung, Farbsensorsteuerung, Schleife



Das Auto muss anhalten können, wenn sich etwas auf der Straße befindet.

Verwendbare Blöcke: Standardsteuerung, Hebelsteuerung, Ultraschallsteuerung und akustische Signale, Farbsensor, Schleife, Schalter



Das Auto muss an einer roten Ampel anhalten und an einer grünen Ampel weiterfahren können. Verwendbare Blöcke: s. o.



Das Auto muss gut in eine Parklücke einparken können, ohne dabei etwas anzufahren. Verwendbare Blöcke: s. o.



#### Mulige løsninger:

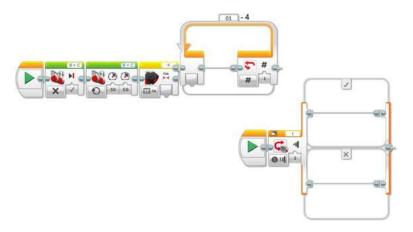
Bilen skal kunne køre langs med en streg på kørebanen.

Brugbare blokke: Tankstyring, bevægelsesstyring, sløjfe.



Bilen skal kunne standse op, hvis der er en forhindring på vejen – naturligvis uden at køre nogen ned.

Brugbare blokke: Tankstyring, tid (ultralydsensor) og akustiske signaler, bevægelsesstyring, tid (farvesensor) og sløjfe.



Bilen skal kunne standse for rødt lys og køre videre, når/hvis der er grønt. Brugbare blokke: Se ovenfor.



Bilen skal kunne køre korrekt ind og parkere på en ledig plads uden at kollidere med noget. Brugbare blokke: Se ovenfor.



#### Pizza-Lieferservice mit LEGO MINDSTORMS®

#### Aufgabe 1:

Aus einem Zeitungsbericht: "Wer bei Domino's in Hamburg eine Pizza bestellt, dem kann es bald passieren, dass wenig später ein Roboter vor die Haustür rollt – knapp 40 Zentimeter hoch, mit sechs Rädern und einer Antenne mit blauem Fähnchen. [...] Der Pizza-Lieferdienst Domino's hat jetzt einen Kooperationsvertrag mit dem Roboterhersteller Starship Technologies geschlossen. Testweise sollen künftig Starship-Roboter Bestellungen im Umkreis von zwei Kilometern der Domino's-Filialen in ausgewählten Städten in Deutschland und den Niederlanden ausliefern. In Deutschland startet die Testphase mit Roboterlieferungen in Hamburg." (Artikel im Handelsblatt vom 29.03.2017:

https://www.handelsblatt.com/unternehmen/handel-konsumgueter/dominos-in-hamburg-roboter-springen-fuer-fehlende-fahrer-ein/19584636-2.html)

Simuliert mit dem LEGO MINDSTORMS®-Roboter eine Pizza-Lieferung. Baut dazu eine Strecke, die die Strecke von der Pizzeria zum Ort der Auslieferung simuliert. Unterwegs müssen sich Hindernisse auf der Strecke befinden. Erstellt eine Liste mit den Dingen, die euer "Pizzaroboter" können soll. Programmiert dann euren Roboter so, dass er den Anforderungen eurer Liste entspricht.

#### Folgende Impulse können zur Unterstützung herangezogen werden:

- 1. Wie müsste die Programmierung aussehen, wenn der Roboter bei der Pizzeria losfährt und die Pizza am Lieferort ausliefert, während er unterwegs den Hindernissen ausweicht?
- 2. Welche Teile habt ihr dazu schon in der Aufgabe zum autonom fahrenden Auto genutzt?

#### Pizzaudbringning med LEGO MINDSTORMS®

#### Opgave 1:

Fra en avishistorie: "Hvis man bestiller en pizza hos Domino's i Hamborg, kan det snart ske, at en robot ruller hen til ens dør lidt senere – knap 40 centimeter høj, med seks hjul og en antenne med små blå flag. [...] Pizzaleveringstjenesten Domino's har for nylig underskrevet en samarbejdsaftale med robotproducenten Starship Technologies. I en testperiode skal det fremover være Starship-robotter, der i udvalgte byer i Tyskland og Holland skal klare leveringen fra Domino's-filialer, hvis disse ligger mindre end to kilometer væk fra kunden. I Tyskland starter testperioden med robotudbringning i byen Hamborg." (Indlæg i Handelsblatt af 29.03.2017

https://www.handelsblatt.com/unternehmen/handel-konsumgueter/dominos-in-hamburg-roboter-springen-fuer-fehlende-fahrer-ein/19584636-2.html)

Lav v. h. a. LEGO MINDSTORMS®-robotten en simulation af en pizzaleverance. Byg hertil et forløb, der simulerer strækningen fra pizzeriaet til stedet, hvor der skal leveres. Der skal være nogen forhindringer undervejs på strækningen. Opstil en liste med, hvad jeres "pizzarobot" skal kunne. Programmér dernæst robotten, så den kan håndtere kravene fra denne liste.

#### Følgende impulser kan inddrages som støtte:

- 1. Hvordan vil programmeringen skulle være, hvis robotten starter sin tur ved pizzeriaet og leverer pizzaen på leveringsstedet og i den forbindelse sørger for at køre udenom forhindringer undervejs?
- 2. Hvilke dele vedr. dette har I allerede stiftet bekendskab med og anvendt i opgaven med den selvkørende bil?

#### c) Verschlüsselungstheorien

## Aktivierung von Vorwissen der Lernenden zum Thema Verschlüsselungstheorie

Vorschlag für eine Einstiegsstunde (90 Min)

Phase	Einführung in das Thema Verschlüsselung
Einführung in das Thema 15 Min.	Die Lehrkraft präsentiert das Thema Verschlüsselung über Fragen oder Bilder mit der Zielfrage: Wo muss man im Alltag sichere Nachrichten verschicken können bzw. sicher Informa- tionen übermitteln können?
Erarbeitung I 10 – 15 Min.	Die Schüler*innen erarbeiten in Parnerarbeit auf Basis ihres Vorwissens und Alltagsvorstellungen eine eigene Definition für den Begriff Verschlüsselung.
	Die Ergebnisse werden z.B. mithilfe von Padlet an einem Whiteboard festgehalten, alternativ könnten auch Plakate erstellt werden.
Sicherung I 5 Min.	In der Lerngruppe sollte der Definitionsbegriff wieder- holt bzw. neu eingeführt werden (https://www.duden.de/ rechtschreibung/Definition; ggf. mehr Zeit einplanen)
Erarbeitung II 15 Min.	Anhand des Definitionsbegriffs werden die von den Schüler*innen erarbeiteten Definitionen auf ihre Eignung hin untersucht und in einer Tabelle Unterschiede sowie Gemeinsamkeiten festgehalten (z. B Worauf bezieht sich die Verschlüsselung?, Was wurde nicht abgedeckt?). Dieses Festhalten erfolgt entwerder als Galeriegang oder in digitaler Form.
Sicherung II 10 – 15 Min.	Mit Hilfe der Tabellen wird versucht, in der Lerngruppe eine einheitliche Definition zu finden. <i>Die Lehrkraft wirkt als Moderator.</i>
	Die Definition wird mit folgender Definition (https://de. wikipedia.org/wiki/Verschl%C3%BCsselung) verglichen. Die Schüler*innen benennen Unterschiede.
Vertiefung & Transfer 15 Min.	Anhand eines Beispieles werden die Begriffe "Klartext", "Chiffre" und "Schlüssel" erklärt. Dies kann durch "Material 1" visuell unterstützt werden. Diese Symboliken der Abbildungen können im weiteren Verlauf einer Einheit zur Kryptographie als Anker genutzt werden.
	Als Rückbezug zur Zielfrage wird im Plenum diskutiert, war- um Verschlüsselung wichtig ist.



Abb. 37: Die Verschlüsselung von Daten spielt in der digitalen Kommunikation eine große Rolle

Fig. 37: Kryptering af data spiller en vigtig rolle i digital kommunikation



#### c) Krypteringsteorier

#### Aktivering af eksisterende viden om emnet krypteringsteori

Forslag til en introduktionstime (90 min.)

Fase	Introduktion til emnet kryptering
Introduktion til dagens emne 15 min.	Læreren lægger op til emnet kryptering v. h. a. spørgsmål eller billeder og stiller målspørgsmålet: Hvor i hverdagen er det nødvendigt at kunne sende beskeder, meddelelser og infor- mation på en sikker måde?
Udarbejdelse I 10 – 15 min.	Eleverne går med udgangspunkt i deres eksisterende viden og kendte egne hverdagsforetéelser sammen om at udarbej- de deres egen definition af begrebet kryptering.
	Resultaterne vises v. h. a. Padlet på et whiteboard – alternativt kan der også laves plakater.
Sikring I 5 min.	I en læringsgruppe skal definitionsbegrebet gentages resp. nyintroduceres – (https://www.duden.de/rechtschreibung/ Definition; evt. må der gives mere tid til dette).
Udarbejdelse II 15 min.	Med udgangspunkt i definitionsbegrebet undersøges egnetheden af de af eleverne udarbejdede definitioner, og i en tabel noteres forskelle og ligheder (f. eks. Hvad relaterer/refererer krypteringen til?, Hvad dækker den ikke?). Resultaterne præsenteres enten af eleverne på omgang (ty.: gallerimetode) eller i digital form.
Sikring II 10 - 15 min.	V. h. a. tabellerne forsøger læringsgruppen at finde frem til en samlet definition. Læreren fungerer her som moderator.  Definitionen sammenlignes med denne her: (https://de. wikipedia.org/wiki/Verschl%C3%BCsselung). Eleverne identificerer forskelle.
Fordybelse & over- førsel 15 min.	Med udgangspunkt i et eksempel redegøres der for termerne "klartekst", "chiffre" og "nøgle". Dette kan supporteres visuelt v. h. a. "Materiale 1". Disse symboler i figurerne kan under forløbet bruges som 'anker' i forbindelse med kryptografien.
	For at komme tilbage til det oprindelige målspørgsmål diskuteres i plenum, hvorfor kryptering er vigtig.

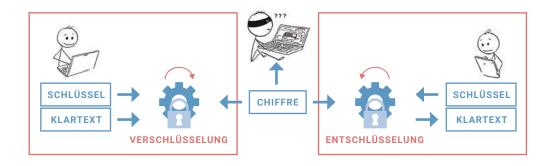
#### Exemplarische Stunde zur Begegnung mit der Cäsar-Verschlüsselung

Phase	Einführung in die Cäsar-Verschlüsselung
Einstieg 10 Min.	L. zeigt Bild der historischen Cäsar-Scheibe (Abb. 26). Schüler*innen stellen Vermutungen an, welchen Zweck dieses Objekt hat, wie man es benutzen könnte, wo es herkommt, etc
	L. erklärt, dass es sich um ein Gerät zur Verschlüsselung handelt. Ursprünglicher Zweck der Verschlüsselung sowie Namensgebung werden erläutert.  L. erklärt, dass es sich bei der Scheibe um eine Weiterentwicklung handelt, Ausgangspunkt war das lateinische Alphabet.
Erarbeitung I 10 Min.	Schüler*innen erarbeiten sich die "Abbildungsvorschrift" mit "Material 2" und einem Beispiel, indem dieser Schlüssel (3) verwendet wird.
Sicherung I 10 Min.	L. vergleicht erarbeitete Abbildungsvorschrift und stellt sicher, dass in dieser Verschlüsselung der Schlüssel richtig erkannt wird (Stellen, um die verschoben wird). L. erarbeitet gemeinsam mit den Schüler*innen an einem Beispiel, wie die Verschlüsselung rückgängig gemacht werden kann (also die Entschlüsselung).
Erarbeitung II 30 Min.	Schüler*innen üben in Partnerarbeit das vorgestellte Verschlüsselungsverfahren, indem:
	<ul> <li>a) aus Klartext und Schlüssel (beide vorgegeben) die Chiffre erstellt wird. Klartextbuchstaben und Schlüssel müssen so gewählt werden, dass "über das 'z' hinaus" geschoben wird.</li> <li>L. bespricht im Plenum, wie das Problem gelöst werden kann und löst das</li> </ul>
	Problem durch Vorstellen der Buchstabenuhr ("Material 3"). b) aus Chiffre und Schlüssel (beide vorgegeben) der Klartext erstellt wird;
	c) aus Nachricht und Chiffre (beide vorgegeben) der Schlüssel abgeleitet wird.
	d) Schüler*innen erstellen eigene Aufgaben und erarbeiten Lösungen, ähnlich zu a) – c). Zur Hilfe kann "Material 4" verwendet werden.
Sicherung II 10 Min.	Beispiele werden verglichen. L. fragt nach aufgetretenen Problemen bei den selbsterstellten Aufgaben (Verweis auf Schlüsseltausch).
Vertiefung 20 Min.	Mit Hilfe einer Beispielchiffre fordert L. die Schüler*innen auf, den Klartext zu ermitteln, ohne den Schlüssel mitzuteilen. Das heißt, die Schüler*innen müssen die Verschlüsselung "knacken". Anschließende Diskusion im Plenum über den gefundenen Schlüssel und das Vorgehen. Schüler*innen beurteilen die Sicherheit des Verfahrens.

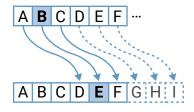
#### Eksemplarisk time for at lære Cæsar-krypteringen at kende

Fase	Introduktion til Cæsar-krypteringen
Introduktion 10 min.	Læreren viser et billede af den historiske Cæsar-skive (fig. 26). Eleverne giver bud på, hvad dén mon er til, hvad man evt. kunne bruge den til, hvor den kom- mer fra osv
	L. forklarer, at den bruges til kryptering – og redegør for det oprindelige for- mål med krypteringen og hvor skiven har sit navn fra. L. forklarer, at skiven er en videreudvikling og at udgangspunktet var det latinske alfabet.
Udarbejdelse I 10 Min.	Eleverne tilegner sig "visningsreglen" v. h. a. "Materiale 2" samt et eksempel, hvor den pågældende nøgle (3) anvendes.
Sikring I 10 min.	L. sammenligner den udarbejdede visningsregel og sikrer, at eleverne for denne krypterings vedkommende har afdækket nøglen korrekt (hvor mange positioner der skal forskydes med). L. finder sammen med eleverne frem til et eksempel, hvordan krypteringen kan ophæves igen (= dekryptering).
Udarbejdelse II 30 min.	Eleverne går sammen om at (ind)øve den viste krypteringsmetode ved at de:
	a) producerer chiffret v. h. a. klartekst og nøgle (begge fastlagt på forhånd). Klartekstbogstaver og nøgle skal vælges på en sådan måde, at der forskydes "ud over bogstavet 'z' ". L. drøfter i plenum, hvordan problemet kan løses – og løser det ved at præ-
	sentere bogstavuret ("Materiale 3"); b) producerer klarteksten v. h. a. chiffre og nøgle (begge fastlagt på forhånd);
	c) udleder/konkluderer nøglen v. h. a. beskedtekst og chiffre (begge fastlagt på forhånd).
	d) Eleverne producerer egne opgaver og udarbejder løsninger, i lighed med a) – c). "Materiale 4" kan bruges som hjælp her.
Sikring II 10 Min.	Eksemplerne sammenlignes. L. spørger, om der har været problemer i forbindelse med de selvproducerede opgaver (med henvisning til skift af nøgle).
Fordybelse 20 min.	V. h. a. et eksemplarisk chiffre opforderer L. eleverne til at afdække klarteksten uden at afsløre nøglen – d. v. s. eleverne er nødt til at "knække" krypteringen. Efterfølgende debat i plenum om den fundne nøgle og fremgangsmåden. Eleverne vurderer metodens sikkerhed/duelighed.

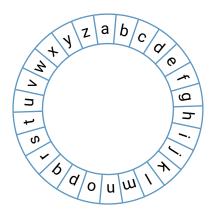
#### Material 1: Einführung Begrifflichkeiten zur Verschlüsselung



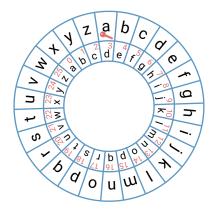
Material 2: Verschiebung bei Cäsar-Verschlüsselung

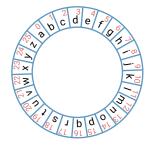


Material 3: Buchstabenuhr

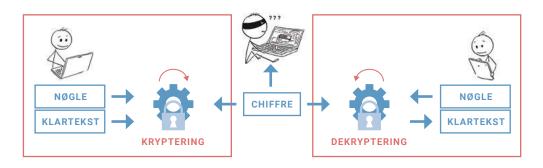


Material 4: Cäsar-Scheibe

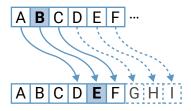




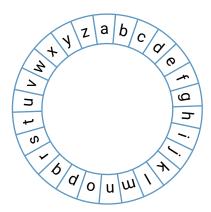
#### Materiale 1: Introduktion til terminologien i forbindelse med emnet kryptering



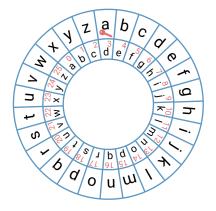
Materiale 2: Forskydning i forbindelse med Cæsar-krypteringen

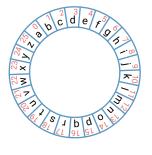


Materiale 3: Bogstavuret



Materiale 4: Cæsar-skive





#### Aufgabe: Verbesserung des Cäsar-Verfahrens:

Beim Cäsar-Verfahren wird die Nachricht verschlüsselt, indem jeder Buchstabe um den gleichen Schlüsselwert weitergerückt wird. Wie könnte ein Verschlüsselungsverfahren aussehen,

- a) bei dem man sich nicht an eine so feste Verschlüsselungsregel halten muss?
- b) bei dem der Schlüssel ein Wort ist?

#### Folgende Impulse können zur Unterstützung herangezogen werden:

- Was ist das Problem der Cäsar-Verschlüsselung?
- Wie kann man dieses Problem lösen?

#### Mögliche Antworten:

Problem: Bei der Cäsar-Verschlüsselung reicht es, 26 Schlüssel auszuprobieren, wodurch der Klartext geliefert wird; eine begrenzte Anzahl an Schlüsseln.

Problemlösung: Man bildet eine Tabelle mit zwei Zeilen. In der ersten befindet sich das normale Alphabet, diese Zeile steht für die Klartext Buchstaben. Die zweite Zeile steht für die Chiffrebuchstaben. Man füllt die Tabelle so, dass sich unter jedem Klartextbuchstabe der passende Chiffrebuchstabe befindet (bspw. mit Schlüssel 3 dann mit d beginnend). Dies erfolgt ähnlich zur Cäsar-Scheibe, allerdings als Strahl / Tabelle. In der zweiten Zeile fällt auf, dass wieder das geordnete Alphabet entsteht, nur mit dem entsprechende Schlüssel verschoben.

- Es gibt eine "1 zu 1"-Zuordnung von Klartext und Chiffre-Buchstaben (s. https://de. wikipedia.org/wiki/Monoalphabetische\_Substitution)
- Anzahl der Schlüssel wird stark erhöht (sicherer gegenüber brute force-Angriff), ist aber immer noch anfällig für andere Angriffe (https://de.wikipedia.org/wiki/H%C3%A4 ufigkeitsanalyse & https://de.wikipedia.org/wiki/Monoalphabetische\_Substitution #H%C3%A4ufigkeitsanalyse)
- Der Wert, um den verschoben wird, hängt davon ab, an welcher Stelle der Klartextbuchstabe steht (https://de.wikipedia.org/wiki/Polyalphabetische\_Substitution & https:// de.wikipedia.org/wiki/Vigen%C3%A8re-Chiffre)

#### Opgave: Forbedring af Cæsar-metoden:

Med Cæsar-metoden krypteres en besked ved, at hvert bogstav forskydes det samme antal positioner (fast nøgleværdi). Hvordan kunne en krypteringsmetode være, hvor

- c) man ikke behøvede at overholde en så fastlagt krypteringsregel?
- d) nøglen eksisterede i form af et ord?

#### Følgende impulser kan inddrages som støtte:

- · Hvad er problemet med Cæsar-krypteringen?
- · Hvordan kan dette problem løses?

#### Mulige svar:

Problem: Med Cæsar-krypteringen kan man "nøjes med" at gennemprøve 26 nøgler for at finde frem til klarteksten; antallet af nøgler er således begrænset.

Problemløsning: Man laver en tabel bestående af to linjer. Den første indeholder det almindelige alfabet og er den, hvor klartekstbogstaverne står. Den anden linje er chiffrebogstaverne. Tabellen udfyldes nu på en sådan måde, at der under hvert klartekstbogstav findes det passende chiffrebogstav (f. eks. med nøgle 3, som så starter med et d). Dette foregår på en lignende måde som med Cæsar-skiven, dog således, at der er tale om en slags "stråle", en tabel. I anden linje kan det ses, at det regulære alfabet genopstår, blot forskudt med den pågældende nøgle.

- Der er tale om en "1 til 1"-tilknytning mellem klartekst og chiffrebogstaver (se https://de.wikipedia.org/wiki/Monoalphabetische\_Substitution).
- Antallet af nøgler øges kraftigt (mere sikkerhed mod såk. brute-force-angreb), men er stadig sårbart overfor andre angreb: (https://de.wikipedia.org/wiki/H%C3%A4 ufigkeitsanalyse & https://de.wikipedia.org/wiki/Monoalphabetische\_Substitution #H%C3%A4ufigkeitsanalyse).
- Værdien, som der forskydes med, hænger på det sted, hvor klartekstbogstavet står: (https://de.wikipedia.org/wiki/Polyalphabetische\_Substitution & https://de.wikipedia.org/wiki/Vigen%C3%A8re-Chiffre).

# VII. Arduinos im Robo-Camp



# VII. Arduinos i Robo Campen



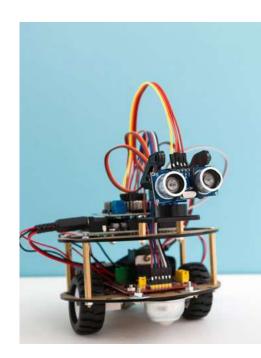
#### Ausführliche Beschreibung eines Ablaufs zum Einsatz von Arduinos

Bjarke Kristian Maigaard Kjær Pedersen / SDU

Während sich die Ausführungen in Kapitel 6 dieses Bandes an erfahrene Lehrkräfte richten, die einen offenen und projektorientierten Unterrichtsansatz wählen, findet der Leser bzw. die Leserin in diesem Kapitel eine sehr detaillierte Anleitung, wie man Arduinos einsetzen kann, worauf man achten sollte und welches inhaltliche Wissen vermittelt werden kann. Diese Anregungen sind am Anfang absichtlich kleinschrittig gestaltetet, damit auch mit der Thematik unerfahrene Personen sich vorstellen können, eine Anleitung dazu zu geben. Gegen Ende des Kapitels ist die Arbeit an dem Projekt hingegen deutlich weniger anleitend, weil zu dem Zeitpunkt alle Einzelaspekte bereits thematisiert worden sind und zusammenhängend angewendet werden sollen. Die Anregungen selbst können natürlich immer noch variabel eingesetzt werden, je nach Vorwissen und Interesse der Lernenden. Alle Aufgaben in diesem Kapitel sind an Lehrende gerichtet, die diese als Impulse als Lernende weitergeben können. Der vorliegende Ablauf ist in mehrere Schritte und Phasen gegliedert, von denen der bzw. die Lehrende frei wählen kann, ob alles aufgegriffen werden soll oder nur die Teile Sinn machen, die am besten zum aktuellen Lernniveau der Lernenden passen. Ebenso ist es natürlich möglich, auf Grundlage des bereitgestellten Materials eigene Sequenzen zu erstellen. Das vorliegende Material eignet sich ggf. auch dazu, Lernenden außerhalb des Unterrichts (als Ferienbeschäftigung oder im HomeSchooling) Freude am Programmieren zu vermitteln.

Das Hauptthema des vorliegenden Materials sind Roboter und Automatisierung. Der Schwerpunkt liegt darauf, den Lernenden ein Verständnis dazu näherzubringen, was ein Roboter ist und was die Gesellschaft manchmal in das Gerät hineininterpretiert. Gleichzeitig sollen Grundkenntnisse in Robotik vermittelt werden, darunter ein Verständnis von Mikrocontrollern, Sensoren und Aktoren. Ziel ist es, die Lernenden mit Fähigkeiten und Wissen auszustatten, damit sie mit dem Arduino-System ein Technologieverständnis entwickeln, um anschließend ihre eigenen Roboter entwickeln und konkrete Probleme lösen zu können.

Das Material wurde im Zusammenhang mit dem Lehr- und Lernangebot des PANaMA-Projekts in Form von Forschungscamps, wie sie u. a. im Herbst 2019 an der Kieler Forschungswerkstatt stattgefunden haben (vgl. Kapitel IV), entwickelt und umgesetzt. Das Camp war nicht in klassische Unterrichtsmodule unterteilt, und die Zeitspannen für die



#### Detaljeret beskrivelse af en procedure til anvendelse af Arduinos

Bjarke Kristian Maigaard Kjær Pedersen / SDU

Mens forklaringerne i kapitel 6 i dette bind er rettet mod erfarne lærere, der vælger en åben og projektorienteret undervisningsmetode, vil læseren i dette kapitel finde meget detaljerede instruktioner om, hvordan man bruger Arduinos, hvad man skal passe på og hvilket indhold viden kan formidles. Disse forslag er bevidst designet i små trin i begyndelsen, så selv folk uerfarne med emnet kan forestille sig at give instruktioner. Mot slutningen af kapitlet er arbejdet med projektet på den anden side langt mindre lærerigt, fordi alle individuelle aspekter på det tidspunkt allerede er tematiseret og bør anvendes sammenhængende. Forslagene i sig selv kan naturligvis stadig bruges varierende, afhængigt af elevernes forudgående viden og interesser. Alle opgaverne i dette kapitel er rettet mod lærere, der kan videregive disse som impulser som elever. Den nuværende proces er opdelt i flere trin og faser, hvorfra læreren frit kan vælge, om alt skal optages, eller kun de dele, der giver mening, der bedst passer til det aktuelle læringsniveau for læreren. Det er naturligvis også muligt at oprette dine egne sekvenser baseret på det leverede materiale. Det aktuelle materiale er også velegnet til at lære eleverne at nyde programmering uden for klassen (som en ferieaktivitet eller i hjemmeskolegang).

Hovedtemaet for dette materiale er robotter og automatisering. Fokus er på at give eleverne en forståelse af, hvad en robot er, og hvad samfundet undertiden tolker i enheden. Samtidig skal grundlæggende viden om robotik overføres, herunder forståelse af mikrokontrollere, sensorer og aktuatorer. Målet er at udstyre eleverne med færdigheder og viden, så de kan udvikle en forståelse af teknologi med Arduino-systemet, så de derefter kan udvikle deres egne robotter og løse specifikke problemer.

Materialet blev udviklet i forbindelse med undervisning og læring af PANaMA-projektet i form af forskningslejre, f.eks fandt sted i efteråret 2019 på Kiel Forskningsværksted (se kapitel IV), udviklet og implementeret. Lejren blev ikke opdelt i klassiske undervisningsmoduler, og tidsspændene for de enkelte faser er derfor ikke skarpt opdelt i moduler på 45 minutter hver. Det materiale, hvoraf dele heraf vil blive præsenteret i nærværende lektionsplanen, kan i sin helhed findes på følgende link (bemærk at materialet for nuværende er på engelsk): http://www.teknologiskolen.dk/robocamp2019/

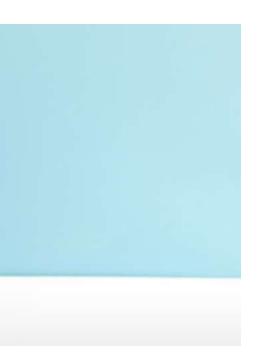


Abb. 38: Die Arduino-Plattform bietet unzählige Möglichkeiten zur Gestaltung und zum Bau von Robotern

Fig. 38: Arduino-platformen giver uendelige muligheder for at designe og bygge robotter

einzelnen Phasen sind daher nicht scharf in Module zu je 45 Minuten unterteilt. Das vollständige entwickelte Material, von dem Teile in diesem Kapitel vorgestellt werden, finden Sie unter dem folgenden Link (bitte beachten Sie, dass das Material derzeit nur in englischer Sprache vorliegt): http://www.teknologiskolen.dk/robocamp2019/.

**Anmerkung:** Das Material basiert auf dem Arduino-Uno-Mikrocontroller sowie den Sensoren und Aktoren, die im Forschungscamp eingesetzt wurden. Dies bedeutet, dass die angegebenen Einheiten ebenfalls hierauf basieren, und ein digitales Signal daher in diesem Material beispielsweise als zwischen o V (LOW) und 5 V (HIGH) liegend beschrieben wird. Die konkreten Werte der Spannung sind dabei plattformabhängig; so arbeiten beispielsweise sowohl micro: bit als auch Calliope Mini mit einer Spannung zwischen o V (LOW) und 3,3 V (HIGH) bei Anschluss über ein USB-Kabel; bei Nutzung einer 3-V-Batterie als Spannungsquelle sind es o V (LOW) und 3 V (HIGH).

# EINFÜHRUNG IN DAS THEMA ROBOTER – WAS IST EIN ROBOTER? (70 MIN.)

**Kurz zum Inhalt:** Der Unterricht erfolgt an dieser Stelle überwiegend über Input der bzw. des Lehrenden, wobei der Dialog mit den Lernenden zu den vorgestellten Aspekten im Vordergrund steht. Der Schwerpunkt liegt auf einem gemeinsamen Verständnis dessen, was ein Roboter ist bzw. nicht ist und aus welchen Kernkomponenten er besteht sowie dem Zusammenspiel der Komponenten und Beziehungen zueinander.

Lernergebnisse: Die Lernenden nennen eine mögliche Definition eines Roboters. Sie können identifizieren und analysieren, welche Objekte als Roboter klassifiziert werden können und warum. Darüber hinaus erhalten sie Einblick in die internen Kernkomponenten eines Roboters, welche Rolle sie im Roboter spielen und welche Interaktion zwischen ihnen besteht. Infolgedessen werden zudem mögliche Missverständnisse in Bezug auf Roboter aufgrund der Darstellung in modernen Medien angesprochen, ausgeräumt und ggf. entmystifiziert.

#### Phase 1 – Einführung in das Thema (30 Min.):

Erklären Sie den Lernenden, dass zunächst behandelt werden muss, was ein Roboter ist, bevor sie mit ihren eigenen Robotern arbeiten und sie herstellen können. Beginnen Sie dann mit der Erörterung darüber, was ein Roboter ist, indem Sie die Lernenden fragen, was sie unter dem Wort Roboter verstehen und was einen solchen ausmacht.

Den Lernenden werden Bilder gezeigt (z. B. humanoide Roboter, nicht humanoide Roboter, elektrische Maschinen und nicht-elektrischen Gerätschaften wie z. B. ein Fahrrad), zu denen die Lernenden für jedes Bild gefragt werden, ob das Objekt auf dem Bild ein Roboter ist oder nicht, und womit die Antwort begründet wird.

**Note:** I lektionerne tages der udgangspunkt i mikrocontrollerteknologien Arduino Uno, samt de sensorer og aktuatorer der indgik i forløbet. Dette betyder at de angivne enheder ligeledes tager udgangspunkt heri, og et digitalt signal vil derfor eks. blive beskrevet som værende mellem oV (LOW) og 5V (HIGH), til trods for at dette er platformsbestemt. F.eks. operer både micro:bit og Calliope Mini, med en spænding på mellem oV (LOW) og 3.3V (HIGH) når disse er forbundet gennem USB, hvorimod hvis de kører på en 3V batteriforsyning, er det følgende der er gældende, oV (LOW) og 3V (HIGH).

# INTRODUKTION TIL ROBOTTER - HVAD ER EN ROBOT? (70 MIN.)

**Kort om indholdet:** På dette tidspunkt er lektioner hovedsageligt baseret på input fra læreren, men med stor vægt på dialog med eleverne. Omdrejningspunktet er at få etableret en fælles forståelse for hvad en robot er/ikke er, samt hvilke kernekomponenter de består af, disses indbyrdes virke og relation til hverandre.

Læringsudbytte: Eleverne bliver bekendt med hvad mulig definition på en robot kan være, de bliver sat i stand til at identificere og analysere hvilke objekter der kan klassificeres som værende en robot, samt hvorfor. Derudover får de indsigt i hvilke interne kernekomponenter en robot indeholder, hvilken rolle de spiller i robotten, samt samspillet mellem dem. I forlængelse heraf, vil potentielle misforståelser vedr. robotter, grundet moderne mediers portrættering af disse, ligeledes blive opklaret, samt evt. afmystificeres.

#### 1. Fase - Den indledende samtale (30 min):

Forklar eleverne at førend man kan arbejde med og lave vores regne robotter, må man først vide hvad en robot er. Indled herefter samtalen om hvad en robot er, ved at spørge klassen indtil hvad de forstår ved ordet robot, samt hvad der udgør en sådanne.

Klassen præsenteres nu for en serie af billeder (eks. af humanoide-robotter, ikke-humanoide robotter, elektriske maskiner, samt ikke elektriske redskaber som evt. en cykel), hvortil eleverne for hvert billede spørges ind til hvorvidt objektet på billedet er en robot eller ikke, samt hvilken begrundelse der lægger til grund for svaret.

Herefter kan den officielle definition eller rettere mangel på samme bringes i spil, der findes nemlig ikke nogen endegyldig definition af, hvad en robot er, om end der er en generel konsensus vedr. en række parametre. En mulig definition herpå kan derfor være følgende: "En programmerbar maskine, hvilken kan fungere autonomt, samt sanse sine omgivelser og tilpasse sine handlinger dertil".

Tal med eleverne om, hvilke aspekter en robot skal udføre for at imødekomme denne definition:

· Sanse verdenen

Danach kann die offizielle Definition bzw. das Fehlen derselben ins Spiel gebracht werden – denn es gibt keine endgültige Definition, was ein Roboter ist, obschon Einvernehmen hinsichtlich einiger diesbezüglicher Parameter besteht. Eine mögliche Definition könnte daher so lauten: "Ein Roboter ist eine programmierbare Maschine, die autonom funktionieren, ihre Umgebung wahrnehmen und ihre Aktionen entsprechend anpassen kann".

Sprechen Sie mit den Lernenden darüber, welche Aspekte ein Roboter leisten muss, um diese Definition zu erfüllen:

- · die Welt wahrnehmen
- · in der Welt agieren
- Entscheidungen treffen, dh. auf Basis des Wahrgenommenen die korrekte Aktion ausführen.

Wenn diese drei Anforderungen erfüllt sind, kann das Gespräch zu einem Diskurs darüber werden, wie Roboter die oben genannten Anforderungen erfüllen können, wobei die Lernenden über folgende Impulse zunächst ihre eigene Sinneswahrnehmung und -interpretation hinterfragen und mit denen des Roboters kontrastieren:

#### Die Welt wahrnehmen:

- Wie nehmen Menschen die Welt wahr? (Sehen, hören, schmecken, riechen, fühlen)
- Wie kann ein Roboter die Welt wahrnehmen? (benötigt mindestens einen Sensor, der z. B. lichtempfindlich ist)

#### In der Welt agieren:

- · Wie agieren Menschen in der Welt? (Muskeln)
- Wie kann ein Roboter agieren? (benötigt mindestens einen Aktor, z. B. einen DC-Motor)

#### Eine Entscheidung treffen:

- · Wie treffen Menschen Entscheidungen? (Gehirn)
- Wie kann ein Roboter entscheiden? (benötigt mindestens einen Mikrocontroller, z. B. einen Arduino)

Damit haben die Lernenden die drei Kernkomponenten identifiziert, die nach der o.g. Definition einen Roboter ausmachen, und die jeweils mindestens einmal vorhanden sein muss: Sensor, Aktor und Mikrocontroller.

**Mögliche Definition eines Roboters:** "Ein Roboter ist eine programmierbare Maschine, die autonom funktionieren, ihre Umgebung wahrnehmen und ihre Aktionen entsprechend anpassen kann".

**Kernkomponenten:** Mikrocontroller, Sensoren und Aktoren.

- · Handle i verdenen
- Tage en beslutning (hvordan der skal handles, i forhold til det sansede).

Hvis disse tre krav er opfyldt, kan samtalen blive til en diskurs om, hvordan robotter kan opfylde ovenstående krav, hvorved lærerne først sætter spørgsmålstegn ved deres egen sensoriske opfattelse og fortolkning ved hjælp af følgende impulser og kontrasterer dem med robotten:

#### Den kan sanse verdenen:

- Hvordan sanser mennesker verdenen? (syns, høre, smage, lugte, følesans).
- Hvordan kan en robot sanse verdenen? (den har brug for mindst én sensor, eks. en lys-sensor).

#### Den kan handle i verdenen:

- · Hvordan handler mennesker i verdenen? (muskler).
- · Hvordan kan en robot handle? (den har brug for mindst én aktuator, eks. en DC motor)

#### Den kan tage en beslutning:

- Hvordan tager mennesker en beslutning? (hjernen).
- Hvordan kan en robot tage en beslutning? (den har brug for mindst én mikrocontroller – eks. en Arduino).

Herved er de tre kernekomponenter der i vores definition, udgør en robot og som en sådanne skal besidde mindst én af hver af, identificeret: Sensorer, aktuatorer og mikrocontrollere.

**Mulig definition af en robot:** "En programmerbar maskine, hvilken kan fungere autonomt, samt sanse sine omgivelser og tilpasse sine handlinger dertil".

**Kernekomponenter:** Mikrocontrollere, sensorer og aktuatorer.

### Phase 2 – Die Kernkomponenten, ihre Funktion und Interaktion (30 Min.):

Nach der Klärung der drei Kernkomponenten können diese näher beleuchtet werden, um zu erklären, was sie tun und wie sie interagieren. Da der Mikrocontroller sowohl die Sensoren als auch die Aktoren steuert, ist er ein guter Ausgangspunkt.

#### Was ist ein Mikrocontroller?

En Mikrocontroller ist vom Grundsatz her ein kleiner Computer. Er hat einen Prozessor (Rechenleistung), RAM (Speicher) und andere notwendige Komponenten, genau wie ein Computer. Mikrocontroller gibt es heute in fast allen Arten von elektrischen Geräten: im Getränkeautomaten, im Fernseher, in der Mikrowelle, im Wecker usw.

Der Mikrocontroller verfügt über mehrere Ein- und Ausgänge, die sogenannten I/O-Pins (kurz für "Input/Output" also Eingabe/Ausgabe). Die I/O-Pins des Mikrocontrollers können so programmiert werden, dass sie entweder die an sich selbst anliegende Spannung messen oder andere Komponenten mit Spannung versorgen.

Oft wird der Mikrocontroller verwendet, um abzulesen, mit welcher Spannung ein Sensor einen bestimmten I/O-Pin versorgt, um darauf basierend eine Auswahl zu treffen und einen Aktor mit einer Spannung zu versorgen, die wiederum eine beabsichtigte Aktion auslöst.

Das Signal, das der Mikrocontroller von einem Sensor empfängt, heißt INPUT, das Signal, das er an einen Aktor sendet, heißt OUTPUT. INPUT-Signale können entweder digital (als LOW (o V) oder HIGH (5 V)) oder analog (von o (o V) bis 1023 (5 V)) gelesen werden, während OUTPUT-Signale digital als (LOW (o V) oder HIGH (5 V)) geschrieben werden können.<sup>1</sup>



Abb. 39: Single Board Microcontroller

Fig. 39: Single Board Microcontroller

Oder auch als Pulsweitenmodulation (Pulse Width Modulation oder PWM), PWM-Signale sind eine Simulation von analogen Signalen und werden erzeugt, indem ein I/O-Pin in einem angepassten Intervall zwischen o V und 5 V schwingungsalterniert. Die Schwingungen erfolgen so schnell, dass das Signal in der Praxis als analoges Signal verwendet werden kann. Im Arduino wird ein PWM-Signal als Zahl zwischen o (o V) und 255 (5 V) geschrieben.

#### Fase – Kernekomponenterne, deres virke og samspil (30 min):

Efter at have identificeret de tre kernekomponenter, kan disse herefter gennemgås én ad gangen for at forklare hvad de gør og hvordan de spiller sammen. I og med at mikrocontrolleren styrer sensorerne såvel som aktuatorerne er denne et godt sted at begynde.

#### Hvad er en mikrocontroller?

En mikrocontroller er basalt set, en lille computer. Den har en processer (regnekraft), RAM (hukommelse) og andre nødvendige komponenter, præcist som en computer har. Mikrocontrollere findes i dag, i næsten alle former for elektrisk udstyr: I sodavandsautomaten, fjernsynet, mikrobølgeovnen, vækkeuret osv.

Mikrocontrolleren råder over en række ind og udgange, dens såkaldte I/O pins (Input/Output). Mikrocontrollerens I/O pins, kan programmeres til enten at måle hvilken spænding der er på dem, eller til selv at forsyne andre komponenter hermed.

Ofte vil man anvende mikrocontrolleren til at aflæse hvilken spænding en sensor forsyner en given I/O pin med, hvorefter den vil foretage et valg baseret herpå, for derefter at forsyne en aktuator med en spænding der udløser en tiltænkt handling, i overensstemmelse med dette valg.

Det signal mikrocontrolleren modtager fra en sensor kaldes INPUT, mens det signal mikrocontrolleren sender til en aktuator kaldes OUT-PUT. INPUT signaler kan enten aflæses som digitale (LOW (oV) eller HIGH (5V)), eller analoge (fra o (oV) til 1023 (5V)), mens OUTPUT signaler kan skrives som digitale (LOW (oV) eller HIGH (5V)) eller PWM (Pulse-width modulation) (fra o (oV) til 255 (5V)) hvilket er en simulering af et analogt signal.<sup>1</sup>

<sup>1</sup> Eller også som pulsbreddemodulation (Pulse Width Modulation eller PWM), PWM-signaler er en simulering af analoge signaler og genereres af en I / O-pin, der skifter oscillation i et justeret interval mellem o V og 5 V. Svingningerne sker så hurtigt, at signalet i praksis kan bruges som et analogt signal. I Arduino skrives et PWM-signal som et tal mellem o (o V) og 255 (5 V).

**Mikrocontroller:** Ein kleiner Computer mit programmierbaren I/O-Pins (Input/Output = Eingabe/Ausgabe), z.B. ein Arduino Uno. Die Rolle des Mikrocontrollers besteht darin, das Programm auszuführen, das zuvor programmiert worden ist, und ist sozusagen das "Gehirn" des Roboters.

Der Mikrocontroller wird häufig so programmiert, dass er ein analoges / digitales INPUT-Signal (o V - 5 V) empfängt, das Signal verarbeitet und eine Entscheidung darüber trifft, wie darauf reagiert werden soll, woraufhin ein digitales-OUTPUT-Signal (o V - 5 V) einen angeschlossenen Aktor versorgt, der die Aktion ausführt.

Digitale Signale (Input und Output): Digitale Signale sind entweder o V oder 5 V.

**Analoge Signale (Input):** Analoge Signale haben einen Wert zwischen o V und 5 V und werden vom Arduino als Zahl zwischen o (o V) und 1023 (5 V) gelesen.

**PWM-Signale (Output):** PWM-Signale (Pulse-width Modulation) sind eine Simulation von analogen Signalen und werden von einem I/O-Pin in einem benutzerdefinierten Bereich erzeugt, der abwechselnd zwischen o V und 5 V oszilliert.

#### Was ist ein Sensor?

Hier bietet es sich an, zunächst die Lernenden zu fragen, welche Beispiele für Sensoren sie bereits kennen. Anschließend kann dies mit einer Beispielliste abgeglichen werden, die folgende Sensoren beinhalten kann: Schalter, Lichtsensoren (LDR für Light Dependent Resistor), Abstandssensoren (Ultraschallsensor), Farbsensoren, Bewegungssensoren, Mikrofone, Temperatursensoren, Infrarotsensoren (IR-Sensor)) usw.

Allen diesen Sensoren gemein ist, dass sie Änderungen in der Umgebung erkennen, in der sie sich befinden. Bei der Gelegenheit kann auch thematisiert werden, dass Roboter mit bestimmten Sensoren Dinge wahrnehmen können, die wir nicht erkennen können, so z.B. Ultraschall oder Infrarotlicht.

Daran schließt sich ein kurzer Exkurs über die allgemeine Verwendung von Sensoren an: Einige Sensoren senden ständig ein Signal an den Mikrocontroller (z. B. LDR) zurück, während andere erst aktiviert werden müssen, bevor ein Signal zurückgesendet wird (z. B. ein Abstandssensor). Gemeinsam haben sie indessen, dass der Mikrocontroller dieses Signal als INPUT empfängt, das dann gelesen und interpretiert werden kann.

Gleichzeitig wird die überwiegende Mehrheit der Sensoren, mit denen die Lernenden arbeiten, ein analoges Signal von o V - 5 V zurückgeben, das vom 10-Bit-ADC (Analog-Digital-Wandler) des Mikrocontrollers in eine Zahl von o bis 1023 (LDR, Temperatur, Potentiometer, IR-Sensor usw.) übersetzt wird, während der Abstandssensor die Zeit misst, die vergeht, wenn ein INPUT-Signal von HIGH (5 V) auf LOW (0 V) wechselt.

Zum Abschluss der Einführung in die Sensoren können Bilder verschiedener Exemplare gezeigt werden, mit denen die Lernenden im weiteren Verlauf arbeiten werden, damit sie sich bereits jetzt visuell mit deren Aussehen vertraut machen können.

**Mikrocontroller:** En lille computer der indeholder programmerbare I/O pins (Input/Output), eks. en Arduino Uno. Mikrocontrollerens rolle er at afvikle det program, programmøren har overført til den og den er derfor så at sige "hjernen" i robotten.

Mikrocontrolleren vil ofte blive programmeret til at modtage et analogt/digitalt INPUT signal (oV -5V), behandle signalet og foretage en beslutning om hvordan der skal handles herpå, hvorefter et digitalt/PWM OUTPUT signal (oV -5V) forsyner en tilsluttet aktuatorer, der udfører denne handling.

Digitale signaler (input og output): Digitale signaler er enten oV eller 5V.

Analoge signaler (input): Analoge signaler har en værdi der befinder sig et sted mellem oV og 5V og læses af Arduinoen som et tal mellem o (oV) og 1023 (5V).

**PWM signaler (output):** PWM signaler (Pulse-width Modulation), er en simulering af analoge signaler og skabes ved at en I/O pin, i et tilpasset interval, svinger mellem skiftevis at være oV og 5V. Svingningerne foregår så hurtigt, at signalet i praksis kan anvendes som et analogt signal. I Arduinoen skrives et PWM-signal som et tal mellem o (oV) og 255 (5V).

#### Hvad er en sensor?

Der kan med fordel begyndes med, at eleverne spørges om, hvilke eksempler på sensorer de kender til i forvejen, hvorefter en eksempelliste fremvises, eks. indeholdende: Kontakter, lyssensorer (LDR – Light Dependent Resistor), afstandssensorer (ultralydssensor), farvesensorer, bevægelsessensorer, mikrofoner, temperatursensorer, infrarødsensorer (IR-Sensor) osv.

Fælles for alle disse sensorer er, at de registrerer ændringer i det miljø de befinder sig i. I samme ombæring kan der kommes ind på om robotter kan have sensorer, der kan opfange ting mennesker ikke kan, så f.eks. ultralyd eller infrarødt lys.

Herefter kan der følge en kort gennem gang af, hvordan sensorer overordnet set anvendes: Nogle sensorer sender konstant et signal tilbage til mikrocontrolleren (eks. LDR), mens andre først skal aktiveres førend de sender et signal tilbage (eks. afstandssensoren). Fælles er dog, at mikrocontrolleren modtager dette signal som et INPUT, der så kan aflæses og fortolkes.

Samtidigt vil langt de fleste sensorer eleverne kommer til at arbejde med, returnere et analogt signal på oV – 5V, hvilket oversættes af mikrocontrollerens 10-bit ADC (analog-to-digital converter) til et tal på 0–1023 (LDR, Temperatur, Potentiometer, IR-Sensor mfl.), hvorimod eks. afstandssensoren måler tiden der går mellem at et INPUT signal går fra HIGH (5V) til LOW (oV).

Afslut gerne ved at vise billeder af de forskellige sensorer eleverne kommer til at arbejde med i undervisningen, sådan at de allerede nu bliver visuelt bekendt med hvordan disse ser ud.

**Sensor:** Der Sensor erkennt Änderungen in der Umgebung, in der er sich befindet, wie z. B. Änderungen des Lichtpegels (LDR, IR-Sensor), des Schallpegels (Mikrofon), der Entfernung des Sensors zum nächsten Objekt (Ultraschallsensor) usw.

**Signal:** Der Mikrocontroller empfängt den Sensorwert als ein INPUT-Signal, welches entweder analog oder digital gelesen wird.

#### Was ist ein Aktor?

Wie bei den Sensoren können die Lernenden auch hier gefragt werden, welche Beispiele sie schon kennen; die Rückmeldung kann mit einer Beispielliste abgeglichen werden, die folgende Objekte enthalten kann: Motoren (DC, Servo, Stepper), LEDs (Leuchtdioden), Lautsprecher, Heizelemente usw.

Danach kann das gemeinsame Merkmal all dieser Objekte identifiziert werden: Sie beeinflussen die Umgebung, in der sie sich befinden. Auch hier kann darauf eingegangen werden, dass beispielsweise Infrarot-LEDs und Ultraschalllautsprecher Aktionen ausführen können, die Menschen nicht ausführen können.

Anders als Sensoren, die ein Signal an den Mikrocontroller senden, den dieser als INPUT empfängt, sendet der Mikrocontroller hier einen OUTPUT, das der Aktor empfängt. Bei den in diesem Material genutzten Aktoren ist das verwendete OUTPUT-Signal entweder digital (LOW (o V) oder HIGH (5 V)) oder PWM (Pulsweitenmodulation, von o (o V) bis 255 (5 V)).

**Aktor:** Der Aktor beeinflusst die Umgebung, in der er sich befindet, sei es durch physisch bewegliche Teile wie Motoren (DC, Servo, Stepper) oder auch durch Änderungen des Lichtpegels (LED), des Schallpegels (Summer, Lautsprecher), der Temperatur (Heizelemente) usw.

Signal: Der Mikrocontroller sendet den Aktorwert als ein OUTPUT-Signal.

#### Zusammenfassung

Fassen Sie Nachstehendes zusammen und gehen Sie mit den Lernenden die Bilderserie von vorhin noch einmal durch. Besprechen Sie noch einmal, ob die Objekte auf den Bildern Roboter sind oder nicht, bzw. die Anforderungen an einen Roboter erfüllen. Können die Lernenden die drei Kernkomponenten identifizieren (unter Beachtung des Umstandes, dass der Mikrocontroller häufig verborgen ist, somit lediglich davon ausgegangen werden muss, dass er vorhanden ist)? Die Zusammenfassung kann vorteilhafterweise im Prozessverlauf mehrmals wiederholt werden – auch um die Lernenden in der Identifizierung und Analyse von Robotern zu schulen und die (Kenntnisse der) Interaktion zwischen Mikrocontrollern, Sensoren und Aktoren aufzufrischen. Verwenden Sie daher gern jedes Mal neue Bilderserien, aber auch Gegenstände aus dem Alltag der Lernenden.

**Sensor:** Sensorer registrerer ændringer i det miljø de befinder sig i, dette kan eks. være interaktionen med en kontakt, ændringer i lysniveauet (LDR), lydniveauet (mikrofon), sensorens afstand til det nærmeste objekt (ultralydssensor) osv.

**Signal:** Mikrocontrolleren modtager sensorens værdi, som et INPUT signal, hvilket enten aflæses analogt eller digitalt.

#### Hvad er en aktuator?

Som ved sensorer, kan eleverne spørges ind til hvilke eksempler de kender til i forvejen, hvorefter en eksempelliste kan fremvises, eks. indeholdende: motorer (DC, servo, stepper), LED'er (Light Emitting Diodes), højtalere, varmeelementer osv.

Herefter kan fællestrækket ved disse identificeres, navnlig at de påvirker det miljø de befinder sig i, mens der kan kommes ind på at eks. infrarøde LED'er og ultralydshøjtalere, kan udføre handlinger som mennesker ikke er i stand til.

I modsætning til sensorer der sender et signal til mikrocontrolleren, hvilket denne modtager som et INPUT, så er det i denne forbindelse mikrocontrolleren der her sender et OUTPUT, hvilket aktuatoren modtager. I forbindelse med de aktuatorer eleverne forventes at arbejde med i forløbet, er signalet der her anvendes som OUTPUT, enten digitalt (LOW (oV) eller HIGH (5V)), eller PWM (Pulse-width Modulation) (fra o (oV) til 255 (5V)).

**Aktuator:** aktuatorer påvirker det miljø de befinder sig i, dette kan eks. være gennem fysisk bevægelige dele såsom motorer (DC, servo, stepper), men også gennem ændringer i lys-niveauet (LED), lyd-niveauet (buzzer, højtaler), temperaturen (varme-elementer) osv.

**Signal:** Mikrocontrolleren sender kommando til aktuatoren, gennem et OUTPUT signal, hvilket enten er digitalt eller PWM.

#### **Opsummering**

Opsummer nedenstående og gennemgå herefter den samme billedserier fra tidligere, i plenum sammen med eleverne. Diskuter endnu engang, hvorvidt objekterne på billederne er robotter eller ej, opfylder de vores krav til en robot? Kan de tre kernekomponenter identificeres (vær obs. på at mikrocontrolleren ofte er skjult, hvorfor man må nøjes med at formode at den er der)? Opsummeringen kan med fordel gentages flere gange gennem forløbet, for derved at træne eleverne i at identificere og analysere robotter, samt genopfriske samspillet mellem mikrocontrollere, sensorer og aktuatorer. Anvend derfor gerne nye billedserier for hver gang, samt objekter fra elevernes hverdag.

#### **Definition:**

"Ein Roboter ist eine programmierbare Maschine, die autonom funktionieren und ihr Verhalten den Umgebungen anpassen kann."

Anders ausgedrückt: Ein Roboter kann seine Umgebung wahrnehmen, auf Basis dessen eine vorprogrammierte Entscheidung treffen und entsprechend reagieren.

#### Kernkomponenten:

- · ein Mikrocontroller, der Entscheidungen trifft.
- ein Sensor, der Veränderungen in seiner Umgebung erfasst.
- ein Aktor, der Veränderungen seiner Umgebung herbeiführt.

#### Übersicht:

	Mikrocontroller:	Sensoren:	Aktoren:
Eigenschaft:	Treffen Entscheidungen, 'hören auf' Sensoren und 'befehligen' Aktoren.	Registrieren Veränderun-gen in ihrer Umgebung.	Verändern ihre Umgebung.
Signal:	Empfangen ein INPUT-Signal von Sensoren. Senden ein OUTPUT-Signal an Aktoren.	Senden ein INPUT-Signal an den Mikrocontroller.	Empfangen ein OUTPUT-Sig- nal vom Mikrocontroller.

#### Phase 3 - Was ein Roboter nicht ist (10 Min.):

Es gibt jetzt eine für den weiteren Verlauf geltende Definition, welche Anforderungen ein Objekt erfüllen muss, bevor es als Roboter bezeichnet werden kann. Gleichzeitig haben die Lernenden jetzt einen kurzen Überblick darüber erhalten, wie die entsprechenden Kernkomponenten funktionieren und in Beziehung zueinander stehen. Aber es ist oft (mindestens) genauso wichtig zu wissen, was ein Roboter nicht kann. Dies ist zum Beispiel sehr relevant, wenn Menschen mit einem Roboter zusammenarbeiten, etwa im Bereich der Industrie.

Fragen Sie die Lernenden , ob sie an dieser Stelle auf Basis ihrer Erfahrungen Beispiele dazu anführen können, was Roboter nicht sind bzw. nicht können – vor dem Hintergrund, dass Roboter ja durchaus, wie wir Menschen auch, die Welt wahrnehmen, in der Welt agieren und Entscheidungen treffen können

Ein gutes Thema wäre z. B., dass Roboter keine 'Gefühle' vergleichbar mit denen von Menschen haben. Eine mögliche Idee wäre, die Lernenden zunächst zwei Roboter kennenlernen zu lassen, einen 'niedlichen' à la NAO, PARO usw., und einen Industrieroboter, z. B. ein Schweißgerät. Anschließend können die Lernenden gefragt werden, welcher der beiden Roboter wohl die größten empathischen Gefühle haben kann. Natürlich ist keiner der Roboter dazu in der Lage, aber die Lernenden werden im Laufe ihres Lebens mit Robotern konfrontiert, die darauf ausgelegt sind, empathische Gefühle vorzugaukeln. Auch werden ihnen durch die modernen Medien Roboter begegnen, die scheinbar zu echten Gefühlen fähig sind.

#### **Definition:**

"En programmerbar maskine, hvilken kan fungere autonomt, samt tilpasse dennes opførsel på omgivelserne".

Dvs. en robot kan sanse omverdenen, tage en beslutning, samt handle i overensstemmelse dermed (evt. handle derefter).

#### Kernekomponenter:

- · En mikrocontroller, der tager beslutninger.
- En sensor, der registrerer ændringer i sit miljø.
- En aktuator, der udfører ændringer i sit miljø.

#### Oversigt:

	Mikrocontrollere:	Sensorer:	Aktuatorer:
Egenskab:	Tager beslutninger. Lytter til sensorer og kommanderer aktuatorer.	Registrerer ændringer i det miljø de befinder sig i.	Påvirker det miljø de befinder sig i.
Signal:	Modtager et INPUT fra sensorer. Sender et OUTPUT til aktuatorer.	Sender et INPUT til mikrocon- trolleren, hvilket aflæses enten analogt eller digitalt.	Modtager et OUTPUT fra mi- krocontrolleren, hvilket enten er digitalt eller PWM.

#### 3. Fase - Hvad er en robot ikke (10 min):

Den for kurset gældende definition på hvilke krav et objekt skal opfylde, førend vi kan kalde det for en robot er nu blevet defineret, samtidigt med at der også har været en kort gennemgang af, hvordan kernekomponenterne fungerer og spiller sammen i relation til hinanden. Men det er ofte mindst lige så vigtigt at vide, hvad en robot ikke kan, som hvad den kan. Dette er eks. meget relevant når mennesker arbejder sammen med robot, eks. i industrien, for robotten tænker ikke på dig og tager derfor heller ikke hensyn den ikke er programmeret til at tage.

Spørg eleverne i plenum, om de har nogle ideer om hvad en robot ikke er/ikke kan, selvom de ligesom os mennesker, kan sanse verdenen, handle i verdenen, samt tage en beslutning?

Nogle gode emner at komme ind på, er eks. at robotter ikke er i stand til at have emotionelle følelser. Denne dialog kan med fordel startes ved at eleverne først præsenteres for to robotter, en "cute" robot a la. NAO, PARO etc., samt en industriel robot, eks. en svejsemaskine. Herefter kan eleverne spørges ind til, hvilken af de to robotter der er i stand til at have de største empatiske følelser. Ingen af robotter er selvfølgeligt i stand hertil, men eleverne vil igennem deres liv, blive mødt med robotter der er designet til at indikere at de besidder empatiske følelser, ligeledes vil de gennem moderne medier, ofte blive præsenteret for robotter som værende i stand hertil.

Det er samtidigt vigtigt at slå fast overfor eleverne, at robotter ikke er selvbevidste og ikke er intelligente i klassisk forstand – igen, uagtet hvad film og andre medier fremstiller dem

Gleichzeitig ist es wichtig, die Lernenden darauf hinzuweisen, dass Roboter im klassischen Sinne nicht selbstbewusst und nicht intelligent sind – unabhängig davon, was Filmindustrie und andere Medien das Publikum glauben machen wollen. Sie agieren und arbeiten ausschließlich aufgrund logischer Schaltkreise, denn alle ihre Entscheidungen basieren ausschließlich auf mathematischen Berechnungen, welche sie allerdings rasend schnell auszuführen in der Lage sind. Und aus genau diesem Grund sollte genügend Zeit der Feststellung gewidmet werden, das Menschen bei Durchführung all dieser Berechnungen mit Stift und Papier zu genau der gleichen Entscheidung kommen würden wie der Roboter.

Aus dem gleichen Grund sind Roboter auch weder reflektierend, intuitiv noch kreativ. Sie tun genau das, wofür sie programmiert worden sind und nichts anderes. Zum Beispiel kann der Roboter keinen schlechten Tag haben oder 'mürrisch drauf' oder unaufmerksam sein. Als Beispiel kann ein Roboter dienen, der so programmiert ist, dass er geradeaus fährt. Was passiert, wenn man diesen Roboter auf einen Tisch stellt? Er fährt ohne zu zögern direkt über die Tischkante hinaus, ohne zu irgendeinem Zeitpunkt eine Meinung über diese Handlungsweise gehabt zu haben oder die Folgen seiner Handlung zu bedenken. Er stoppt auch nicht intuitiv vorher ab, wie ein Mensch das tun würde. Der zu Beginn angesprochene Roboter in der Industrie denkt auch nicht an seinen menschlichen Kollegen oder nimmt keine Rücksicht auf Dinge, die er gemäß Programmierung nicht berücksichtigen soll.

Roboter sind **nicht** intelligent im klassischen Sinne, nicht intuitiv oder reflektierend und sich ihrer eigenen Existenz weder bewusst noch in der Lage, Gefühle zu haben.

Roboter treffen Entscheidungen, die auf Mathematik und Logik beruhen. Wenn Menschen genügend Zeit einplanen, werden sie mit Papier und Bleistift nach dem vorgegebenen Entscheidungsbaum immer die gleiche Entscheidung treffen wie der Roboter.

som værende. Forklar eleverne at de udelukkende operer på baggrund af et logisk kredsløb, dvs. alle deres beslutninger udelukkende er baseret på matematiske beregninger, hvilket de til gengæld er lynenes hurtige til at udføre. Men af samme årsag, skulle man derfor afsætte tilstrækkeligt med tid dertil, vil man ved at gennemgå alle beregningerne med papir og blyant, altid komme frem til den samme beslutning som robotten.

Af samme årsag er robotter heller ikke hverken reflekterende, intuitive eller kreative. De gør præcist det de er programmeret til, og intet andet. Robotter kan derfor eks. ikke have en dum dag, være træls eller ikke gide gøre det vi siger til dem. Som et eksempel kan man tage en robot der er programmeret til at køre lige ud. Sætter denne robot nu op på et bord, hvad sker der så? Den kører direkte udover kanten uden på noget tidspunkt at have haft en mening herom, eller på nogen måde reflektere over konsekvenserne herved, hverken før eller efter. Den stopper altså ikke intuitivt op, på samme måde som vi mennesker ville gøre det. Dette kan nogle gange være svært for eleverne at forstå og det er derfor vigtigt at slå det fuldstændig fast, at robotter udelukkende gør det de er programmeret til, hverken mere eller mindre.

**Robotter er ikke:** Intelligente i klassisk forstand, intuitive, reflekterende og selvbevidste eller i stand til at have følelser.

Robotter tager beslutninger baseret på matematik og logik: Hvis man derfor afsætter tilstrækkeligt med tid, vil man med papir og blyant, altid komme frem til den samme beslutning som robotten.

# EINFÜHRUNG IN DAS THEMA ROBOTER – WARUM ES SO SCHWER IST, SIE HERZU-STELLEN (30 MIN.)

**Kurz zum Inhalt:** Der Schwerpunkt dieses Abschnitts liegt auf einer körperlichen Aktivität, bei der Lernende in Dreiergruppen interagieren und sich in die Rolle eines Roboters begeben sollen, indem sie seine drei Kernkomponenten simulieren: den Mikrocontroller, den Sensor und den Aktor.

Lernergebnisse: Durch körperliche Aktivität erfahren die Lernenden die grundlegenden internen Kommunikationskanäle, aus denen die Interaktion zwischen Mikrocontroller, Sensor und Aktor (INPUT- und OUT-PUT-Signal) besteht. Gleichzeitig lernen sie die Einschränkungen von Robotern kennen, die im Gegensatz zu der oft intuitiven Verhaltens- und Vorgehensweise von Personen bei neuen Rahmenbedingungen immer einem vorgegebenen Schema folgen.

Physisches Material je Gruppe (3 Lernende je Gruppe): 2 Augenbinden, 3 einfarbige Kunststoffbecher o. ä.

### Phase 1 – Einführung, Regeln, Aufbau (Setup), Herausforderungen und Ziele (12 Min.):

Diskutieren Sie mit den Lernenden, warum es so schwierig ist, Roboter herzustellen, obwohl einige Gemeinsamkeiten mit Menschen bestehen (Gehirn: Mikrocontroller, Sinne: Sensoren, Muskeln: Aktoren). Hier spielt insbesondere die Erörterung in der vorangegangenen Phase darüber, was Roboter nicht sind, eine zentrale Rolle. Man denke nur daran, wie oft Personen z. B. intuitiv handeln, wenn sie einfach mal einen Waldspaziergang machen.

Fragen Sie auch die Lernenden, welcher Roboter ihrer Meinung nach am einfachsten herzustellen ist: einer, der in einer kontrollierten Umgebung tausende Sachen ausführen kann, oder einer, der in einer unkontrollierten Umgebung eine und nur diese eine Aufgabe ausführen kann? Einfacher ist natürlich die kontrollierte Umgebung, da hier die relevanten Variablen bekannt sind – weshalb der Mangel an Intelligenz, Reflexionsfähigkeit und Intuition beim Roboter nicht problematisch ist.

Gleichzeitig wurde die menschliche Kommunikation zwischen Gehirn, Sinnen und Muskeln durch Millionen von Jahren natürlicher Evolution ständig weiter verfeinert, weshalb sie so sehr in unser System

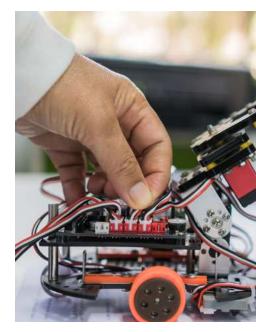


Abb. 40: Konstruktion, Bau und Programmierung von Robotern bieten einen guten Einstieg in digitale Technologien

Fig. 40: Design, bygning og programmering af robotter giver en god introduktion til digitale teknologier



### INTRODUKTION TIL ROBOTTER – HVORFOR ER DET SÅ SVÆRT AT LAVE ROBOTTER? (30 MIN.)

**Kort om indholdet:** Lektionens omdrejningspunkt er en fysisk aktivitet, hvor elever i grupper af tre, skal interagere sammen og gøre det ud for en robot, ved at simulere dennes tre kernekomponenter: Mikrocontrolleren, sensorerne og aktuatorerne.

**Læringsudbytte:** Eleverne bliver gennem fysisk aktivitet bekendt med de grundlæggende interne kommunikationskanaler der udgøres af samspillet mellem mikrocontrolleren, sensorerne og aktuatorerne (INPUT og OUTPUT). Samtidigt lærer de om robotters begrænsninger, hvilket sættes i relation til vores intuitive måde at agere på i ukendte miljøer.

Fysiske materialer pr. gruppe (3 elever pr. gruppe): 2 stk. stof der kan anvendes som bind for øjnene, samt 3 stk. ens farvede plastikkrus eller lign. objekter.

### 1. Fase – Introduktion, Regler, Opsætning, Udfordringer og Mål (12 min):

Introducer eleverne til hvorfor det er så svært at lave robotter, selvom de indtil videre langt hen ad vejen, kan sammenlignes med mennesker (hjernen:mikrocontrolleren, sanser:sensorer, muskler: aktuatorer)? En af de grundlæggende forklaringer er selvfølgeligt relaterbar til den afsluttende snak i sidste lektion om, hvad robotter ikke er. Tænk blot på hvor ofte man vi foretager os eks. en intuitiv handling, blot ved at gå rundt i skovbunden.

Spørg evt. eleverne hvilken robot de tror er det nemmeste at lave: En robot der kan udføre 1000vis af ting i et kontrolleret miljø, eller en robot der kan udføre 1 ting i et ukontrolleret miljø? Forklar herefter at det nemmeste er det kontrollerede miljø, fordi alle variabler der indgår derved er kendt på forhånd, hvorfor robottens manglende intelligens og evne til at reflektere, samt tænke intuitivt ikke længere problematisk i samme grad.

Samtidigt er vores kommunikation mellem hjernen, sanserne og musklerne, forfinet gennem millioner af års naturlig udvikling, hvorfor det nu er så indlejret i vores system at vi ikke længere, aktivt skal tænke over hvordan inputtet fra vores sanser skal fortolkes, eller hvordan outputtet til vores muskler skal formuleres.

eingebettet ist, dass nicht mehr aktiv darüber nachgedacht werden muss, wie Inputs der eignen Sinne interpretiert werden sollen oder Outputs an die eigenen Muskeln formuliert werden müssen.

Nach der Einführung in das Problemfeld werden die Lernenden in die Regeln, den Aufbau sowie die Herausforderungen und Ziele der Aktivität eingeführt (siehe unten):

Bei Lebewesen sind das Gehirn, die Sinne und Muskeln über Millionen Jahre natürlicher Evolution genau aufeinander abgestimmt. Mit Robotern ist das nicht so. Gleichzeitig sind Roboter im klassischen Sinne nicht intelligent oder in der Lage zu reflektieren und intuitiv zu handeln. Daher ist es äußerst schwierig, Roboter herzustellen, weil sie sich wegen ebendieser Mängel nicht automatisch an unbekannte Variable anpassen können.

#### Regeln:

- Drei Lernende bilden zusammen einen Roboter (Mikrocontroller, Sensor und Aktor):
- · Der Mikrocontroller (trägt eine Augenbinde):
  - Kann Entscheidungen treffen.
  - Kann den Sensor nach Informationen fragen und sich die Antwort anhören.
  - Kann dem Aktor Befehle erteilen.
- Der Sensor:
  - Kann die Welt wahrnehmen und dem Mikrocontroller antworten, wenn dieser ihn fragt.
- Der Aktor (trägt eine Augenbinde):
  - Kann durch Hören Befehle des Mikrocontrollers aufnehmen und sie ausführen.

#### Aufbau:

- Weisen Sie jeder Gruppe einen Tisch zu, vor dem sich die Gruppe hinstellt.
- Nachdem Mikrocontroller und Aktor die Augen verbunden worden sind, verteilen Sie drei Kunststoffbecher im nahen Umfeld des "Roboters" – gern in unterschiedlichen Höhen (unter einem Stuhl, auf dem Nachbartisch u. Ä.).

#### Herausforderungen und Ziele:

- Lokalisieren der drei Kunststoffbecher.
- · Einsammeln der drei Kunststoffbecher.
- Stapeln der drei Kunststoffbecher auf dem zugewiesenen Tisch.

#### Phase 2 - Die praktische Umsetzung (18 Min.):

Die Lernenden sollen – ein\*e jede\*r in ihrer bzw. seiner Roboterfunktion – zusammenarbeiten, um die Aufgabe zu bewältigen und die Kunststoffbecher auf ihren jeweiligen Tischen zu stapeln. Wenn das Ziel erreicht ist, wird rotiert, d. h. die Lernenden schlüpfen in eine andere Roboterfunktion – bis sie alle drei Rollen (Mikrocontroller, Sensor, Aktor) ausprobiert haben.

Efter introduktionen til problemstillingen, introduceres eleverne til aktivitetens regler, opsætning samt udfordringer og mål, se nedenstående:

**Problemstilling:** Som levende væsner er vores hjerne, sanser og muskler gennem millioner af års naturlig evolution, blevet fintunet til at arbejde sammen, dette gør sig ikke gældende for robotter. Samtidigt er robotter ikke intelligente i klassisk forstand eller i stand til at reflektere og foretage intuitive handlinger. Derfor er det også utroligt svært at lave robotter, fordi deres mangler gør, at de ikke automatisk kan tilpasse sig ukendte variabler.

#### Regler:

- Tre elever udgør tilsammen én robot (mikrocontroller, sensor og aktuator):
- Mikrocontroller (har bind for øjnene):
  - Kan tage beslutninger.
  - Kan spørge sensoren efter information og lytte til svaret.
  - Kan give aktuatoren kommandoer.
- · Sensoren:
  - Kan sanse verdenen og svare mikrocontrolleren, når denne bliver spurgt.
- · Aktuatoren (har bind for øjnene):
  - Kan lytte til kommandoer fra mikrocontrolleren, samt udføre dem.

#### Opsætning:

- Udpeg et gruppebord for hver gruppe og placer dem foran dette.
- Efter at mikrocontrolleren og aktuatoren har fået bind for øjnene, fordeles de tre plastikkrus i "robottens" nærmiljø, gerne i forskellige højde niveauer (under en stol, på nabobordet og lign.).

#### Udfordringer og mål:

- · Lokaliser de tre plastikkrus.
- · Indsaml de tre plastikkrus.
- Stabl de tre plastiskkrus på det udpegede bord.

#### 2. Fase – Aktiviteter (18 min):

Inddel eleverne i grupper af tre elever pr. gruppe, hvorefter aktiviteten kan påbegyndes. Herefter skal eleverne nu i virke af den robot de udgør, forsøge at arbejde sammen om at få løst udfordringen og nå i mål med at stable plastikkrussene på deres respektive borde. Når målet er nået, roteres der og eleverne i gruppen tildeles en ny rolle i robotten, dette gentages tre gange indtil alle tre elever, har forsøgt sig med alle tre roller (mikrocontroller, sensor, aktuator).

## EINFÜHRUNG IN DEN ARDUINO – KENNENLERNEN DES ARDUINO-SYSTEMS (6 STD. UND 30 MIN.)

**Kurz zum Inhalt:** Dieser Abschnitt mischt Theorie mit praktischen, konkreten Hands-on-Übungen zu Verständnis und Anwendung verschiedener ausgewählter Sensoren und Aktoren sowie zu grundlegenden Programmierprinzipien. Die hier besprochenen Programmierprinzipien, Sensoren und Aktoren sind Teil des abschließenden Projekts.

**Lernergebnisse:** Die Lernenden eignen sich ein grundlegendes Verständnis von Elektrizität, Mikrocontrollertechnologie (Arduino Uno), Programmierprinzipien sowie Sensoren und Aktoren an. Darüber hinaus können sie mit einfachen Input/Output-Systemen selbst arbeiten und diese erstellen.

Konkretes Material je Gruppe (2 Lernende je Gruppe): Dieser Abschnitt enthält eine größere Auswahl verschiedener Materialien und Komponenten. Die vollständige Liste finden Sie im Downloadbereich auf der Webseite des Projekts (http://panama-project.eu/). Dies gilt auch für die entwickelten Spickzettel (cheat sheets), von denen Teile in diesem Abschnitt enthalten sind, sowie für Programmbeispiele für die verschiedenen Sensoren und Aktoren; ebenso Lösungsvorschläge für die verschiedenen Aufgaben. Die entwickelten Spickzettel können in der ersten Phase dieses Abschnitts ausgedruckt und verteilt werden.

**Hinweis:** Nach jeder Lektion gibt es ein paar Aufgabenbeispiele für die Schülerinnen und Schüler, mit denen sie mit den Schaltkreisen und dem Code aus der Lektion experimentieren können. Es wird daher empfohlen, den Schülern zu Beginn jeder Lektion die darin verwendeten Materialien auszuhändigen, damit sie die Schaltung und den Code für sich selbst kopieren können.

#### Erläuterung zum weiteren Procedere (10 Min.):

Im weiteren Verlauf sollten die Schülerinnen und Schüler eine Einführung in das endgültige Themenprojekt und die folgenden Phasen erhalten, um die bevorstehenden Lektionen in einem konkreten Kontext sehen zu können. Sie lernen so beispielsweise den mit IR-Sensoren versehenen Transportroboter kennen sowie in diesem Zusammenhang die Verwendung von Mikrocontrollern.

### Phase 1 – Einführung in die Themen Elektrizität und Aktoren (LED) (20 Min.):

Diskutieren Sie mit den Lernenden, wie Elektrizität grundlegend in elektrischen Schaltkreisen fließt, und erläutern Sie kurz, was Strom, Spannung und Widerstand sind. Auf detaillierte Berechnungen kann verzichtet werden.

### INTRODUKTION TIL ARDUINO – LÆR ARDUINO SYSTE-MET AT KENDE (6 TIMER OG 30 MINUTTER)

**Kort om indholdet:** Lektionen blander teori med praktiske og konkrete hands-on øvelser, vedr. forståelsen for og anvendelsen af en række udvalgte sensorer og aktuatorer, samt grundlæggende programmerings principper. De programmeringsprincipper, sensorer og aktuatorer er en del af det afsluttende temaprojekt.

**Læringsudbytte:** Eleverne opnår en basal forståelse for elektricitet, mikrocontroller teknologi (Arduino Uno), programmeringsprincipper, samt sensorer og aktuatorer. Derudover bliver de sat i stand til selv at kunne arbejde med og konstruere simple input/output systemer.

Fysiske materialer pr. gruppe (2 elever pr. gruppe): Lektion indbefatter et større udvalg af forskellige materialer og komponenter, den fulde liste kan findes i downloadområdet på projektets hjemmesiden (http://panama-project.eu/). Det samme kan de udviklede cheat sheets, hvoraf dele er inkluderet i dette afsnit, samt eksempler på programmer til de forskellige sensorer og aktuatorer, såvel som løsningsforslag til de forskellige opgaver. De udviklede cheat sheets, kan med fordel printes ud og uddeles allerede i første fase af denne lektion.

**Note:** Efter hver fase er der givet et par eksempler på opgaver, der lader eleverne eksperimentere med de gennemgåede opstillinger. Det anbefales derfor, at hver fase påbegyndes ved at de anvendte materialer deri udleveres til eleverne, sådan at de løbende kan kopiere disse.

#### Introduktion til det videre forløb (10 min):

Introducer eleverne til det afsluttende temaprojekt, for derved at sætte de kommende lektioner i en konkret kontekst, med det formål at fjerne en del af det abstrakte og ukonkrete element heraf. De følgende faser kan ligeledes med fordel – hvor det er muligt – præsenteres i relation til temaprojektet. Dette således at eleverne eks. bliver bekendt med at transporteringsrobotten anvender IR-sensorer, samt hvorfor og hvor de er placeret, førend de går i gang med at lære om disses anvendelse i relation til mikrocontrolleren.

#### 1. Fase - Introduktion til elektricitet og aktuatorer (LED) (20 min):

Introducer eller genopfrisk basalt kendskab til elektricitet og hvordan strøm flyder i elektriske kredsløb, forklar kort hvad strøm, spænding og modstand er. De nærmere udregninger kan udelades.

Tegn et diagram for en strømkilde (5V), der forsyner en LED (Rød – 5mm), ink. formodstand 150 $\Omega$  og gennemgå dette for eleverne.

Zeichnen Sie das Diagramm einer Stromquelle (5 V), die eine LED (rot – 5 mm) einschließlich Vorwiderstand von 150  $\Omega$  (für eine Wiederholung der Widerstandsgrößen und -farbmarkierungen vgl. Abb. 42) versorgt, und erläutern Sie dieses.

Erläutern Sie nun das Steckbrett (breadboard) und wie die Öffnungen untereinander leitend verbunden sind, gern mithilfe bildlicher Darstellungen (vgl. u. a. Abb. 41). Danach kann der Schaltkreis auf dem Steckbrett aufgebaut werden, während der Arduino mit seinen 5V- und GND (Ground, deutsch: Erdung)-Pins als Stromversorgung hierfür dienen kann. Lassen Sie die Lernenden die Schaltung auf ihrem eigenen Steckbrett nachbauen.

Erklären Sie, dass damit eine der drei Kernkomponenten an einen Roboter realisiert ist: der Einsatz einer LED als Aktor.

#### Phase 2 - Einführung in den Arduino (20 Min.):

Stellen Sie den Arduino und seine allgemeine Funktionalität vor, insbesondere mit Fokus auf die Pins und deren Funktion. Führen Sie ebenfalls in die Programmierungsumgebung des Arduino ein.

Verwenden Sie jetzt den Arduino, um die LED von vorhin zu steuern anstatt ihn nur zur Stromversorgung zu verwenden. Die Anleitung sollte für alle gleichzeitig sichtbar sein, damit die Lernenden es an ihren eigenen Computern nachvollziehen und das Programm gleichzeitig mit der Lehrkraft programmieren können. Beispiele für die Programmierung wie auch für die nachfolgenden Aufgaben finden Sie in den Abbildungen 44 –44.

Beginnen Sie mit der Einführung in die Variablen. Hier wäre es auch möglich, kurz auf die am häufigsten verwendeten Datentypen (int, String, boolean) einzugehen, da der angeschlossene Pin später als int (integer, deutsch: Zahl) gespeichert wird. Erklären Sie anschließend, wie die Programmstruktur funktioniert, und stellen Sie den angeschlossenen Pin als OUTPUT ein. Dann wird im Loop-Teil der angeschlossene Pin auf HIGH eingestellt. Besprechen Sie gern auch noch einmal den Unterschied zwischen INPUT und OUTPUT, und warum für die LED stets OUTPUT verwendet wird sowie den Unterschied zwischen LOW (o V) und HIGH (5 V).

#### Mögliche Analogie:

An dieser Stelle ist ein Vergleich zur Verdeutlichung hilfreich, beispielsweise mit befüllten Umzugskartons, die in der Garage stehen und auf mit einem Aufkleber versehen sind. So ist es einfach, stets den gewünschten Karton zu finden und das zu finden, was enthalten ist. Ebenfalls kann ein Gegenstand herausgenommen und durch



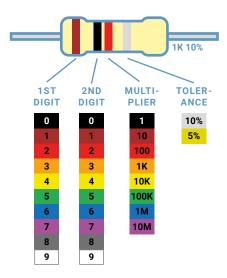


Abb. 42: Übersicht darüber, wie die Farbcodes der Widerstände abzulesen sind

Fig. 42: Oversigt over hvordan farvekoderne på modstande aflæses

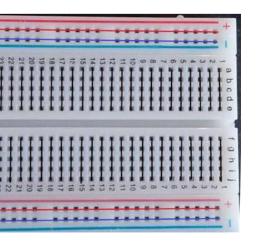


Abb. 41: Übersicht über die internen Verbindungen des Steckbretts

Fig. 41: Oversigt over breadboardets interne forbindelser

Introducer dernæst breadboardet og hvordan dets huller hænger internt sammen, vis gerne illustrationer heraf. Konstruer kredsløbet på breadboardet, mens Arduinoen kan agere strømforsyning hertil, ved at anvende dennes 5V og GND pins.

Ved anvendelsen af en aktuator (LED), er ét af de tre krav til en robot, hermed opfyldt.

#### 2. Fase - Introduktion til Arduinoen (45 min):

Introducer Arduinoen og gennemgå overordnet dens funktionalitet, hvad er pins, hvad er deres funktion mm. samt introducer dem til Arduino programmeringsmiljøet.

Anvend nu Arduinoen, til at kontrollere LED'en fra tidl., fremfor at den blot agerer strømforsyning. Dette kan med fordel gøres på en projekter, så eleverne kan følge med på deres egne computere og programmere programmet samtidigt med underviseren. Eksempler på programmering og de følgende opgaver findes i figur 44–46.

Start med at introducere hvad variabler er og forklar herefter hvordan programstrukturen fungerer, sæt dernæst den tilkoblede pin til OUT-PUT i setup, mens den i loop sættes til HIGH. Gennemgå igen forskellen på INPUT og OUTPUT, hvorfor LED'en sættes til OUTPUT, samt forskellen på LOW (oV) og HIGH (5V). Det kan også være en god ide, i samme ombæring kort at komme ind på de mest anvendte datatyper (int, String, boolean), i forbindelse med at den tilkoble pin, gemmes som en int.

#### Mulig analogi på variabler:

En god analogi kan være at sammenligne dem med de flyttekasser mange har stående ude i garagen, hvorpå der er skrevet en label og en ting er kommet deri. Nu kan den ønskede kasse altid findes frem igen og det i lagte tilgås, ligesom det kan udskiftes med noget andet. Forklar at variabler derfor er en metode til at gemme ting i mikrocontrollerens hukommelse, så de senere kan findes frem igen.

Når alle elever har fået LED'en til at lyse, kan denne sættes til at blinke ved at lade den tilkoblede pin veksle mellem HIGH og LOW, men uden at der er indsat et delay. Eleverne vil hurtigt opdage at de ikke kan se nogen forskel på før og nu. Dette er en fin indgangsvinkel til at lade dem opleve hvor hurtigt en processor kan udføre beregninger og handlinger baseret derpå. Introducer delay funktionen, men spørg først eleverne hvor denne

einen neuen ersetzt werden. Variablen sind somit eine Methode zum Ablegen von Informationen im Speicher des Mikrocontrollers, damit sie später abgerufen werden können.

Nachdem die Lernenden die LED zum Leuchten gebracht haben, können sie sie blinken lassen, indem die Einstellung für den angeschlossenen Pin zwischen HIGH und LOW alterniert, ohne dass jedoch zunächst eine Verzögerung eingebaut ist. Der Unterschied zwischen dem einfachen Leuchten und dem Blinken wird für die Lernenden kaum erkennbar sein. Sie können hier aber sehen wie schnell ein Prozessor Berechnungen anstellen und darauf basierende Aktionen ausführen kann. Nun kann die Verzögerungsfunktion (delay) eingeführt werden, wobei die Schülerinnen und Schüler zunächst selbst einschätzen sollen, an welcher Stelle des Programmcodes diese eingefügt werden soll. In vielen Fällen lautet die Antwort, dass es zwischen dem HIGH- und LOW-Setzen des Pins sein sollte. Infolge dieser Antwort bietet sich eine Übung an, da nach wie vor wie o.g. kein signifikanter Unterschied erkannt wird. Ein\*e Lernende\*r kann gebeten werden, in Abständen von ca. 2 Sekunden die Beleuchtung ein- und auszuschalten. Lassen Sie die anderen Lernenden genau verbalisieren, was sie sehen: Licht einschalten, warten, Licht ausschalten, warten, Licht einschalten usw. Bitten Sie die Lernenden nun, den Code des Programms nachzuvollziehen: Einschalten, Warten, Ausschalten, Warten, Einschalten usw. Hier werden sie höchstwahrscheinlich schnell feststellen, dass bei Einsatz einer Schleife (loop) keine Wartezeit zwischen dem Übergang von der letzten zur ersten Programmzeile der Schleife auftritt, weshalb auch hier eine Verzögerung explizit eingebaut werden muss, nachdem der Pin auf LOW gesetzt wurde.

#### Aufgabe für die Lernenden:

- Eksperimente mit der Blinkfrequenz (also den Zeiteinstellungen) interessant ist es hier insbesondere den Zeitintervall in Millisekunden zu finden, bei dem das Auge den Wechsel vom konstanten Leuchten zum Blinken der LED wahrnehmen kann.
- Experiment mit zwei LEDs, die erst gleichzeitig und dann abwechselnd blinken.

Durch Anpassen der Größenverhältnisse der eingebauten Delays kann der Eindruck entstehen, dass die LED nur beispielsweise halb so stark leuchtet. Durch das Experimentieren kann auch verdeutlicht werden, welche Unterschiede es zwischen digitalen, analogen und PWM-Signalen (Pulse Width Modulation, deutsch: Pulsweitenmodulation) gibt.

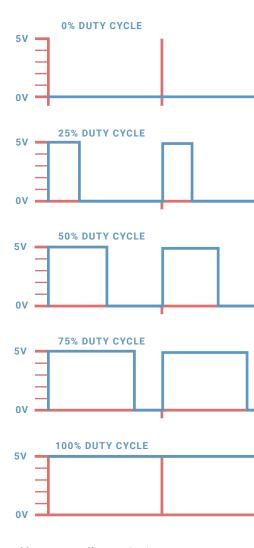
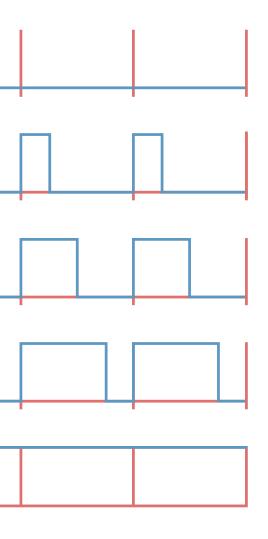


Abb. 43: Darstellung, wie ein PWM-Signal ein Analogsignal simuliert

Fig. 43: PWM-signal simulerer et analogt-signal



skal indsættes. I mange tilfælde vil svaret blive, at det er mellem pin'en sættes til HIGH og LOW. Hvis dette er tilfældet, leder det videre til endnu en god øvelse for eleverne, for som før vil de igen ikke se nogen nævneværdig forskel som effekt heraf. En elev kan med fordel bedes om at blinke med lyset i klasseværelset i intervaller af ca. 2 sekunder, samt forklare hvad denne gør, forud for hver handling: Tænd, Vent, Sluk, Vent, Tænd osv. Bed derefter eleverne gennemgå koden i programmet: Tænd, Vent, Sluk, Tænd osv. Her vil de med stor sandsynlighed hurtigt opdage at ved anvendelsen af et loop, er der ingen ventetid mellem overgangen fra den sidste linje, til den første linje, hvorfor der ligeledes skal indsættes et delay efter at pin'en sætte til LOW.

#### Opgaver til eleverne:

- Eksperimenter med blink hastigheden. Hvor ligger overgangen i millisekunder, fra at øjet kan registrere at denne blinker og til at det ligner at den er tændt konstant.
- Eksperimenter med at få to LED'er til at blinke enten samtidigt, eller skiftevis.

Vis hvordan der ved at justere på størrelsesforholdene på de indsatte delays, kan simuleres hvordan LED'en eks. ser ud som om den tændt på halv styrke. Dette leder over til en samtale med eleverne om forskellen på digitale, analoge og PWM (Pulse-width Modulation) signaler.

Gennemgå forskellene på digitale og analoge signaler og forklar samtidigt eleverne at Arduino'en kun kan læse analoge signaler, men ikke generere dem. Arduinoen anvender i stedet et simuleret analogt signal, kaldet PWM. Forklar dem at et PWM-signal basalt set det samme, som det de selv konstruere lige før, gennem manipulationen af størrelsesforholdet på de anvendte delays (se Figur 43).

Erläutern und diskutieren Sie an dieser Stelle die Unterschiede zwischen digitalen und analogen Signalen und erklären Sie gleichzeitig, dass der Arduino analoge Signale nur lesen, aber nicht erzeugen kann. Stattdessen verwendet der Arduino ein simuliertes analoges Signal namens PWM. Das PWM-Signal ist im Grunde das gleiche wie das, was die Lernenden kurz zuvor durch das Manipulieren des Größenverhältnisses der verwendeten Delays (vgl. Abb. 43) selbst konstruiert haben.

#### Beispiele für die Arduino-Programmierung:

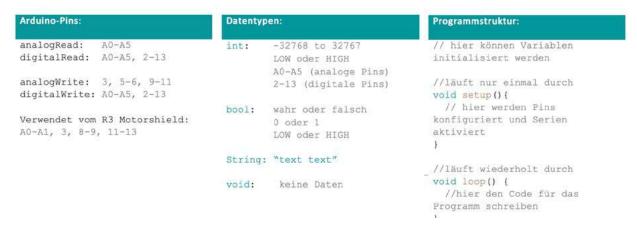


Abb. 44: Übersicht über die Arduino-Pins, die meistgenutzten Datentypen und die Programmstruktur

```
Variable (mit einem Wert):

//Datentyp Name = Wert
int varName = 13;

//Datentyp Name = Wert
bool varName = true;

// Datentyp Name = Wert
String varName = "Hello!";

Delay:

void loop() {
    //pausiert für 1000
    Millisekunden
    delay(1000);
}

// Datentyp Name = Wert
String varName = "Hello!";
```

Abb. 45: Übersicht über die Art der Nutzung von Variables und Delays

#### Eksempler på programmering med Arduino:

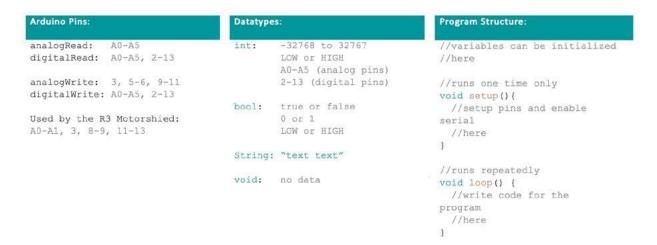


Fig. 44: Oversigt over Arduinoens pins, samt de mest brugte datatyper og Arduino programstrukturen

```
Variable (with a value):

//datatype name = value
int varName = 13;

//datatype name = value
bool varName = true;

//datatype name = value
String varName = "Hello!";

Delay:

void loop() {
    //pause for 1000 milliseconds
    delay(1000);
}

//datatype name = value
String varName = "Hello!";
```

Fig. 45: Oversigt over hvordan variabler og delays kan anvendes

```
LED:
                                             5 mm rote LED (Größe Widerstand: 150Ω)
int LEDPin = 2;
                                 //der verwendete
Pin
void setup() {
  pinMode (LEDPin, OUTPUT);
                                 //Pin als Output
einstellen
}
void loop() {
                                                           GND: kürzeres Metall
  //LED leuchtet nicht (LOW) oder leuchtet
                                                           PIN: längeres Metall
(HIGH) *
  digitalWrite (LEDPin, HIGH);
                                                           *PIN: 2, 4-7, 10, A2-A5
                                                            **PIN: 5, 6, 10
  //kontinuierlich von aus (0) zu ein (255)
wechseln **
  analogWrite (LEDPin, 180);
```

Abb. 46: Beispiel für den Einsatz einer LED

#### Phase 3 – Einführung in das Thema Sensoren (Schalter) (20 Min.):

Stellen Sie den Schalter als Bauteil vor und erläutern Sie seine Funktion: im geöffneten Zustand ist der Stromkreis geöffnet und durch Betätigen des Schalters lässt sich der Kreis schließen, so dass Strom fließen kann.

Stellen Sie damit einen Schaltkreis auf dem Steckbrett her und schließen Sie ihn an den Arduino an. Lassen Sie dies parallel von den Lernenden bauen. Erläutern Sie kurz, dass in diesem Zusammenhang einen Widerstand verwenden werden muss, um den Schaltkreis nicht zu "verschmoren". Erklären Sie, dass Schalter auch Sensoren sind und somit im Setup-Teil des Arduino-Codes als INPUT angegeben werden müssen. Für später ist anzumerken, dass der verwendete Widerstand nicht unbedingt erforderlich ist – es können stattdessen die eingebauten Pull-up-Widerstände des Arduino verwendet werden, die durch Ersetzen von "INPUT" durch "INPUT\_PULLUP" aktiviert werden. Für den Moment macht eine Verwendung eines Widerstands jedoch Sinn, da dies eine Einführung in die Verwendung von Spannungsteilern beim Aufbau diverser Sensoren ermöglicht.

Erstellen Sie ein neues Programm, in dem der Wert des Schalters gelesen wird. Verzichten Sie zunächst auf die gesamte serielle Kommunikation (Serial.begin (9600) und Serial.println (switchValue)), so dass nur analogRead (switchPin) vorhanden ist. Sprechen Sie dann mit den Lernenden darüber, wie ein Einblick in das "Denken" des Mikrocontrollers gewonnen werden kann. Es ist hier durchaus möglich, dass der Eine oder die Andere möglicherweise aus früherem Unterricht oder eigenständigen Projekten in der Freizeit mit Printstatements vertraut ist.

Stellen Sie dann den seriellen Monitor sowie die Art und Weise vor, wie die serielle Kommunikation aktiviert und sog. Print-Statements eingesetzt werden, um Werte von Variablen und eigene Textmitteilungen an den Monitor auszugeben (print, deutsch zu drucken). Erklären Sie gleichzeitig, wie Print-Statements zur Fehlerbehebung (debugging) genutzt werden können -

```
LED:
                                                  5mm Red LED (resistor size: 150Ω)
int LEDPin = 2;
                                 //the used pin
void setup() {
  pinMode (LEDPin, OUTPUT);
                                 //set pin as
OUTPUT
1
void loop() {
  //turn the LED off (LOW) or on (HIGH) *
                                                           GND: Short leg
  digitalWrite (LEDPin, HIGH);
                                                           PIN: Long leg
  //turn it gradually from off (0) to on (255)
                                                           *PIN: 2, 4-7, 10, A2-A5
                                                           **PIN: 5, 6, 10
  analogWrite(LEDPin, 180);
```

Fig. 46: Eksempel på hvordan en LED kan anvendes

#### 3. Fase – Introduktion til sensorer (kontakt) (25 min):

Introducer kontakten og gennemgå for eleverne hvordan den virker og kan ses som en knækket ledning, der når den trykkes sammen, skaber forbindelse igennem sig.

Lav et kredsløb med den på breadboardet og tilslut det til Arduinoen, lad samtidigt eleverne kopiere dette. Gennemgå kort hvorfor der det er nødvendigt at anvende en modstand i forbindelse hermed, for ikke at brænde kredsløbet af. Forklar at fordi kontakter er sensorer, skal de i setup delen af Arduino koden, angives som INPUT. Anvendelsen af INPUT, samt en modstand ifm. kontakten, giver en naturlig introduktion til spændingsdelere og konstruktionen af egne sensorer (mere om det i de følgende afsnit). Til senere forløb er det værd at bemærke at den anvendte modstand ikke er strengt nødvendig, fordi der i stedet kan anvende Arduinoens indbyggede pull-up modstande, hvilke aktiveres ved at udskifte "INPUT" med "INPUT\_PULLUP".

Lav et nyt program, hvori kontaktens værdi aflæses. Udelad i første omgang alt seriel kommunikation (Serial.begin(9600); og Serial.println(switchValue)), sådan at der kun er analogRead(switchPin). Tag herefter en snak med eleverne om, hvordan man kan få indsigt i hvad mikrocontrolleren "tænker", for selvom der er indsat en analogRead kommando, kan de pt. ikke se værdien noget sted. Det er ikke utænkeligt at en eller flere af dem, evt. fra tidl. undervisning eller selvstændige projekter hjemmefra, er bekendt med print-statements.

Introducer herefter serial monitoren, samt hvordan seriel kommunikation aktiveres og print statements anvendes, til at få printet variablers værdier, samt egne tekstbeskeder til monitoren. Forklar hvordan print statements udgør et vigtigt redskab ifm. debugging, fordi dette er den mest umiddelbare metode til at få indsigt i hvilke dele af programmet der udføres, samt hvilke værdier der opereres med. Ved anvendelsen af en sensor (kontakt), er to af kravene til en robot, hermed opfyldt. Introducer derfor og gennemgå hvordan betingelser kan anvendes til at lade mikrocontrolleren tænde/slukke for LED'en, alt efter om kontakten er trykket ned eller ej, hvorved det tredje og sidste krav er blevet opfyldt.

sie stellen eine unmittelbare Möglichkeit dar, nachzuvollziehen, welche Teile des Programms ausgeführt werden und mit welchen Werten operiert wird.

Erklären Sie den Lernenden, dass mit dem Einsatz des Schalters als Sensor eine weitere Anforderung für die Definition eines Roboters erfüllt ist, nämlich die Wahrnehmung der Welt. Um die dritte Anforderung zu erfüllen, muss der Arduino 'lediglich' noch dazu gebracht werden, daraus abgeleitet Entscheidungen zu treffen. Stellen Sie somit den Aufbau eines Programmes vor und legen Sie dar, wie ein Ein- und Ausschalten der LED durch den Mikrocontroller herbeigeführt werden kann, je nachdem, ob der Schalter gedrückt ist oder nicht.

#### Aufgabe:

Experimentieren Sie mit den Lernenden, welche Größe eine Verzögerung möglicherweise haben sollte, bevor die Schwingungen herausgefiltert werden, damit der
Schalter nicht öfter als tatsächlich gedrückt gedrückt wird.

#### Beispiele für die Programmierung:

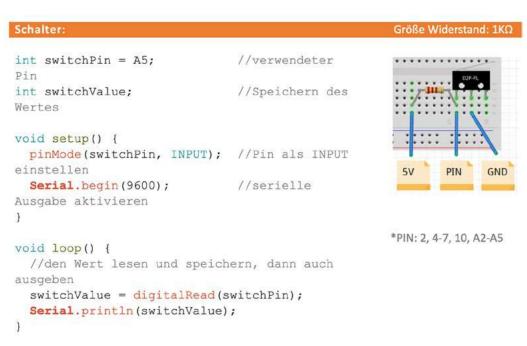


Abb. 47: Beispiel für den Einsatz eines Schalters

Alt efter typen af switch, men den godt stå og svinge et par gange mellem at være oppe og nede, når denne skifter stadie. Det kan derfor ofte være en god ide at indsætte et ganske kort delay, på nogle få millisekunder.

#### Opgave til eleverne:

 Eksperimenter med hvilken størrelse et delay potentielt skal have, førend svingningerne filtreres fra, så kontakten ikke registreres som værende trykket ned flere gange end den reelt er.

#### Eksempler på programmering:

```
int switchPin = A5;
                             //the used pin
                            //to store the
int switchValue;
value
void setup() {
  pinMode(switchPin, INPUT); //set pin as
INPUT
  Serial.begin (9600);
                               //enable
serial
void loop() {
                                                    *PIN: 2, 4-7, 10, A2-A5
  //read and store the value, then prints it
  switchValue = digitalRead(switchPin);
  Serial.println(switchValue);
```

Fig. 47: Eksempel på hvordan en kontakt kan anvendes

```
Bedingung (Struktur):
                                         Bedingung (Operatoren):
void loop() {
                                             gleich
  if (x == y) {
                                             nicht gleich
    //falls Statement wahr,
                                        != größer
Code ausführen
                                             kleiner
                                             größer gleich
  else if (x < 4 && y < 4) {
                                             kleiner gleich
    //nur falls Statement wahr,
diesen Code ausführen
                                             und
                                            oder
  else {
    //sonst diesen Code
ausführen
                                         8.3
                                        11
*es kann nur einen "falls"-Befehl geben
*es kann mehrere "nur falls"-Befehle geben
*es kann nur einen "sonst"-Befehl geben
```

Abb. 48: Übersicht über die Nutzung von Bedingungen

#### Phase 4 – Erläutern von Spannungsteilern und Sensoren (LDR) (45 Min.):

Spannungsteiler bilden die Basis vieler Sensoren. Führen Sie daher in das Thema Spannungsteiler ein und erläutern Sie, wie die Widerstände interagieren.

Mögliche Analogie: Die Spannung kann als eine Schale mit fünf Stück Kuchen gesehen werden (ein Stück Kuchen für jedes der 5 Volt). Die beiden Widerstände werden nun zwei Personen gleichgesetzt. Hinzu kommt die Anforderung, dass alle Kuchenstücke gegessen sein müssen, nachdem die Schale an der letzten Person vorbeigekommen ist. Es ist wichtig zu betonen, dass es keine Rolle spielt, wie hungrig die Personen sind, sondern dass lediglich alle fünf Kuchenstücke gegessen werden müssen.

Entscheidend ist vielmehr, wie hungrig die Personen im Verhältnis zueinander sind. Wenn sie gleich hungrig sind, teilen sie sich die Kuchenstücke gleichmäßig auf. Aber wenn die erste Person im Vergleich zur zweiten sehr großen Hunger hat, isst sie die meisten Kuchenstücke, während die zweite weniger Kuchenstücke isst.

Nachdem die Kuchenschale an der ersten Person vorbeigekommen ist, lässt sich feststellen, wie viele Kuchenstücke noch übrig sind. Dies ist der Indikator für das Verhältnis zwischen dem Hunger der beiden Personen, da bekannt ist, dass die andere Person auf jeden Fall den Rest essen muss.

Nach der Einführung und Erläuterung der Funktionsweise von Spannungsteilern können die folgenden Übungen durchgenommen und ausgeführt werden (der Schalter wird im Stromkreis durch einen Widerstand ersetzt – danach ist der Aufbau der gleiche wie zuvor). Denken Sie daran, zu wiederholen, dass analoge Werte des Arduino als Zahl zwischen o (o V) und 1023 (5 V) gelesen werden.

```
Conditional (Structure):
                                         Conditional (Operators):
void loop() {
                                            equal
  if (x == y) {
                                             not equal
    //if yes, run code
                                             larger
                                             smaller
  else if (x < 4 && y < 4) {
                                            larger or equal
    //if yes, run this instead
                                        <
                                             smaller or
                                             equal
  else {
    //else, run this instead
                                             and
                                             OI
}
                                         88
*there can only be one "if"
*there can be multiple "else if"
                                         1.1
"there can only be one "else"
```

Fig. 48: Oversigt over hvordan betingelser kan anvendes

#### 4. Fase - Forklaring af spændingsdelere og Sensorer (LDR) (65 min):

Spændingsdelere danner grundlaget for mange sensorer og det er ligeledes en nem måde at konstruere vores egne på. Introducer derfor spændingsdelere og hvordan forholdene mellem modstandene spiller sammen.

Mulig analogi på variabler: En god analogi til formålet, er at spændingen kan ses som et fad med fem kager (en kage for hver de 5 volt). De to modstande kan derfor sammenlignes med to personer. Tilføj kravet at alle kagerne skal være spist, efter at fadet har passeret den sidste person. Det er vigtigt at understrege at det derfor ikke afgørende hvor sultne personerne er, for alle fem kager skal spises uanset hvad. Forklar at det der derimod er afgørende, er forholdet mellem hvor sultne de er. Hvis de er lige sultne i forhold til hinanden, deles kagerne ligeligt mellem dem. Mens hvis den første person er meget sulten i forhold til den anden, vil han derfor spise flest kager, mens den anden ikke spiser ret mange og omvendt.

Efter at fadet med kager har passeret den første person, kan man nu se hvor mange kager der er tilbage på fadet, dette er vores indikator på forholdet mellem de to personernes sult, fordi man ved at den anden person uanset hvad, skal spise de resterende.

Efter at have introduceret og forklaret hvordan spændingsdelere fungerer, kan nedenstående øvelser herefter gennemgås og laves (kontakten tages blot ud af kredsløbet og erstattes herefter af en modstand – derefter er opstillingen den samme som fra tidl.). Repetere gerne at analoge værdier af Arduinoen læses som et tal mellem o (oV) og 1023 (5V).

Widerstand 1:	Widerstand 2:	Abgelesener Wert:	Wert in Volt:	
1ΚΩ	1ΚΩ			
56οΩ	56οΩ			
820Ω	18οΩ			
180Ω	820Ω			
1ΚΩ	Unendlich (Schalter oben)			
1ΚΩ	Keiner (Schalter unten)			

Wenn die Übungen absolviert sind, kann man sich der Formel für die Spannungsteiler widmen, wonach weitere Aufbauten zunächst berechnet und dann in der Praxis erprobt werden können. Erklären Sie den Lernenden, dass die soeben konstruierten Spannungsteiler keine Sensoren sind, auch wenn viele Sensoren auf Spannungsteilern basieren. Das liegt daran, dass es sich hier um statische Spannungsteiler handelt und daher keine Änderungen in der Umgebung erkannt werden soll bzw. kann.

$$U1 = U * \frac{R1}{R1 + R2}$$

Erläutern Sie den Lernenden nun, dass zur Konstruktion ihres eigenen Sensors anstelle von statischen Widerständen ein dynamischer Widerstand verwendet werden kann, der seinen Innenwiderstand im Takt der Änderungen seiner Umgebung ändert. Stellen Sie daher den lichtabhängigen Widerstand (LDR, Light Dependent Resistor, deutsch: Fotowiderstand) vor und erläutern Sie, dass sein Innenwiderstand dynamisch variiert, je nachdem, wie viel Licht auf ihn fällt. Daher soll der LDR verwendet werden, um einen ersten Sensor zu konstruieren.

Setzen Sie den LDR in den vorgesehenen Spannungsteiler ein, lesen Sie den Wert des Spannungsteilers ab und geben Sie ihn an den Monitor aus. Da der Spannungsabfall über dem einzelnen Widerstand im Spannungsteiler jetzt nicht mehr konstant ist, repräsentiert der abgelesene Wert jetzt die Änderungen in der Umgebung, in der sich der Sensor befindet (Lichtstärke).

#### Diese Aufgaben können Sie als Impulse an die Lernenden geben:

- Erstellen Sie ein System, das eine LED bei Dunkelheit ein-, bei Helligkeit jedoch wieder ausschaltet. Dieses System wird z. B. bei Straßenlaternen genutzt.
- Erstellen Sie ein System, das die Helligkeit der LED schrittweise erhöht bzw. verringert, wenn sich die Lichtverhältnisse ändern. Dies ist eine gute Übung zum Einüben des Verhältnisses von abgelesenen analogen Signalen (0−1023) zu PWM-Signalen (0−255). Dieses System lässt sich auch auf Smart Homes beziehen, bei denen beispielsweise ein konstantes Lichtniveau im Wohnzimmer angestrebt wird.

Modstand 1:	Modstand 2:	Aflæst værdi (Arduino):	Værdi i volt:
1ΚΩ	1ΚΩ		
560Ω	56οΩ		
820Ω	18οΩ		
180Ω	820Ω		
1ΚΩ	Uendelig (kontakt oppe)		
1ΚΩ	Ingen (kontakt nede)		

Med øvelserne overstået, kan formlen for spændingsdelere gennemgås, hvorefter yderligere opstillinger kan beregnes først og derefter efterprøves i praksis.

Forklar eleverne, at selvom mange sensorer er baseret på spændingsdelere, så er de ovenstående der lige har været konstrueret, ikke sensorer. Dette fordi de er statiske af natur og altså

$$U1 = U * \frac{R1}{R1 + R2}$$

ikke registrerer ændringer i det miljø de befinder sig i. Hvis der i stedet for statiske modstande, havde været anvendt dynamiske der ændrer deres interne modstand i takt med ændringer i de omgivelser de befinder sig, havde dette opfyldt kravene til en sensor. Introducer derfor LDR'en (Light Dependent Resistor) og forklar at dennes interne modstand netop er dynamisk varierende alt efter hvor meget lys der falder på den.

Indsæt LDR'en i spændingsdeleren og aflæs herefter og print værdien af spændingsdeleren til monitoren. Fordi at spændingsfaldet over den enkelte modstand i spændingsdeleren ikke længere er konstant, repræsenterer den aflæste værdi nu de ændringer i miljøet hvor sensoren befinder sig (lys-niveauet).

#### Opgaver til eleverne:

- Konstruer et system lignende en gadelampe, så der tændes for en LED når det er mørkt og slukkes igen når det er lyst.
- Konstruer et system inspireret af smarthomes, hvor der gennem anvendelsen af betingelser, gradvist skruer op/ned for lysstyrken på LED'en, i takt med at lyset ændrer sig, så man opnår et konstant lys-niveau inde i stuen.

```
LDR (Light Dependent
                                                      Größe Widerstand: 1ΚΩ
int LDRPin = A5;
                                //verwendeter
Pin
int LDRValue;
                                //Speichern des
Wertes
void setup() {
 pinMode(LDRPin, INPUT);
                               //Pin als INPUT
einstellen
 Serial.begin(9600);
                               //serielle
                                                        5٧
                                                              PIN
                                                                   GND
Ausgabe aktivieren
}
void loop() {
                                                      *PIN: A2-A5
  // den Wert lesen und speichern, dann auch
ausgeben
  LDRValue = analogRead(LDRPin);
  Serial.println(LDRValue);
}
```

Abb. 49: Beispiel für den Einsatz eines LDRs

```
LDR (Light Dependent Resistor /
                                                           Resistor size: 1KΩ
int LDRPin = A5;
                                //the used pin
int LDRValue;
                                //to store the
value
void setup() {
                               //set pin as
  pinMode (LDRPin, INPUT);
INPUT
 Serial.begin(9600);
                               //enable
                                                              PIN
                                                                    GND
serial
}
void loop() {
                                                      *PIN: A2-A5
 //read and store the value, then prints it
 LDRValue = analogRead(LDRPin);
 Serial.println(LDRValue);
}
```

Fig 49: Eksempel på hvordan en LDR kan anvendes

### Phase 5 – Erweiterung um zusätzliche Sensoren (Potentiometer, IR-Sensor) (45 Min.):

Führen Sie die Lernenden in das Potentiometer bzw. den IR-Sensor ein und erläutern Sie, wie diese aufgebaut sind und wie sie funktionieren. Gleichzeitig kann erklärt werden, dass auch diese wie der bereits konstruierte Lichtsensor (LDR) um einen Spannungsteiler herum aufgebaut sind. Deshalb können die Bauteile untereinander ausgetauscht werden, während der Code und der Rest des Setups unverändert bleiben.

Erklären Sie, dass das Potentiometer einen stufenlosen Übergang im Verhältnis der Werte der beiden Widerstände aufweist, je nachdem, wo der Regler mit dem kreisförmigen Widerstand interagiert.

Der IR-Sensor hingegen ist mit einer IR-LED sowie einem IR-empfindlichen Fotowiderstand versehen – das ist der IR-empfindliche Fotowiderstand, der zusammen mit einem statischen Widerstand den Spannungsteiler bildet – dessen Widerstand variiert, je nachdem, von wieviel Infrarotlicht er getroffen wird. Die Schüler werden daher auch feststellen, dass er, wenn sie ihn von der Oberfläche weg auf Abstand halten, "schwarz" liest (oder vielmehr einen Mangel an Reflexion feststellt).

#### Aufgaben, Potentiometer:

- Erstellen Sie ein System, das die LED nach und nach in fünf Schritten einschaltet, wenn das Potentiometer von links nach rechts gedreht wird.
- Erstellen Sie ein System, das die LED nach und nach stufenlos einschaltet, wenn das Potentiometer von links nach rechts gedreht wird. Den Bezug haben wir hier z. B. zu einer Stereoanlage, in der häufig Potentiometer verbaut sind – oder auch zum heimischen Beleuchtungsdimmer.

#### Aufgaben, IR-Sensor:

- Lassen Sie die Schüler experimentieren, in welcher Entfernung der IR-Sensor den Unterschied zwischen weißem und schwarzem Papier am besten erfasst (ca. 0,5 bis 0,7 cm über der Oberfläche).
- Erstellen Sie ein System, in dem eine LED aufleuchtet, wenn der IR-Sensor einen niedrigen Wert liest (massive Blendung durch weißes Papier), jedoch erlischt, wenn ein hoher Wert gelesen wird (minimale Blendung durch schwarzes Papier).
- Erstellen Sie ein System, bei dem eine LED in fünf Intervallen mehr und mehr aufleuchtet, basierend auf dem Widerschein (der Blendung) auf den IR-Sensor.

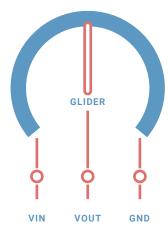
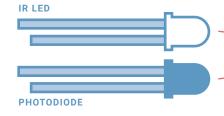


Abb. 50: Aufbau und Funktionsweise eines Potentiometers

Fig. 50: Konstruktion af et voltmeter



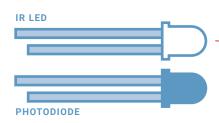


Abb. 51: Abbildung der Funktionsweise eines IR-Sensors

Fig. 51: Illustration af hvordan IR-sensoren fungerer

### 5. Fase – Udvidelse med yderligere sensorer (Potentiometer, IR-Sensor) (60 min):

Introducer eleverne til henholdsvis potentiometeret og forklar at ligesom lyssensoren (LDR), bygger denne også på en spændingsdelerer, hvorfor komponenterne frit kan skiftes rundt, uden at der behøves laves ændringer i koden. Forklar at potentiometeret har en glidende overgang i forholdet mellem de to modstandes værdier, baseret på hvor glideren interagerer med den cirkulære modstand.

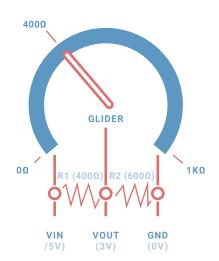
Introducer IR-sensoren og gennemgå hvordan den er opbygget af en IR-LED, samt en IR Fotodiode. IR-sensoren fungerer ved at IR-LED'en udsender et infrarødt lys (dette er usynligt for det menneskelige øje, men kan ofte ses med en kameraet i en mobiltelefon) der herefter i varierende grad reflekteres tilbage til IR Fotodioden afhængig af overfladen. Det er mængden af reflekteret lys der aflæses af sensoren. Eleverne vil derfor også finde, at hvis de holder den væk fra overfalden, så læser den "sort" (eller rettere, den registrerer en mangel på refleksion).

#### Opgaver til eleverne (potentiometer):

- Konstruer et system til lysdæmpning, der gradvist tænder for LED'en, med en fem-trins overgang, når potentiometeret drejes fra venstre mod højre.
- Konstruer et system til lysdæmpning, der gradvist tænder for LED'en, med en flydende overgang, når potentiometeret drejes fra venstre mod højre.

#### Opgaver til eleverne (IR-sensor):

- Eksperimenter med, på hvilken afstand IR-sensoren er bedst til at opfange forskellen mellem hvidt og sort papir.
- Genbrug koden fra potentiometeret og konstruer et system, hvor en LED tænder, hvis IR-sensoren læser en lav værdi (massivt genskin fra hvidt papir), mens den slukker hvis der læses en høj værdi (minimalt genskin fra sort papir).
- Genbrug koden fra potentiometeret og konstruer et system, hvor en LED tænder mere og mere, i fem intervaller, baseret på mængden af genskin på IR-sensoren.





DARK COLORED SURFACE

#### int PotPin = A5; //the used pin int PotValue; //to store the value void setup() { pinMode(PotPin, INPUT); //set pin as INPUT GND Serial.begin(9600); //enable serial } \*PIN: A2-A5 void loop() { //read and store the value, then prints it PotValue = analogRead(PotPin); Serial.println(PotValue); }

Abb. 52: Beispiel für das Ablesen des Potentiometers

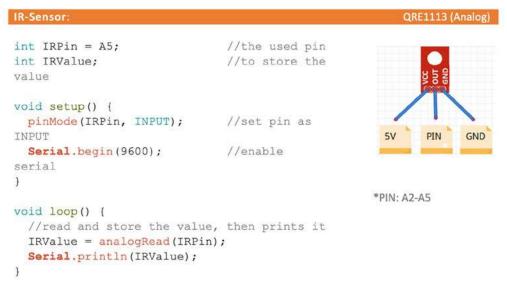


Abb. 53: Beispiel für das Ablesen des IR-Sensors

#### int PotPin = A5; //the used pin int PotValue; //to store the value void setup() { pinMode(PotPin, INPUT); //set pin as GND Serial.begin (9600); //enable serial \*PIN: A2-A5 void loop() { //read and store the value, then prints it PotValue = analogRead (PotPin); Serial.println (PotValue);

Fig. 52: Eksempel på hvordan Potentiometeret kan aflæses

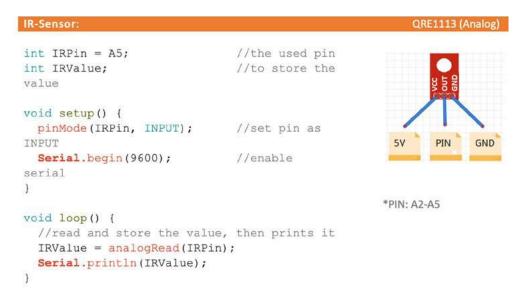


Fig. 53: Eksempel på hvordan IR-Sensoren kan aflæses

#### Phase 6 - Aktoren (Servo- und DC-Motor) (45 Min.):

Führen Sie die Schüler in den Servomotor und damit auch in die Bibliotheken ein und erklären Sie ihnen, dass dies eine Sammlung vorprogrammierter Codes ist, was wir in diesem Fall zur Steuerung des Servomotors anwenden wollen. Lassen Sie die Lernenden den Servomotor auf eine bestimmte Position einstellen und dann versuchen, ihn so einzustellen, dass er sich von einer Seite zur anderen bewegt. Hier ist es wichtig, auf den Einsatz von Delaylicht zu sprechen zu kommen, weil der Servomotor ohne diese Verzögerungen keine Zeit hat, in die angegebenen Position zu gelangen, bevor er das Signal für die nächste Position erhält und umgekehrt.

#### Aufgaben, Servomotor:

- Erstellen Sie ein System, bei dem die Position des Servomotors der des Potentiometers folgt (diesmal ist der Bezug z. B. das Ellbogengelenk eines Exoskeletts, damit weiter gedacht ältere Menschen Dinge heben können, die sie sonst nicht mehr heben könnten).
- Entwerfen Sie ein System zum automatischen Öffnen des Fensters, bei dem der Servomotor beim Betätigen eines Schalters eine Position zwischen o und 90° ermöglicht.

Führen Sie sie sodann in die Motorabschirmung und in den Gleichstrommotor ein. Im gleichen Zug können die eingebauten H-Brücken der Motorabschirmung (sie hat zwei, für jeden Motor eine) ganz kurz erklärt werden. Abbildung 54 zeigt eine vereinfachte Version des Schaltkreises einer H-Brücke. Wichtig sind die vier Transistoren P1, P2 und N1, N2 sowie die beiden Verbindungen dorthin, F1 und F2. Sie sind so eingestellt, dass beim Anlegen einer Spannung an F1 P1 geöffnet und N1 geschlossen wird und umgekehrt, wenn an F1 keine Spannung anliegt. Entsprechend hängen P2 und N2 von der Spannung an F2 ab. Wenn man also 5 V an F1 und o V an F2 anlegt, fließt der Strom in eine Richtung durch den Motor und in die andere Richtung, wenn wir o V an F1 und 5 V an F2 legen. Legt man sowohl an F1 als auch F2 5 V oder o V an, so kann der Strom weder in die eine noch die andere Richtung fließen. Der Grund, warum wir die Motorabschirmung und ihre H-Brücken zur Steuerung der Gleichstrommotoren verwenden, ist, dass wir dadurch einen stärkeren Strom durch die Motoren leiten können, als ihn der Arduino selbst liefern kann. Der Strom, den wir hindurchleiten, kommt direkt vom angeschlossenen Akkusatz (9 V).

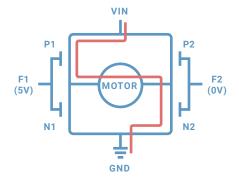
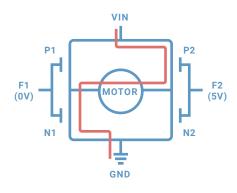


Abb. 54: Vereinfachte Darstellung des Schaltkreises einer H-Brücke

Fig. 54: Forsimplet illustration af kredsløbet i en H-bro

#### 6. Fase – Aktuatorer (Servo- og DC-Motor) (60 min):



Introducer eleverne til servomotoren og derigennem også til biblioteker og forklar dem at dette er en samling af forprogrammeret kode, hvilket i dette tilfælde skal anvendes til at kontrollere servo-motoren. Lad eleverne indstille servomotoren til en given position og lad dvem herefter prøve indstille den til at bevæge sig fra side til side. Her er det vigtigt at komme ind på anvendelsen af delays, foruden hvilke servo motoren ikke har tid til at gå til den angivne position, førend den får signalet om den næste position og vice versa.

#### Opgaver til eleverne:

- Konstruer et system der kan anvendes i et eksoskelet, med det formål at forstærke det menneskelige albueled, så lagerarbejdere ikke slider kroppen gennem tunge løft, ved at lade servomotorens position, følge potentiometerets.
- Konstruer et system til automatisk åbning af vinduer, ved at lave servo motoren veksle mellem at stå i en position af o og 900, når der trykkes på en kontakt.

Introducer eleverne til motorshieldet, samt DC-motoren. I samme ombæring kan motorshieldets indbyggede H-broer (den har to, én til hver motor) forklares ganske kort. På Figur 54 kan der ses en forsimplet udgave af kredsløbet for en H-bro, det vigtige at bide mærke i, er de fire transistorer P1, P2 og N1, N2, samt de to forbindelser dertil F1 og F2. De er sat op således at når der sættes spænding på F1, så åbnes P1, mens N1 lukkes og vice versa når der ingen spænding er på F1. På samme måde afhænger P2 og N2 af spændingen på F2. Ved derfor at sætte 5V til F1 og oV til F2, vil strømmen løbe den ene vej gennem motoren, mens den vil løbe den anden vej hvis der blev sat oV til F1 og 5V til F2. Satte man derimod 5V eller oV til både F1 og F2, vil strømmen ikke kunne løbe nogen af vejene. Forklar at grunden til at motorshieldet og dets H-broer anvendes til at styre DC-motorerne, er at disse kan lede en stærkere strøm gennem motorerne end Arduinoen selv kan forsyne dem med. Den strøm motorshieldet leder gennem DC motorerne, kommer direkte fra "Vin" (voltage in - dvs. den tilsluttede batteripakke (9V)).

#### Aufgaben, DC-motor:

- Bringen Sie den Gleichstrommotor dazu, 5 Sekunden lang abwechselnd im und gegen den Uhrzeigersinn zu laufen.
- Bringen Sie den Gleichstrommotor dazu, seine Drehzahl allmählich zu ändern, wenn das Potentiometer von links nach rechts und umgekehrt gedreht wird.

#### Servomotor: #include <Servo.h> //include library Servo servoMotor; //create new servo object int servoPin = 2; //the used pin void setup() { //attach the used pin to the servo object servoMotor.attach (servoPin); void loop() { \*PIN: 2, 4-7, 10, A2-A5 //turn the servo to a position (value: 0servoMotor.write(90); }

Abb. 55: Beispiel, wie der Servomotor in eine bestimmte Position gebracht werden kann

#### Gleichstrommotor (DC): int motorASpeed = 3; //PWM A int motorADirection = 12; //DIR A void setup() { //set all motor pins to PUTPUT pinMode (motorASpeed, OUTPUT); pinMode (motorADirection, OUTPUT); void loop() { //DIRECTION: Turn clockwise = HIGH, //turn counter-clockwise = LOW digitalWrite (motorADirection, HIGH); \*In Motor B genutzte Pins, siehe Motorab-//SPEED: no speed = 0, full speed = 255 schirmung analogWrite (motorASpeed, 150); }

Abb. 56: Beispiel einer Art der Steuerung des DC-Motors

## Opgaver til eleverne:

- Eksperimenter med at få DC-motoren til at køre skiftevis med og mod uret i intervaller af 5 sekunder, samt forskellige hastigheder.
- Konstruer et system beregnet til ventilation, hvor DC-motoren gradvist ændrer sin hastighed, i takt med at et potentiometer drejes fra venstre mod højre og vice versa

```
Servo-Motor:
#include <Servo.h>
                           //include library
Servo servoMotor;
                           //create new servo
object
int servoPin = 2;
                          //the used pin
void setup() {
  //attach the used pin to the servo object
  servoMotor.attach(servoPin);
                                                                 PIN
}
void loop() {
                                                      *PIN: 2, 4-7, 10, A2-A5
  //turn the servo to a position (value: 0-
179)
  servoMotor.write(90);
}
```

Fig. 55: Eksempel på hvordan servomotoren kan sættes i en given position

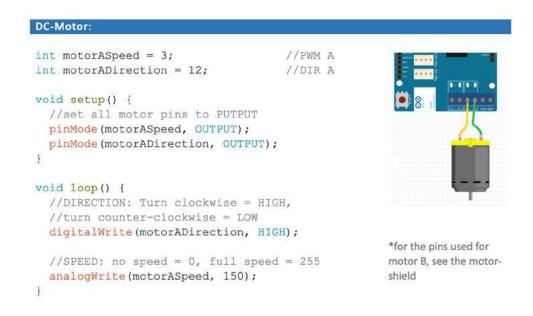


Fig. 56: Eksempel på hvordan DC-motoren kan styres

# Phase 7 - Andere Sensorarten (Abstandssensor, Farbsensor) (30 Min.):

Stellen Sie den Schülern den Abstandssensor vor und erklären Sie, wie wir ihn zum Messen der Entfernung einsetzen können. Es ist so aufgebaut, dass er einen Lautsprecher und ein Mikrofon hat. Der Lautsprecher sendet einen Ping (den wir nicht hören können, weil die Frequenz zu hoch ist), der auf eine Oberfläche trifft und dann zurück zum Mikrofon reflektiert wird. Die Zeit zwischen dem Senden und Zurückprallen des Pings kann verwendet werden, um den Abstand zwischen dem Sensor und dem getroffenen Objekt zu berechnen. Zu diesem Zweck wird die Methode pulsIn() genutzt, die die Zeit aufzeichnet, die ein Ping benötigt, um entweder von HIGH nach LOW oder von LOW nach HIGH zu gelangen. Die Zeit wird in Mikrosekunden (0,000001 s) gemessen. Wir können daher folgende Formel anwenden, um den Abstand in cm zu berechnen:

Pulse In (in Mikrosekunden gemessen, 0.000001s)

$$Schallgeschwindigkeit = \frac{343 \, m}{s} = \frac{34300 \, cm}{0.034 \, cm} = \frac{0.034 \, cm}{ms}$$
 Entfernung zum Objekt in cm : Pulse In (ms) \*  $\frac{0.034 \, cm}{1 \, ms}$  / 2 (vor und zurück)

## Aufgaben für die Lernenden:

- Lassen Sie die Schüler verschiedene Entfernungen im Klassenzimmer messen und diese auf dem Bildschirm ausgeben. Lassen Sie sie ermitteln, in welcher Entfernung es funktioniert und wann es nicht funktioniert (ca. 2 cm-400 cm).
- Lassen Sie die Schüler ein Alarmsystem bauen, das eine LED aufleuchten lässt, wenn ein Objekt zu nahe kommt.

Stellen Sie nun den Farbsensor vor und erklären Sie kurz, wie er funktioniert. Er ist so konzipiert, dass er über einen Filter den reflektierten Wert von rotem, grünem und blauem (RGB-) Licht misst – eine Farbe nach der anderen. Aus den RGB-Werten kann man alle Farben erstellen, wobei es wichtig ist zu wissen, dass alle drei zusammen Weiß ergeben, während das Fehlen aller drei zu Schwarz führt. Im Idealfall verhalten sich die Werte wie in Abbildung 57, in der Realität ist dies jedoch nicht der Fall. Daher erkennen wir bei Einsatz des Sensors zuerst eine Farbe und notieren, welche RGB-Werte der Sensor zurückgibt. Anschließend können wir diese Werte nutzen, um genau unsere Farbe zu definieren (in der library können nur die Farben Rot, Grün, Blau und Gelb definiert werden – siehe Abbildung 59). Weil die RGB-Werte die Menge des reflektierten Lichts darstellen, sind außerdem wechselnde Lichtverhältnisse zu beachten. Hat man beispielsweise in gleißendem Sonnenlicht die Farbe Gelb aufgrund mehrerer gelesener RGB-Werte definiert, so sind diese nicht unbedingt gleich, wenn es Abend ist. Um dies zu vermeiden, empfiehlt es sich daher, das Gelesene abzuschirmen, um die Lichtverhältnisse statischer zu halten.

# 7. Fase - Andre typer af sensorer (Afstandssensor, Farvesensor) (60 min):

Introducer afstandssensoren til eleverne og gennemgå hvordan den kan anvendes til måle afstand fra sensoren til det nærmeste objekt. Gennemgå hvordan den er opbygget over en højtaler og en mikrofon, hvor højtaleren sender et ping ud (ping'et kan ikke høres af mennesker, fordi frekvensen er for høj), der når dette rammer en overflade, reflekteres tilbage til mikrofonen. Tiden det tager mellem at ping'et sendes ud og kommer tilbage, kan herefter bruges til at udregne distancen mellem sensoren og det ramte objekt. Til det formål anvendes metoden pulseln(), hvilken registrerer tiden det tager fra at en pin enten går fra HIGH til LOW, eller LOW til HIGH, tiden måles i mikrosekunder (0.000001s). Følgende formel kan derfor anvendes, til at udregne afstanden i centimer:

Pulse In (måles i mikrosekunder, 0.000001s)

$$Lydens\ hastighed = \frac{343m}{s} = \frac{34300cm}{0.034cm} = \frac{0.034cm}{ms}$$
 Afstand til objektet i cm : Pulse In (ms) \*  $\frac{0.034cm}{1ms}$  / 2 (frem og tilbage)

# Opgaver til eleverne:

- Eksperimenter med at måle forskellige afstande inde i klasseværelse og udskrive disse til skærmen. Find den mindste og længste afstand som sensoren virker på.
- Konstruer et alarmsystem til en industrirobot, der advarer medarbejderne om at de er for tæt på maskinerne, ved at lade en LED lyse op, når afstandssensoren registrerer at der er en person indenfor 2 meters afstand.

Introducer farvesensoren og forklar kort hvordan denne virker. Den er opbygget således at den gennem et filter, måler den reflekterede værdi af henholdsvis Rød, Grøn og Blå (RGB) lys, én farve ad gangen. Ud fra RGB-værdierne, kan man skabe alle farver, men det er vigtigt at vide, at de alle tre tilsammen giver hvid, mens fraværet af alle tre resulterer i sort. Ideelt set, vil værdierne opføre sig på Figur 57, i virkeligheden er dette dog ikke tilfældet. Når sensoren anvendes, registrerer den først en farve og nedskriver hvilke RGB-værdier sensoren returnerer, hvorefter disse værdier kan anvendes til at definere en given farve (i det udviklede bibliotek, er det kun farverne rød, grøn, blå og gul, der kan defineres – se figur 59). Desuden skal man grundet at RGB-værdierne repræsenterer mængden af reflekteret lys, være opmærksom på vekslende lysforhold. Er eks. farven gul blevet defineret i høj solskin, ud fra et sæt af læste RGB-værdierne, vil disse altså ikke nødvendigvis være de samme når det er blevet aften. For at komme rundt om dette, er det derfor ofte en god ide at afskærme det man læser, for derved i større grad at holde lysforholdene statiske.

# Aufgaben zum Farbsensor:

- Lassen Sie die Schüler mit dem Messen und Definieren der Farben Rot, Grün, Blau, Gelb, Weiß und Schwarz experimentieren. Welche farblichen RGB-Werte unterscheiden sich am meisten voneinander, und liegen einige näher beieinander als andere? (Mischfarben wie Gelb sind möglicherweise schwieriger zu definieren, da sie nicht nur aus einer einzigen Farbe bestehen.)
- Experimentieren Sie mit den Differenzen der Farbwerte in Abhängigkeit von den Lichtverhältnissen.
- Lassen Sie die Schüler eine Konstruktion erstellen, die Bedingungen nutzt, um die registrierten Farben auf dem Bildschirm auszugeben.

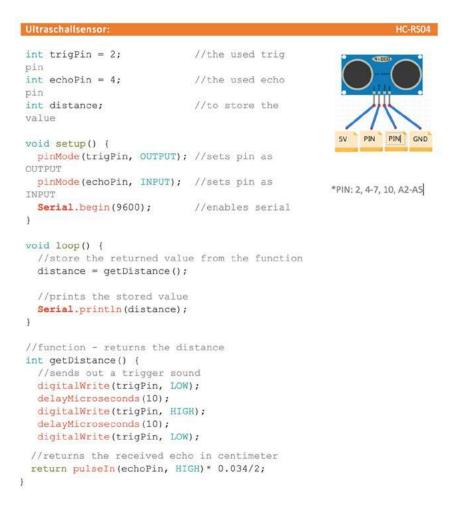


Abb. 58: Beispiel für einen Einsatz des Abstandssensors

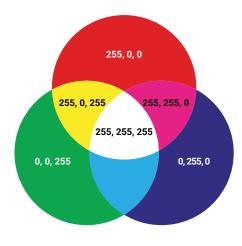


Abb. 57: Übersicht über die idealen RGB-Werte

Fig. 57: Oversigt over de Ideelle RGB-værdier

## Opgaver til eleverne:

- Eksperimentere med at måle og definere farverne Rød, Grøn, Blå, Gul, Hvid og Sort. Hvilke farvers RGB-værdier adskiller sig mest fra hinanden og er der nogle der ligger tættere end andre?
- Eksperimenter med hvilken forskel i værdi farverne har, alt efter hvilke lysforhold der er gældende.
- Konstruer et system, der anvender betingelser til at udskrive de registrerede farver til skærmen.

```
Ultra-Sound Sensor:
int trigPin = 2;
                             //the used trig
pin
int echoPin = 4;
                             //the used echo
pin
int distance;
                             //to store the
value
                                                              PIN GND
                                                          PIN
void setup() {
  pinMode(trigPin, OUTPUT); //sets pin as
  pinMode(echoPin, INPUT); //sets pin as
                                                     *PIN: 2, 4-7, 10, A2-A5
  Serial.begin(9600);
                             //enables serial
1
void loop() {
  //store the returned value from the function
  distance = getDistance();
  //prints the stored value
  Serial.println(distance);
//function - returns the distance
int getDistance() {
  //sends out a trigger sound
  digitalWrite(trigPin, LOW);
  delayMicroseconds (10);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds (10);
  digitalWrite(trigPin, LOW);
  //returns the received echo in centimeter
  return pulseIn(echoPin, HIGH) * 0.034/2;
```

Fig. 58: Eksempel på hvordan afstandssensoren kan anvendes

Farbsensor: TCS3200

```
//includes the library
 #include <Color.h>
 //the used pins
 int S0 = 2;
 int S1 = 4;
int S2 = 5;
int S3 = 6;
int OUT = 7;
 //creates a new color-sensor
 object
 Color color(SO, S1, S2, S3, OUT);
 //used to store the color-value
 String colorValue = "";
 void setup() {
  setColors();
  Serial.begin (9600); //enables serial
 }
 void loop() {
  //prints the R(ed), G(reen), B(lue) and W(hite) values from the
  //write down these values for each color you want to identify
  //(red, green, blue, yellow) and insert them into the
  //setColors function.
  Serial.println(color.getRGBValues());
  //gets the color and prints it
  colorValue = color.getColor();
  Serial.println(colorValue);
}
//Defines the Colors
void setColors() {
  //color.setRed(R, G, B, W);
  color.setRed(40, 104, 22, 81);
  color.setBlue(46, 36, 14, 56);
  color.setGreen(89, 56, 16, 33);
  color.setYellow(29, 36, 12, 55);
}
```

READ ME NOTE: In order to enable the inclusion of the Color library, copy paste the entire "Color Library" folder, into the "Documents -> Arduino -> Libraries" folder.

Abb. 59: Beispiel für einen Einsatz des Farbsensors

Color-Sensor: TCS3200

```
//includes the library
 #include <Color.h>
//the used pins
int SO = 2;
int S1 = 4;
int S2 = 5;
int S3 = 6;
int OUT = 7;
//creates a new color-sensor
object
Color color(SO, S1, S2, S3, OUT);
 //used to store the color-value
String colorValue = "";
void setup() {
  setColors();
  Serial.begin(9600); //enables serial
 }
void loop() {
  //prints the R(ed), G(reen), B(lue) and W(hite) values from the
  //sensor.
  //write down these values for each color you want to identify
  //(red, green, blue, yellow) and insert them into the
  //setColors function.
  Serial.println(color.getRGBValues());
  //gets the color and prints it
 colorValue = color.getColor();
 Serial.println(colorValue);
}
//Defines the Colors
void setColors() {
  //color.setRed(R, G, B, W);
 color.setRed(40, 104, 22, 81);
 color.setBlue(46, 36, 14, 56);
 color.setGreen(89, 56, 16, 33);
  color.setYellow(29, 36, 12, 55);
}
```

READ ME NOTE: In order to enable the inclusion of the Color library, copy paste the entire "Color Library" folder, into the "Documents -> Arduino -> Libraries" folder.

Fig. 59: Eksempel på hvordan farvesensoren kan anvendes

# Phase 8 – Einführung in weitere Programmierungsgrundsätze (Funktionen und While-Schleifen) (20 Min.):

Für den Einsatz des Abstandssensors und des Farbsensors wurde für beide eine Funktion angewandt. Führen Sie die Schüler daher in das Thema Funktionen ein und erklären Sie deren Arbeitsweise, indem Sie einen Codeabschnitt der dann immer wieder aufgerufen werden kann. Dies erleichtert sowohl die Anwendung (wir müssen den Code nur an einer Stelle korrigieren und nicht an vielen Stellen) als auch das Lesen (die Funktion kann viele Liniencodes unter einem einfachen und aussagekräftigen Namen abgrenzen).

# Aufgaben zum Thema Funktionen:

- Erstellen Sie eine Funktion namens "printHello", die bei Aufruf "Hello" ausgibt.
- Erstellen Sie eine Funktion namens "sayHelloTo", die als Parameter eine Zeichenkette (String) verwendet, die bei Aufruf "Hello" und den Parameter ausgibt.
- Erstellen Sie eine Funktion namens "doubleThisNumber", die als Parameter einen int verwendet, der bei Aufruf den Wert des Parameters mit zwei multipliziert, woraufhin der neue Wert zurückgespielt wird. Der zurückgespielte Wert ist in einer Variablen zu speichern und auszugeben.

Führen Sie in das Thema while-Schleifen ein und erläutern Sie, wie diese eine Codeabschnitt ständig und immer weiter wiederholen, solange die daran geknüpfte Bedingung wahr ist.

# Aufgaben zu while-Schleifen

- Lassen Sie die Schüler eine Funktion erstellen, die beim Drücken der Taste aufgerufen wird und eine while-Schleife enthält, die zehnmal ausgeführt wird, wenn nichts weiter unternommen wird, die jedoch vorzeitig unterbrochen wird, wenn die Taste zuvor wieder losgelassen wird. In der while-Schleife kann sinnvollerweise "while" ausgegeben werden, damit deutlich ist, dass sie jetzt aktiv ist.
- Erstellen Sie eine Funktion mit einer while-Schleife, die in 1-Sek.-Abständen "Happy Birthday to You" ausgibt; bei Drücken eines Schalters soll die Schleife unterbrochen werden.

# 8. Fase – Introduktion til yderligere programmeringsprincipper (Funktioner og While-Løkker) (45 min):

I forbindelse anvendelsen af afstandsasensoren og farvesensoren, blev der til begge anvendt en funktion. Introducer derfor eleverne for funktioner og forklar hvordan de fungerer ved at indkapsle et stykke kode, der herefter kan kaldes igen og igen. Dette gør det både lettere at anvende (ændringer i koden skal kun foretages ét sted, fremfor mange steder), samt lettere at læse (funktionen kan indkapsle mange linjers kode, under et enkelt og meningsgivende navn).

# Opgaver til eleverne:

- Lav en funktion kaldet "printHello", der når den kaldes printer "Hello" til skærmen.
- Lav en funktion kaldet "sayHelloTo" hvilken tager en String som parameter, der når den kaldes printer "Hello", samt parameteret.
- Lav en funktion kaldet doubleThisNumber, der tager en int som parameter, der når den kaldes gange værdien fra parameteret med to, hvorefter den nye værdi returneres. Den returnerede værdi skal gemmes i en variabel og printes.

Introducer while-løkker og forklar hvordan de gentager et stykke kode igen og igen, så længe den tilknyttede betingelse er sand.

## Opgaver til eleverne:

- Lav en funktion indeholdende en while-løkke der kører 10 gange, hvori counteren printes.
- Lav en funktion med en while-løkke der printer "Happy Birthday to You" med intervaller af 1 sekund, trykkes en kontakt ned skal løkken afbrydes.

```
Funktion (ohne Wertrückgabe):
                                     Funktion (mit Wertrückgabe):
void loop() {
                                     void loop() {
  //runs the code inside-
                                       //stores the returned value-
  //myFunction
                                       //from myFunction
  myFunction();
                                       int returnedValue;
                                       returnedValue = myFunction();
//datatype name() {}
void myFunction() {
                                     //datatype name() {}
  //some code to run
                                     int myFunction() {
                                       //some code to run
                                       //return a value
                                       int value = 100;
                                       return value;
```

Abb. 60: Übersicht über die Anwendung von Funktionen

```
While-Schleife:
                                      Unterbrechung:
void loop() {
                                      void loop() {
  int x = 0;
                                        //while (argument - is
                                      correct)
  //while (argument is
                                        while (true) {
correct)
  while (x < 10) {
                                          if(something) {
                                             //break out of the loop
    x = x + 1;
                                            break;
}
                                        }
                                      }
```

Abb. 61: Übersicht über die Anwendungsmöglichkeiten von while-Schleifen

```
Function (return no value):
                                     Function (return a value):
void loop() {
                                     void loop() {
                                       //stores the returned value-
 //runs the code inside-
 //myFunction
                                       //from myFunction
 myFunction();
                                      int returnedValue;
                                       returnedValue = myFunction();
//datatype name() {}
void myFunction() {
                                     //datatype name() {}
                                     int myFunction() {
 //some code to run
}
                                       //some code to run
                                       //return a value
                                      int value = 100;
                                       return value;
```

Fig 60: Oversigt hvordan funktioner kan anvendes

```
While loop:
                                     Break:
void loop() {
                                     void loop() {
 int x = 0;
                                       //while (argument - is
                                     correct)
  //while (argument is
                                       while(true) {
correct)
 while (x < 10) {
                                         if(something) {
   x = x + 1;
                                           //break out of the loop
                                           break;
  }
}
                                         }
                                       }
```

Fig. 61: Oversigt over hvordan while-loops kan anvendes

# THEMENPROJEKT - AUTOMATISIERUNG (12 × 45 MIN.)

**Kurz zum Inhalt:** Das Projekt befasst sich mit der Automatisierung einer Kaffeebohnenzentrale, für die zwei Robotertypen entwickelt werden. Ein Roboter, der die Bohnen nach Qualität sortiert (Farbsortierung), und ein Roboter, der sie durch das Lager transportiert (Linienverfolgung). Die Schüler wählen aus, mit bzw. an welchem der beiden Roboter sie arbeiten möchten. Danach sollen sie sich in kleineren Gruppen relativ eigenständig mit dem Projekt beschäftigen (empfohlen werden hier Zweiergruppen).

**Lernergebnisse:** Die Schüler lernen die Anwendung von Inhalten aus früheren Lektionen in einem praktischen Kontext, um konkrete Probleme zu lösen – mit Bezug zu industriellen Arbeitsplätzen im 'wirklichen Leben'. Gleichzeitig erhalten sie einen tieferen Einblick in und Verständnis für die Entwicklung komplexer automatisierter Systeme sowie deren Grenzen.

Material: Im erarbeiteten Material wurde LEGO als Plattform für die zu entwickelnden Roboter verwendet. Um die Integration der Elektronik in die LEGO-Plattform zu erleichtern, wurden gleichzeitig mehrere hiermit kompatible 3D-Modelle entwickelt. Anleitungen hierfür und für 3D-Modelle werden – wie auch schon zuvor beschriebenes Material – auf der Webseite zur Verfügung gestellt. Gleiches gilt für einen vollständigen Überblick über die im Projekt verwendeten Komponenten, Lösungsvorschläge und Vorschläge zur Weiterentwicklung der Roboter usw. Es ist wichtig zu beachten, dass die Roboter auf viele verschiedene Arten und aus vielen verschiedenen Materialien konstruiert werden könn(t)en, weshalb die entwickelten LEGO-Handbücher nur eine von mehreren Lösungen sind.

# Projektkonzept:

Das folgende Projekt basiert auf einer Kaffeebohnenzentrale, in der die Bohnen zunächst nach Qualität sortiert und dann im zugehörigen Lager verteilt werden. Die Bohnen wurden zuvor in der Zentrale von Hand sortiert und anschließend von Hand im Lager verteilt. Dies erfordert anstrengende körperliche und zudem zeitaufwändige Arbeit. Die Zentrale möchte daher diese beiden Prozesse automatisieren, weshalb zwei Roboterarten gebaut und programmiert werden sollen, die jeweils die entsprechenden Funktionen ausführen können.

## Sortieren von Kaffeebohnen (Farbsortierung):

Die Bohnen werden nach Qualität sortiert, die an ihrer Farbe erkennbar ist. Für dieses Projekt bestehen die Bohnen aus 3D-gedruckten Bauklötzen (rot, grün, blau und gelb). Daher muss der Roboter eine Bohne entgegennehmen, sie einer Farbkategorie zuordnen, vier zugehörige Kartons so drehen, dass der entsprechende Karton dem Förderband zugewandt ist, und schlussendlich die Bohne in diesen ablegen und die Kartons in ihre Ausgansposition zurückdrehen können.

# TEMAPROJEKT - AUTOMATISERING (12 × 45 MIN.)

**Kort om indhold:** Projektet omhvandler automatiseringen af en kaffebønnecentral, hvortil der skal udvikles to typer af robotter. En robot der sorterer bønnerne efter kvalitet (farvesortering), samt en robot der transporter dem rundt på lageret (linjefølgning). Eleverne vælger hvilken en af de to robotter de vil arbejde med, hvorefter det er tiltænkt at de arbejder relativt selvstændigt med projektet i mindre grupper (det anbefales at eleverne er to elever pr. gruppe).

**Læringsudbytte:** Eleverne lærer at anvende det lærte fra tidl. lektioner i en praktisk kontekst, til at løse konkrete problemstillinger inspireret fra virkelighedens industrielle arbejdspladser. Samtidigt giver dette dem en dybere indsigt i og forståelse for udviklingen af komplekse automatiserede systemer, samt disses begrænsninger.

Materialer: I det udviklede materiale er LEGO blevet anvendt som platform for de udviklede robotter. For at lette integreringen af elektronik i LEGO-platformen, er der derfor samtidigt blevet udviklet en række 3D modeller der er kompatible hermed. Manualer hertil, samt 3D modeller er ligesom tidl. beskrevet materiale, gjort tilgængelig på hjemmesiden. Det samme gør sig gældende for en komplet oversigt over de i projektet anvendte komponenter, løsningsforslag, samt forslag til videre udvikling af robotterne mm. Det er vigtigt at notere sig, at robotterne kan konstrueres på mange forskellige måder og af mange forskellige materialer, de udviklede LEGO-manualer er derfor kun et bud på en løsning hertil.

# **Projektconcept:**

Introducer eleverne til følgende projekt, hvilket tager sit udgangspunkt i en kaffebønne central, hvor bønnerne først sorteres efter kvalitet, hvorefter de fordeles rundt på det tilknyttede lager. På centralen har man tidligere håndsorteret bønnerne, hvorefter de ved håndkraft er blevet fordelt på lageret. Dette indebærer hårdt fysisk arbejde, samtidigt med at det tager langtid. Centralens ønske er derfor, at disse to processer skal automatiseres, hvorfor der skal bygges og programmeres to typer af robotter, der hver kan varetage en af disse funktioner.

Kaffebønnesortering (farvesortering): Bønnerne sorteres efter kvalitet, hvilken afgøres af deres farve, i forbindelse med projektet udgøres bønnerne af 3D printede klodser (rød, grøn, blå og gul). Robotten skal derfor kunne modtage en bønne, registrere hvilken farvekategori denne tilhører, rotere fire tilknyttede kasser rundt, således at den korresponderende kasse står ud for transportbåndet, for derefter at aflevere bønnen heri og rotere kasserne tilbage til udgangspunktet igen.

**Transportering (linjefølgning):** De sorterede bønner pakkes i kasser, der i forbindelse med projektet udgøres af 3D printede kasser (rød, grøn, blå og gul), farven indikerer hvilket af fire korresponderende områder på lageret, de skal afleveres på. Lageret er blevet udstyret med

## Transportieren (Folgen einer Linie):

Die sortierten Bohnen werden in Kartons verpackt, die im Rahmen des Projekts aus 3D-gedruckten Kartons (rot, grün, blau und gelb) bestehen. Die Farbe gibt an, an welchen der vier entsprechenden Bereiche im Lager sie geliefert werden sollen. Das Lager wurde mit schwarzen Linien auf dem Boden versehen, nach denen die Roboter navigieren sollen. Zur Vermeidung von Unfällen sollen die Roboter außerdem anhalten, wenn sich vor ihnen ein Gegenstand befindet, der im Weg steht. Dies können Mitarbeiter aus Fleisch und Blut oder andere Roboter sein.

Die Zentrale ist bereits mit einem Bereich ausgestattet, in dem die Sortierung stattfindet. Außerdem haben die Mitarbeiter kürzlich die Arbeit mit dem Aufbringen der Linien abgeschlossen, auf denen die Roboter navigieren sollen, siehe Abb. 62.

Wenn die Schülerinnen und Schüler gewählt haben, mit welchem der beiden Roboter sie arbeiten möchten, können sie das in den nächsten beiden Abschnitten beschriebene Verfahren weiterverfolgen.

Die folgenden Videos zeigen Beispiele aus der Praxis – von Robotern, die für das automatische Sortieren und Transportieren von Kaffeebohnen entwickelt wurden.

**Sortieren von Kaffeebohnen:** https://www.youtube.com/watch?

reload=9&v=ionmIl-bNLI

**Transportieren:** https://www.youtube.com/watch?v=WIIS3vNSuQ4

# Sortieren von Kaffeebohnen:

Der Roboter besteht aus zwei Hauptteilen, die zu einem zusammengefasst sind. Der erste Teil ist das Förderband, auf das die Kaffeebohnen zuerst gelegt werden. Dann werden sie unter einen Farbsensor verbracht, der ihre Farbe erfasst (rot, grün, blau oder gelb). Anschließend soll der Förderer sie das letzte Stück hin zu einem Karton und in diesen hinein verfrachten. Der zweite Teil ist der Mechanismus zum Drehen der entsprechenden Kartons, sodass die Bohnen, wenn sie das Ende des Förderbandes erreichen, in den richtigen Karton hineinfallen. Siehe nachstehende Abb. 63.

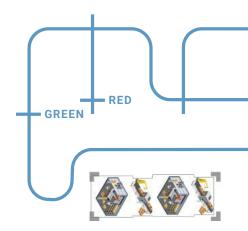


Abb. 62: Übersicht über die Ladezone und Beförderungsstrecken des Lagers

Fig. 62: Oversigt over lagerets lastezone og transporteringsruter

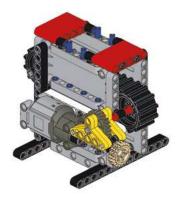
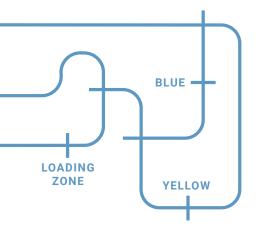


Abb. 63: LEGO-Modelle des Förderbandes und der Kartons

Fig. 63: LEGO-modellerne for transportbåndet og kasserne



sorte streger i gulvet, hvilke robotterne skal navigere efter. Derudover skal robotterne for at undgå ulykker, stoppe op hvis der er et objekt foran dem der spærrer vejen, dette kan være menneskelige medarbejdere eller andre robotter.

Centralen er allerede indrettet med et område hvorpå sorteringen foregår og derudover har medarbejder lige færdiggjort arbejdet, med at optegne de streger robotterne skal navigere efter, se Figur 62.

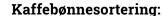
Når eleverne har valgt hvilken af de to robotter de vil arbejde med, kan de følge den tilhørende fremgangsmåde fra de efterfølgende to afsnit.

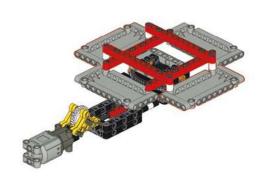
I videoerne nedenfor, kan der ses eksempler fra virkelighedens verden, på robotter der er designet til automatisk sortering af kaffebønner, samt transportering.

Kaffebønnesortering: https://www.youtube.com/watch?v=

ionmII-bNLI

Transportering: https://www.youtube.com/watch?v=WIIS3vNSuQ4





Robotten består af to hoveddele, hvilke kombineres til én. Den første del er transportbåndet, hvorpå kaffebønnerne først placeres, herefter fragtes de hen under en farvesensor hvorefter deres farve registreres (rød, grøn, blå eller gul), dernæst er det transportbåndets opgave at fragte dem det sidste stykke hen til og ned i en korresponderende kasse. Den anden del er den mekanisme der roterer de korresponderende kasser, således at når bønnerne når enden af transportbåndet, så falder de ned i den korrekte kasse. Se Figur 63 nedenfor.

## **Erste Schritte:**

#### 1. Entwickeln des Roboters:

Um eine Arbeitsgrundlage zu haben, ist die erste Aufgabe die Konstruktion des Roboters selbst.

# Aufgabe:

- Gehen Sie nach der LEGO-Anleitung "Bean Sorter Belt" vor und bauen Sie das Förderband.
- Gehen Sie nach der LEGO-Anleitung "Bean Sorter Rotator" vor und bauen Sie den Rotationsmechanismus für die Kartons.
- Gehen Sie nach der LEGO-Anleitung "Bean Sorter Combine" vor und bauen Sie die beiden Teile zusammen.
- Weisen Sie nun den Kartons ihre jeweile Farbe zu, ggf. mit einem Marker, und legen Sie fest, welche standardmäßig dem Förderband zugewandt sein soll.

#### Material:

2 x LEGO Power Functions Motor (Large).

## 2. Integrieren der Arduino-Plattform:

Jetzt soll der Arduino ins Spiel gebracht werden, also der Mikrocontroller zur Steuerung unseres Roboters, sowie die zugehörige Motorabschirmung, eine Erweiterung des Arduino, mit dem wir die Motoren steuern, und zusätzlich ein Steckbrett.

## Aufgabe:

• Gehen Sie nach der Anleitung "Bean Sorter – Integrate the Arduino platform" vor und fügen Sie Arduino, Motorabschirmung, Steckbrett und Akkusatz hinzu.

## Material:

- Arduino Uno
- Motorabschirmung R3
- Steckbrett (halbe Größe)
- Akkusatz + Kabel
- Arduinomodul (3D-gedruckt)
- Steckbrettmodul (3D-gedruckt)
- Akkusatzmodul (3D-gedruckt)
- · 2 x Stecker/Stecker-Kabel

## De Indledende Skridt:

#### 1. Konstruer robotten:

Det første der skal have gøres, er at konstruere selve robotten, så der er noget at arbejde ud fra.

# Opgave:

- Følg LEGO manualen "Bean Sorter Belt" og byg transportbåndet.
- Følg LEGO manualen "Bean Sorter Rotator" og byg rotationsmekanismen til kasserne.
- Følg LEGO manualen "Bean Sorter Combine" og byg de to dele sammen.
- Tildel nu kasserne hver deres individuelle farve, evt. med en tusch og vælg hvilken der som standard skal vende ind mod transportbåndet.

## Materialer:

· 2stk. LEGO Power Functions Motor (Large).

## 2. Integrer Arduino platformen:

Nu skal Arduinoen tilføjes, hvilken er den mikrocontroller der skal stå for at kontrollere vores robot, samt det dertilhørende motorshield, hvilket er en udvidelse til Arduinoen som man anvender til at styre motorerne med, samt et breadboard.

## Opgave:

 Følg manualen "Bean Sorter – Integrate the Arduino platform" og tilføj Arduinoen, motor shieldet, breadboardet, samt batteripakken.

## Materialer:

- Arduino Uno.
- Motor shield. R3
- Breadboard (halv størrelse).
- Batteripakke + Ledning.
- Arduino modul (3D printet).
- Breadboard modul (3D printet).
- Batteripakke modul (3D printet).
- 2stk. han/han ledninger.

# 3. Test der Motoren:

Im folgenden Schritt sollen die Motoren angeschlossen und gleichzeitig einige kleinere Tests durchgeführt werden, um sicherzustellen, dass sie wie vorgesehen funktionieren.

# Aufgabe:

- · Motoren mit der Motorabschirmung verbinden.
- Förderband 2 Sek. lang vorwärts laufen lassen mit Höchstgeschwindigkeit.
- · Förderband 2 Sek. lang rückwärts laufen lassen mit halber Geschwindigkeit.
- · Kartons 2 Sek. lang im Uhrzeigersinn drehen lassen mit Höchstgeschwindigkeit.
- Kartons 2 Sek. lang gegen den Uhrzeigersinn drehen lassen mit halber Geschwindigkeit.

## **Cheat sheets:**

- · Aktoren DC-Motor
- Code pinMode
- · Code digitalWrite
- Code analogWrite
- Code delay

# 3. Test motorerne:

Det er nu tid til at få tilsluttet motorerne og samtidigt gennemføre en række mindre test, for at sikre os at de virker efter hensigten.

# Opgave:

- · Forbind motorerne til motor shieldet.
- Få transportbåndet til at køre fremad i 2 sekunder, med fuld hastighed.
- Få transportbåndet til at køre baglæns i 2 sekunder, med halv hastighed.
- Få kasserne til at rotere med uret i 2 sekunder, med fuld hastighed.
- Få kasserne til at rotere mod uret i 2 sekunder, med halv hastighed.

# **Cheat Sheets:**

- Effectors DC Motor.
- Code pinMode.
- Code digitalWrite.
- Code analogWrite.
- · Code delay.

## Förderband:

Für diesen Teil empfiehlt sich das Erstellen einer neuen Programmdatei – aber speichern Sie unbedingt die vorherigen aus den *Ersten Schritten*.

## 1. Einsatz von Funktionen zur Steuerung des Förderbandes:

Funktionen sind eine gute Methode, um eine Codeabschnitt abzugrenzen, damit sie ihn immer wieder verwenden können, ohne dass wir den Code jedes Mal neu schreiben müssen. Gleichzeitig macht es den Code für den/die Anwender\*in besser lesbar.

## Aufgabe:

- Erstellen Sie eine Funktion "startBelt", die das Förderband startet.
- Erstellen Sie eine Funktion "stopBelt", die das F\u00f6rderband anh\u00e4lt.
- Erstellen Sie eine Funktion "deliverBean", die die Bohne "anliefert" (das Förderband startet und essoll 5 Sek. lang laufen bis lässt, woraufhin es wieder angehalten wirdstoppt).
- Überprüfen Sie, dass alle Funktionen einwandfrei arbeiten.

## **Cheat sheets:**

Code – Functions

#### 2. Integrieren des Farbsensors:

Vor dem Einsatz des Förderbandes müssen wir den Farbsensor integrieren und prüfen, ob er bestimmungsgemäß funktioniert.

## Aufgabe:

- Gehen Sie nach der Anleitung "Bean Sorter Integrate Color-Sensor" vor und integrieren Sie den Farbsensor.
- Legen Sie eine "Bohne" auf das Förderband unter den Farbsensor und geben Sie am Bildschirm ein, um welche Farbe es sich handelt. Führen Sie dies für alle Farben aus (Rot, Grün, Blau und Gelb).

## Material:

- Farbsensor (TCS3200)
- Farbsensormodul (3D-gedruckt)
- 7 x Stecker/Buchse-Kabel

## **Cheat sheets:**

- Sensors Color-Sensor (TCS3200)
- Code Variables
- · Code Serial Print

# Transportbåndet:

Det anbefales at oprette en ny programfil til denne del, men gem gerne den tidligere fra De Indledende Skridt.

# 1. Anvend funktioner til at styre transportbåndet:

Funktioner er en god måde at indkapsle et stykke kode på, så det kan bruges mange gange, uden at man skal skrive koden hver eneste gang. Samtidigt gør det koden lettere at læse for mennesker.

# Opgave:

- Lav en funktion "startBelt", der starter transportbåndet.
- Lav en funktion "stopBelt", der stopper transportbåndet.
- Lav en funktion "deliverBean", der afleverer bønnen (starter transportbåndet og lader det køre i 5 sekunder, hvorefter det stoppes igen).
- · Test at alle tre funktioner virker.

## **Cheat Sheets:**

· Code - Functions.

# 2. Integrer farvesensoren:

Førend transportbåndet kan anvendes, skal farvesensoren først integreres heri, samt at det skal testes om den virker efter hensigten.

## Opgave:

- Følg manualen "Bean Sorter Integrate Color-Sensor" og integrer farvesensoren.
- Placer en "bønne" på transport båndet under farvesensoren og print til skærmen, hvilken farve det er. Gør dette for alle farverne (rød, grøn, blå og gul).

## Materialer:

- Farvesensor (TCS3200).
- Farvesensor modul (3D printet).
- 7stk. han/hun ledninger.

## **Cheat Sheets:**

- Sensors Color-Sensor (TCS3200).
- · Code Variables.
- · Code Serial Print.

# 3. Auf erkannte Farben reagieren:

Wir können jetzt unser Förderband steuern und den Farbsensor einsetzen, um Farben zu erkennen. Bis dahin macht das Programm aber immer nur das Gleiche. Daher möchten wir nun eine Struktur für das Programm erstellen, das es ermöglicht verschiedene Aktionen auszuführen, je nachdem, welche Farbe der Farbsensor erkennt (Rot, Grün, Blau oder Gelb).

# Aufgabe:

- · Starten Sie das Förderband.
- Wenn der Farbsensor "red" liest:
  - "red" an den Bildschirm auseingeben.
  - Förderband anhalten und die "Bohne" abliefern.
- Überprüfen, dass es funktioniert wenn ja, einen "if else" für jede der anderen Farben herstellen (Grün, Blau und Gelb), der die gleiche Funktionalität wie die der Farbe Rot enthält.

# **Cheat sheets:**

· Code - Conditionals.

# 3. Reager når farver læses:

Transportbåndet kan nu kontrolleres, samt anvende farvesensoren til at registrere farver, men indtil videre gør den det samme hele tiden. Det er derfor på tide at oprette en struktur for programmet, der anvender betingelser til at lade programmet foretage forskellige handlinger, alt efter hvilken farve farvesensoren registrerer (rød, grøn, blå eller gul).

# Opgave:

- · Start transportbåndet.
- · Hvis farvesensoren læser "red":
  - Print "red" til skærmen.
  - Stop transportbåndet og aflever "bønnen".
- Test at dette virker, når det gør, lav da en "if else" for hver af de andre farver (grøn, blå og gul), der indeholder den samme funktionalitet som den for den røde farve

## **Cheat Sheets:**

• Code – Conditionals.

## Die Kartons:

Für diesen Teil empfiehlt sich das Erstellen einer neuen Programmdatei – aber speichern Sie die vorherige zum Förderband.

#### 1. Funktionen zum Rotieren der Kartons einsetzen:

Wie beim Förderband möchten wir auch hier Funktionen zum Abgrenzen des Codes für die Motorsteuerung einsetzen, damit er wiederverwendet werden kann.

# Aufgabe:

- Erstellen Sie eine Funktion "startRotation", die die Kartons in eine Drehung gegen den Uhrzeigersinn versetzt.
- Erstellen Sie eine Funktion "stopRotation", die die Drehung der Kartons anhält.
- Erstellen Sie eine Funktion "rotatePosition", die einen int als Parameter nimmt (nennen Sie den Parameter "position") abgesehen vom Ausgeben des Parameters an den Bildschirdas Programmm bleibt diese Funktion ansonsten leer.
- · Überprüfen Sie, dass die Funktionen einwandfrei arbeiten.

## **Cheat Sheet:**

· Code - Functions (mit Parametern).

## 2. Schalter integrieren:

Bisher konnten wir die Kartons nur zeitabhängig hin und her rotieren, aber da die Zeit, die zum Rotieren einer Runde benötigt wird, davon abhängt, wie viel Energie noch in den Batterien vorhanden ist oder wie viel Schmutz ggf. in das Getriebe gelangt ist usw., ist dies keine optimale Lösung. Daher wollen wir einen Schalter in unseren Aufbau integrieren, mit dem wir jedes Mal die Rotation der Kartons um eine Position erkennen.

# Aufgabe:

- Gehen Sie nach der Anleitung "Bean Sorter Integrate Switch" vor und integrieren Sie den Schalter.
- Überprüfen Sie durch Anzeigen seines Zustandes an das Programm, dass er funktioniert.

# Material:

- Schalter
- Schaltermodul (3D-gedruckt)
- Widerstand 1 KΩ
- 3 x Stecker/Stecker-Kabel.

#### Kasserne:

Det anbefales at oprette en ny programfil til denne del, men gem den tidl. fra Transportbåndet.

## 1. Anvend funktioner til at rotere kasserne:

Anvend som ved transportbåndet, endnu engang funktioner til at indkapsle koden for motorstyring, sådan at den kan genbruges igen og igen.

## Opgave:

- · Lav en funktion "startRotation", der sætter kasserne i gang med at rotere mod uret.
- Lav en funktion "stopRotation", der stopper rotationen af kasserne.
- Lav en funktion "rotatePosition" der tager en int som parameter (kald parameteret "position"), på nær at printe parameteret til skærmen, efterlades denne funktion tom.
- · Test at funktionerne virker.

#### **Cheat Sheets:**

Code – Functions (with parameters).

## 2. Integrer switchen:

Indtil nu kasserne kun kunnet rotere frem og tilbage, baseret på tid, men fordi tiden det tager at rotere en omgang varierer alt efter hvor meget strøm der er tilbage på batterierne, hvor meget snavs der er i kommet ind i gearingen mm. er dette ikke nogen god løsning. Integrere derfor switch i konstruktionen, sådan at denne kan bruges til at registrere hver gang kasserne har roteret en plads.

# Opgave:

- Følg manualen "Bean Sorter Integrate Switch" og integrer switchen.
- · Test at switchen virker, ved at printe dens tilstand til skærmen.

#### Materialer:

- · Switch.
- Switch modul (3D printet).
- 1KΩ modstand.
- 3stk. han/han ledninger.

# **Cheat Sheets:**

Sensors – Switch.

## **Cheat Sheet:**

Sensors – Switch.

## 3. Anzahl der Kartonrotationen überprüfen:

Da der Schalter bei jedem Passieren eines Kartons automatisch gedrückt wird, können wir damit prüfen, um wie viele Positionen sich die Kartons bewegt haben, indem wir die Anzahl des Drückens registrieren. Auf diese Weise können wir immer sicherstellen, dass die Kartons sich um genau die Anzahl der gewünschten Positionen verschieben. Hierzu müssen wir die Funktion "DeliverBean" abgeschlossen haben, damit sie die gewünschte Anzahl an Rotationen ausführt.

# Aufgabe:

- In der Funktion "deliverBean":
  - Rotation der Kartons starten.
  - Neue Zählervariable namens "i" erstellen und ihr den Wert o zuordnen.
  - Eine while-Schleife mit der Bedungung (i < position) erstellen.
    - Wenn Schalter gedrückt ist:
      - "i" um 1 erhöhen.
      - Ca. 200 ms warten (je nach Rotationsgeschwindigkeit).
  - Kartonrotation anhalten.
- Funktion durch Aufrufen mit verschiedenen Parametern (1, 2, 3 usw.) überprüfen.
- Zählen Sie nun, wie oft die Kartons um eine Position rotieren müssen, bevor sie sich vor dem Förderband befinden, und wie oft sie rotieren müssen, bevor sie wieder an ihrem Ausgangspunkt sind. Schreiben Sie diese Zahlen auf ein Blatt Papier.

# 3. Kontroller hvor mange gange kasserne roterer:

Fordi at switchen automatisk trykkes ned hvergang en kasse kommer forbi den, kan denne bruge dette til at kontrollere hvor mange positionerne kasserne har flyttet sig, ved at tælle antallet af gange switchen trykkes ned. På den måde kan man altid sikre sig, at kasserne flytter sig præcist det antal positionerne der ønskes. Til det formål skal funktionen "rotatePosition" færdiggøres, således at den kan kaldes med det antal af rotationer der ønskes.

# Opgave:

- Inde i funktionen "rotatePosition":
  - Start rotationen af kasserne.
  - Lav en ny tæller variabel kaldet "i" og giv den værdien o.
  - Lav et while loop med betingelsen (i < position).
    - Hvis knappen er trykket ned:
      - Inkrementer "i" med 1.
      - Vent ca. 200ms (afhængig af rotations hastigheden).
  - Stop rotationen af kasserne.
- Test at funktionen virker, ved at kalde den med forskellige parametre (1, 2, 3 osv).
- Tæl nu hvor mange gange hver af kasserne skal rotere en position, førend de står foran transportbåndet, samt hvor mange gange de skal rotere førend de er tilbage ved deres udgangspunkt igen. Skriv disse tal ned på et stykke papir.

## Förderband und Kartons kombinieren:

Für diesen Teil empfiehlt sich das Erstellen einer neuen Programmdatei – aber speichern Sie wieder unbedingt die vorherigen Funktionen.

# 1. Kombinieren des gesamten Aufbaus:

Jetzt ist es Zeit, das Förderband mit den Kartons zu kombinieren, damit der Roboter vollautomatisch funktioniert. Wir haben bereits fast alle dafür erforderlichen Arbeiten durchgeführt, insbesondere durch den Einsatz von Funktionen.

# Aufgabe:

- · Förderband starten.
- · Wenn der Farbsensor "red" liest:
  - "red" an den Bildschirm ausgeben.
  - Förderband anhalten.
  - Kartons so rotieren lassen, dass der rote vor dem Förderband steht.
  - "Bohne" abliefern.
  - Kartons so rotieren lassen, dass der rote erneut an seinem Ausgangspunkt steht
- Überprüfen Sie, dass dies alles funktioniert und erweitern Sie dann das Programm um die übrigen Farben (Grün, Blau und Gelb).

## Varianten der Aufgabe:

Sie können die Aufgabe weiter variieren bzw. weiterentwickeln und so die Funktionen des Aufbaus erweitern, z. B. verschiedene LEDs zum Leuchten bringen, um anzuzeigen, welche Aktionen der Roboter gerade ausführt. Eine weitere Möglichkeit wäre, einen Halter zu bauen, der mit Hilfe eines Servomotors nacheinander neue Bohnen auf das Förderband legt.

# Kombiner Transportbåndet og Kasserne:

Det anbefales at oprette en ny programfil til denne del, men gem gerne den tidl. fra De Indledende Skridt.

## 1. Kombiner det hele:

Det er nu på tide at kombinere transportbåndet med kasserne, så robotten er fuldt automatiseret. Heldigvis er næsten alt arbejdet der kræves hertil allerede lavet, gennem vores anvendelse af funktioner.

# Opgave:

- · Start transportbåndet.
- Hvis farvesensoren læser "red":
  - Print "red" til skærmen.
  - Stop transportbåndet.
  - Roter kasserne sådan at den røde kasse står ud for transportbåndet.
  - Aflever "bønnen".
  - Roter kasserne sådan at den røde kasse igen står på dens udgangspunkt.
- Test at dette virker og udvid derefter programmet til også at omfatte de øvrige farver (grøn, blå og gul).

## Ideer til Videreudvikling:

Kom gerne selv på nogle ideer til potentielle udvidelser, det kan eks. være at få forskellige LED'er til at lyse for at indikere hvilke handlingerne robotten pt. er ved at foretage sig. Det kan også være at bygge en holder der ved hjælp af en servo-motor, selv kan lægge nye bønner på transportbåndet, én ad gangen.

# **Transportieren**

Der Roboter ruht auf zwei Vorderrädern (die ihn antreiben) und einem Kugelrad. Siehe nachstehende Abb. 64.

## **Erste Schritte:**

#### 1. Entwickeln des Roboters:

Die erste Aufgabe ist die Konstruktion des Roboters selbst, um eine Arbeitsgrundlage zu haben.

# Aufgabe:

Gehen Sie nach der LEGO-Anleitung "Transporter – Transporter" vor und bauen Sie das Fördererfahrzeug (Transporter).

#### Material:

LEG® + 2 x LEGO Power Functions Motor (L).

## 2. Integrieren der Arduino-Plattform:

Jetzt soll der Arduino angewendet werden, also der Mikrocontroller zur Steuerung unseres Roboters, sowie die zugehörige Motorabschirmung, eine Erweiterung des Arduino, mit dem wir die Motoren steuern, sowie ein Steckbrett.

# Aufgabe:

 Gehen Sie nach der Anleitung "Transporter – Integrate the Arduino platform" vor und fügen Sie Arduino, Motorabschirmung, Steckbrett und Akkusatz hinzu.

## Material:

- Arduino Uno
- Motorabschirmung R3
- Steckbrett (halbe Größe)
- Akkusatz + Kabel
- Arduinomodul (3D-gedruckt)
- Steckbrettmodul (3D-gedruckt)
- Akkusatzmodul (3D-gedruckt)
- · 2 x Stecker/Stecker-Kabel.

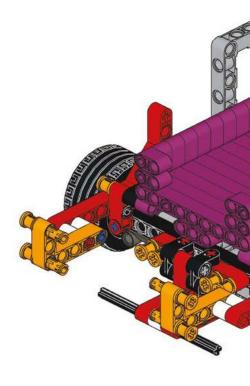
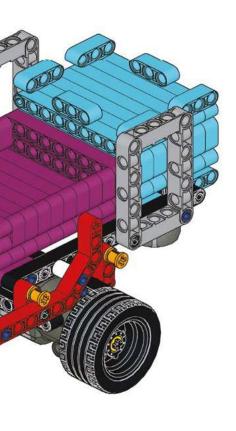


Fig. 64: LEGO-modellen til transportrobotten



# Transportering:

Robotten hviler på to forhjul, hvilke fungerer som dens drivkraft, samt et kuglehjul. Se Figur 64 nedenfor.

## De Indledende Skridt:

## 1. Konstruer robotten:

Det første der skal gøres, er at konstruere selve robotten, så der er noget at arbejde ud fra.

# Opgave:

 Følg LEGO manualen "Transporter – Transporter" og byg transporteren.

#### Materialer:

LEGO + 2stk. LEGO Power Functions Motor (L).

## 2. Integrer Arduino platformen:

Nu skal Arduinoen tilføjes, hvilket er den mikrocontroller der skal stå for at kontrollere vores robot, samt det dertilhørende motorshield, hvilket er en udvidelse til Arduinoen som man anvender til at styre motorerne med, samt et breadboard.

# Opgave:

 Følg manualen "Transporter – Integrate the Arduino platform" og tilføj Arduinoen, motor shieldet, breadboardet, samt batteripakken.

## Materialer:

- Arduino Uno.
- Motor shield. R3
- · Breadboard (halv størrelse).
- · Batteripakke + Ledning.
- Arduino modul (3D printet).
- Breadboard modul (3D printet).
- Batteripakke modul (3D printet).
- 2stk. han/han ledninger.

# 3. Test der Motoren:

Jetzt ist es Zeit, die Motoren anzuschließen und gleichzeitig einige kleinere Tests durchzuführen, um sicherzustellen, dass alles wie gedacht funktioniert.

# Aufgabe:

- · Motoren mit der Motorabschirmung verbinden.
- Den Transporter 2 Sek. lang vorwärts laufen lassen mit Höchstgeschwindigkeit.
- Den Transporter 2 Sek. lang rückwärts laufen lassen- mit halber Geschwindigkeit.
- Den Transporter 2 Sek. lang nach links drehen lassen mit Höchstgeschwindigkeit.
- Den Transporter 2 Sek. lang nach rechts drehen lassen mit halber Geschwindigkeit.

## **Cheat Sheet:**

- · Aktoren DC-Motor
- Code pinMode
- · Code digitalWrite
- · Code analogWrite
- Code delay

# 3. Test motorerne:

Det er nu tid til at få tilsluttet motorerne og samtidigt gennemføre en række mindre test, for at sikre os at de virker efter hensigten.

# Opgave:

- · Forbind motorerne til motor shieldet.
- Få transporteren til at køre fremad i 2 sekunder, med fuld hastighed.
- Få transporteren til at køre baglæns i 2 sekunder, med halv hastighed.
- Få transporteren til at dreje til venstre i 2 sekunder, med fuld hastighed.
- Få transporteren til at dreje til højre i 2 sekunder, med halv hastighed.

# **Cheat Sheets:**

- Effectors DC Motor.
- · Code pinMode.
- · Code digitalWrite.
- Code analogWrite.
- · Code delay.

# Nach Linien navigieren:

Für diesen Teil empfiehlt sich das Erstellen einer neuen Programmdatei nachdem die vorherigen Schritte gespeichert wurden.

## 1. Funktionen zum Steuern des Transporters einsetzen:

Funktionen sind eine gute Methode, um einen Codeabschnitt abzugrenzen, damit sie immer wieder verwendet werden können, ohne dass wir den Code jedes Mal neu schreiben müssen. Gleichzeitig macht es den Code für den/die Anwender\*in besser lesbar.

## Aufgabe:

- Erstellen Sie eine Funktion "driveForward", die den Roboter vorwärts fahren lässt.
- Erstellen Sie eine Funktion "crossLine", die den Roboter ca. 5 cm vorwärts fahren und ihn dann wieder anhalten lässt.
- Erstellen Sie eine Funktion "turnLeft", die den Roboter eine viertel Drehung nach links ausführen lässt.
- Erstellen Sie eine Funktion "turnRight", die den Roboter eine viertel Drehung nach rechts ausführen lässt.
- Erstellen Sie eine Funktion "uTurn", die den Roboter eine halbe Drehung ausführen lässt – Richtung egal.
- Erstellen Sie eine Funktion "stopRobot", die den Roboter für 1 Minute anhalten lässt.
- Überprüfen Sie, dass alle sechs Funktionen einwandfrei arbeiten.

#### **Cheat Sheet:**

Code – Functions.

# 2. Integrieren der IR-Sensoren:

Bevor wir den Roboter zum Folgen der Bodenlinien bringen können, müssen wir die beiden IR-Sensoren einbauen und überprüfen, ob sie richtig funktionieren.

## Aufgabe:

- Gehen Sie nach der Anleitung "Transporter Integrate IR-Sensors" vor und integrieren Sie die IR-Sensoren.
- Zunächst den Wert des linken IR-Sensors ablesen und ans Programm den Bildschirm ausweitergeben; danach die Werte für Weiß und Schwarz notieren. Den Vorgang für den rechten IR-Sensor wiederholen.

# Naviger efter Linjerne:

Det anbefales at oprette en ny programfil til denne del, men gem gerne den tidligere fra De Indledende Skridt.

# 1. Anvend funktioner til at styre transporteren:

Funktioner er en god måde at indkapsle et stykke kode på, så det kan bruges mange gange, uden at man skal skrive koden hver eneste gang. Samtidigt gør det koden lettere at læse for mennesker.

# Opgave:

- · Lav en funktion "driveForward", der kører robotten fremad.
- Lav en funktion "crossLine", der kører robotten ca. 5cm frem og stopper den igen.
- Lav en funktion "turnLeft", der drejer robotten en kvart omgang mod venstre.
- Lav en funktion "turnRight", der drejer robotten en kvart omgang mod højre.
- Lav en funktion "uTurn", der drejer robotten en halv omgang, retningen er ligegyldig.
- Lav en funktion "stopRobot", der stopper robotten i 1 minut.
- · Test at alle seks funktioner virker.

## **Cheat Sheets:**

· Code - Functions.

# 2. Integrer IR-sensorerne:

Førend transporteren kan følge linjerne i gulvet, skal dens to IR-sensorer først integreres, samt at det skal testes om de virker efter hensigten.

# Opgave:

- Følg manualen "Transporter Integrate IR-Sensors" og integrer IR-sensorerne.
- Læs først den venstre IR-sensors værdi og print denne til skærmen, noter dernæst værdierne for hvid og sort. Gentag herefter dette for den højre IR-sensor.

#### Materialer:

- IR-sensor (QRE1113, Analog).
- IR-sensor modul (3D printet).
- · 6stk. han/han ledninger.

#### Material:

- IR-Sensor (QRE1113, analog).
- IR-Sensormodul (3D-gedruckt)
- 6 x Stecker/Stecker-Kabel

# Cheat sheets (Spickzettel):

- Sensors IR-sensor (QRE1113, analog).
- Code Variables.
- Code analogRead.
- · Code Serial Print.

## 3. Folgen der Linie:

Jetzt können wir den Roboter mithilfe der IR-Sensoren so steuern, dass er einer Linie folgt. Dazu müssen wir für jeden IR-Sensor den Wert berechnen, der genau zwischen dem Wert für Weiß und dem Wert für Schwarz liegt. Wir nennen diesen Wert den Schwellenwert. Um einer Linie zu folgen, lassen wir die Motoren laufen, wenn ihr jeweiliger IR-Sensor auf Weiß sieht, andernfalls müssen sie anhalten. Auf diese Weise korrigiert der Roboter automatisch, wenn er droht vom Kurs abzukommen und dabei die Linie zu überfahren. Auf die gleiche Weise stoppt er auch automatisch, wenn er an eine Kreuzung kommt, weil beide Sensoren jetzt Schwarz registrieren. Beginnen Sie mit der Simulation, indem Sie den Roboter von Hand steuern.

#### Aufgabe:

- Den linken Motor zum Laufen veranlassen, wenn der linke IR-Sensor weniger als den Schwellenwert liest; sonst soll er anhalten.
- Den rechten Motor zum Laufen veranlassen, wenn der rechte IR-Sensor weniger als den Schwellenwert liest; sonst soll er anhalten.
- Überprüfen Sie, dass der Roboter jetzt der Linie zu folgen vermag und anhält, wenn er eine Kreuzung erreicht.

## **Cheat sheets**

Code – Conditionals.

# 4. Folgen der Linie – als Funktion:

Bisher stoppt unser Roboter automatisch, wenn er eine Kreuzung erreicht, aber er fährt auch von selbst wieder an, wenn man ihn manuell über die Kreuzung hinausschiebt oder auf eine neue Linie stellt. Wenn wir also unsere in den Funktionen "turnLeft" und "turnRight" gespeicherten Abbiegebefehle anwenden möchten, wenn wir eine Kreuzung erreichen, gibt es ein Problem. Versuchen Sie mal dies mit dem Befehl auszuführen – der Roboter wird sich jetzt einfach nur drehen und der Linie überhaupt nicht mehr folgen. Das liegt daran, dass die Liniencodes, nach denen er der Linie folgt, im Bruchteil einer Sekunde ausgeführt werden, woraufhin

#### **Cheat Sheets:**

- Sensors IR-sensor (QRE1113, Analog).
- Code Variables.
- Code analogRead.
- · Code Serial Print.

# 3. Følg linjen:

Det er nu på tide at anvende IR-sensorerne, til at kontrollere robotten så den kan følge en linje. Hertil skal der for hver IR-sensor, udregnes den værdi der ligger lige mellem værdien for hvid og værdien for sort. Denne værdi kaldes for tærsklen. For at følge en linje, skal motorerne køre hvis deres respektive IR-sensor ser hvidt, ellers skal de stoppe. På den måde vil robotten automatisk rette op, hvis den er ved at køre skævt og derved krydse ind over linjen, på samme måde vil den også automatisk stoppe når den kommer til et kryds, fordi begge sensorer nu ser sort. Begynd gerne med at simulere dette, ved at styre robotten med hånden.

### Opgave:

- Få venstre motor til at køre hvis venstre IR-sensor læser mindre end tærsklen, ellers skal den stoppe.
- Få højre motor til at køre hvis højre IR-sensor læser mindre end tærsklen, ellers skal den stoppe.
- Test at robotten nu kan f
  ølge linjen, samt at den stopper n
  år den n
  år til et kryds

# **Cheat Sheets:**

Code – Conditionals.

## 4. Følg linjen, som en funktion:

Indtil videre stopper robottenautomatisk når den når et kryds, men den starter ligeledes også af sig selv igen, hvis man manuelt skubber den forbi krydset, eller sætter den over på en ny linje. Hvis man derfor gerne vil bruge dreje kommandoerne gemt i funktionerne "turnLeft" og "turnRight", til at navigere i et kryds, skaber dette et problem. Prøv gerne at kalde disse fra koden – nu står robotten blot og drejer rundt og følger slet ikke linjen mere. Dette skyldes at de linjer kode der får den til at følge linjen, er udført på et splitsekund, hvorefter dreje kommandoen kaldes igen. Der skal derfor laves en selvstændig funktion der får robotten til at følge linjen, indtil den når et kryds, hvorefter funktionen afsluttes.

der Befehl zum Drehen erneut aufgerufen wird. Wir müssen daher eine eigenständige Funktion erstellen, die bewirkt, dass der Roboter der Linie folgt, bis er eine Kreuzung erreicht, wonach die Funktion endet.

# Aufgabe:

- · Erstellen Sie eine Funktion "followLine".
  - Verwenden Sie eine while-Schleife mit der Bedingung (true).
    - Den linken Motor zum Laufen veranlassen, wenn der linke IR-Sensor weniger als den Schwellenwert liest; sonst soll er anhalten.
    - Den rechten Motor zum Laufen veranlassen, wenn der rechte IR-Sensor weniger als den Schwellenwert liest; sonst soll er anhalten.
    - Wenn beide IR-Sensoren weniger als den Schwellenwert lesen, halten Sie beide Motoren an und unterbrechen Sie die while-Schleife.
- Überprüfen Sie, dass die Funktion einwandfrei arbeitet.
- Überprüfen Sie sodann, ob sie zusammen mit den Drehfunktionen funktioniert –
  und zudem die Funktion des Geradeausfahrens an einer Kreuzung –, indem Sie
  die untenstehenden Funktionen aufrufen und daraufhin beobachten, ob der Roboter
  die vorgesehene Route fährt. Denken Sie daran, dass nach Ablauf einer Minute die
  Funktion "stopRobot" zu Ende ist, und der Ablauf (die Sequenz) von vorn beginnt.
  - followLine
  - turnLeft
  - followLine
  - crossLine
  - followLine
  - turnRight
  - followLine
  - uTurn
  - stopRobot

\_

### **Cheat sheets:**

· Code - While loop (mit Unterbrechung).

# Opgave:

- · Lav en funktion "followLine".
  - Anvend et while loop med betingelsen (true).
    - Få venstre motor til at køre hvis venstre IR-sensor læser mindre end tærsklen, ellers skal den stoppe.
    - Få højre motor til at køre hvis højre IR-sensor læser mindre end tærsklen, ellers skal den stoppe.
    - Hvis begge IR-sensorer læser mindre end tærsklen, stop begge motorer og afbryd while loopet.
- · Test om funktionen virker.
- Test nu om den virker sammen med dreje funktionerne, samt funktionen for at køre lige ud i et kryds, ved eksempelvis at kalde funktionerne nedenfor og derefter observere om robotten kører den forventede rute. Husk at når der er gået et minut er funktionen "stopRobot" færdig, også starter sekvensen forfra.
  - followLine.
  - turnLeft.
  - followLine.
  - crossLine.
  - followLine.
  - turnRight.
  - followLine.
  - uTurn.
  - stopRobot.

# **Cheat Sheets:**

• Code – While loop (with break).

# Nach Umfeld navigieren:

Bevor Sie fortfahren, empfiehlt es sich das erstellte Programm zu speichern und eine Kopie zu erstellen, um anschließend in der Kopie weiterzuarbeiten.

## 1. Integrieren des Abstandssensors:

Der Roboter kann jetzt gemäß den Linien auf dem Boden des Lagers navigieren, aber was ist, wenn sich ein anderer Roboter vor ihm befindet oder ein Mitarbeiter übersehen hat, dass er direkt auf ihn zusteuert? Um mögliche Probleme und Verletzungen zu vermeiden, bringen wir jetzt einen Abstandssensor ins Spiel, mit dem der Roboter die Entfernung zum nächsten Objekt messen kann. Dadurch können wir im nächsten Schritt den Roboter so programmieren, dass er anhält, falls ein Objekt ihm (oder er dem Objekt) zu nahe gekommen ist.

# Aufgabe:

- Gehen Sie nach der Anleitung "Transporter Ultrasound-Sensor" vor und integrieren Sie den Abstandssensor.
- · Geben Sie die Entfernung zum nächsten Objekt an den Bildschirm aus.

## Material:

- Abstandssensor (HC-RSo<sub>4</sub>).
- Abstandssensormodul (3D-gedruckt)
- 4 x Stecker/Stecker-Kabel.

#### **Cheat sheets**

Sensors – Ultrasonic-Sensor (HC-RSo<sub>4</sub>).

## 2. Beim einem im Weg stehenden Objekt Roboter anhalten:

Unsere Funktion "followLine" wollen wir nun so aufwerten, das der Roboter stoppt, wenn der Abstand zu einem Objekt vor ihm 20 cm unterschreitet.

## Aufgabe:

- Funktion "followLine" wie folgt aufwerten:
  - Zuerst wird in der while-Schleife der Abstand zum nächsten Objekt gelesen.
  - Werten Sie die Bedingungen für erlaubtes Motorlaufen so auf, dass die Bedingung jetzt Folgendes vorsieht:
    - Wenn linker IR-Sensor unter Schwellenwert und Abstand über 20 cm, dann linken Motor laufen lassen, sonst anhalten.
    - Wenn rechter IR-Sensor unter Schwellenwert und Abstand über 20 cm, dann rechten Motor laufen lassen, sonst anhalten.

# Naviger efter Omgivelserne:

Det anbefales at oprette gemme en kopi af programmet, for derefter at fortsætte arbejdet i kopien.

## 1. Integrer afstandssensoren:

Robotten er nu i stand til at navigere efter linjerne på lagerets gulv, men hvad hvis der er en anden robot foran den, eller at en medarbejder har overset at den kommer lige imod dem. For at undgå de problemer og skader der kan opstå herved, kan der integreres en afstandssensor, der lader robotten måle afstanden hen til det nærmeste objekt. Derved kan den bagefter programmeres til at stoppe, skulle et objekt være for tæt på den.

# Opgave:

- Følg manualen "Transporter Ultrasound-Sensor" og integrer afstandssensoren.
- Print afstanden til det nærmeste objekt, på skærmen.

#### Materialer:

- · Afstandssensor (HC-RSo<sub>4</sub>).
- Afstandssensor modul (3D printet).
- · 4stk. han/han ledninger.

# **Cheat Sheets:**

Sensors – Ultrasonic-Sensor (HC-RSo<sub>4</sub>).

# 2. Stop for objekter der blokerer for robotten:

Opgrader funktion "followLine", sådan at robotten stopper hvis afstanden til et objekt foran, er under 20cm.

#### Opgave:

- · Opgrader funktionen "followLine", sådan at:
  - Det første der sker i while loopet, er at læse afstanden til det nærmeste objekt.
  - Opgrader betingelserne for at motorerne må køre, sådan at betingelsen nu lyder:
    - Hvis venstre IR-sensor er under tærsklen og afstanden er mere end 20cm, kør venstre motor, ellers stop den.
    - Hvis højre IR-sensor er under tærsklen og afstanden er mere end 20cm, kør højre motor, ellers stop den.

# Karton mit Bohnen "anliefern":

Bevor Sie fortfahren, empfiehlt es sich auch hier das erstellte Programm zu speichern und eine Kopie zu erstellen, um anschließend in der Kopie weiterzuarbeiten.

## 1. Integrieren des Servomotors:

Mitarbeiter können jetzt einen Karton auf dem Roboter platzieren und ihn so programmieren, dass er einer Route folgt – aber er kann den Karton bei Ankunft noch nicht selbst abliefern. Wir wollen ihn daher mit einem Servomotor ausstatten, der die Kartons von der Ladefläche kippen kann, woraufhin der Roboter zurückfährt, um einen neuen Karton abzuholen.

# Aufgabe:

- Gehen Sie nach der Anleitung "Transporter Servo-Motor" vor und integrieren Sie den Servomotor.
- Überprüfen Sie den Servomotoren auf ordnungsgemäße Funktionalität, indem Sie ihn in verschiedene Positionen versetzen.

#### Material:

- · Servomotor.
- · 6 x Stecker/Stecker-Kabel.

# **Cheat sheets:**

· Effectors - Servo-Motor.

# 2. Karton "anliefern":

Jetzt muss der Roboter nur noch den Karton freigeben, wenn er den vorgesehenen Bestimmungsort erreicht hat.

# Aufgabe:

- Erstellen Sie eine Funktion "deliverBox", die mittels des Servomotors die Ladefläche in Kippposition bringt, damit der Karton herunterfällt, woraufhin der Servomotor in seine Ausgangsposition zurückkehrt.
- Überprüfen Sie, dass es entsprechend funktioniert und der Roboter nunmehr einem Streckenverlauf folgen kann, auf dem er am richtigen Ort den beförderten Karton ablädt.

# Varianten der Aufgabe:

Sie können sich gern eigene Ideen für mögliche Erweiterungen ausdenken, z. B. verschiedene LEDs, die anzeigen, welche Aktionen der Roboter derzeit ausführt. Oder der Einsatz eines LDRs (Light Dependent Resistor) zwecks Feststellung, ob sich ein Karton auf der Ladefläche des Roboters befindet.

## Aflever Kassen med Bønner:

Det anbefales at oprette gemme en kopi af programmet, for derefter at fortsætte arbejdet i kopien.

## 1. Integrer servo-motoren:

Medarbejderne kan nu sætte en kasse på robotten og programmere den til at følge en rute, men den kan endnu ikke selv aflevere kassen når den er nået frem. Integrer derfor en servo-motor, der kan anvendes til at tippe kassen af ladet, hvorefter robotten kan køre tilbage efter en ny kasse.

## Opgave:

- Følg manualen "Transporter Servo-Motor" og integrer servo-motoren.
- Test at servo-motoren virker efter hensigten, ved at sætte den i forskellige positioner.

#### Materialer:

- Servo motor.
- · 6stk. han/han ledninger.

## **Cheat Sheets:**

Effectors – Servo-Motor.

#### 2. Aflever kassen:

Anvendt en funktion til at lade robotten aflevere en kasse, når den når frem til det korrekte område.

# Opgave:

- Lav en funktion "deliverBox", der ved hjælp af servo-motoren tipper ladet så kassen falder af, hvorefter servo-motoren kører tilbage til sit udgangspunkt.
- Test at funktionen virker, samt at robotten nu kan køre en rute, hvor den ved det korrekte sted, nu kan aflevere kassen den fragter.

# Ideer til Videreudvikling:

Kom gerne selv på nogle ideer til potentielle udvidelser, det kan eks. være at få forskellige LED'er til at lyse for at indikere hvilke handlingerne robotten pt. er ved at foretage sig. Det kan også være at anvende en LDR (Light Dependent Resistor) til at tjekke for om robotten har en kasse på ladet.

# **Impressum**

#### Das PANaMa-Projekt - Vernetzung von Schule und Arbeitswelt

Bd. 3 Science Camps und Technologieverständnis – Wege zu beruflichen Kompetenzen / Science Camps og teknologiforståelse – metoder til at opnå faglige færdigheder

Hrsg. Stefanie Herzog, Marc Wilken / Projekt PANaMa www.panama-project.eu ISBN 978-3-89088-299-4

**Autoren:** Linda Ahrenkiel, Stefanie Herzog, Annette Jäpelt, Kaj Nedergård Jepsen, Jørgen Christian Larsen, Claus Michelsen, Jacob Nielsen, Ilka Parchmann, Bjarke Kristian Maigaard Kjær Pedersen, Marc Wilken

Übersetzung: R. Marquardt, S. Krebs

Design / Layout: amatik Designagentur, Kiel

Druck: Neue Nieswand Druck GmbH Druckerei, Kiel

Auflage: 500 Exemplare

© IPN Leibniz-Institut für die Pädagogik der Naturwissenschaften und Mathematik

Olshausenstr. 62, 24118 Kiel, Deutschland

Postadresse: IPN, 24098 Kiel, Deutschland

www.ipn.uni-kiel.de

Diese Publikation wird gefördert mit Mitteln des Europäischen Fonds für regionale Entwicklung. Dette bogudgivelse finansieres af midler fra Den Europæiske Fond for Regionaludvikling.

Fotonachweis: Alle Rechte für Abbildungen und Fotos liegen beim IPN Leibniz-Institut, Kiel, mit Ausnahme von: S. 8/9 © Monkey Business/AdobeStock; S. 16/17 © Kadmy/AdobeStock; S. 68/69 © Jelleke Vanooteghem/unsplash, © PaulPaladin/AdobeStock, © stem.T4L/unsplash; S. 70/71 © Stock Rocket/AdobeStock; S. 72/73 © Mimi Thian/unsplash; S. 88/89 Bildquellen: https://commons.wikimedia.org/wiki/File:Skytale.png, https://de.wikipedia.org/wiki/Datei:CipherDisk2000.jpg; S. 102/103 © Monkey Business/AdobeStock; S. 110/111 © stem.T4L/unsplash; S. 112/113 © Maya Kruchancova/AdobeStock; S. 114/115 © phonlamaiphoto/AdobeStock; S. 126/127 © naka/AdobeStock; S. 136/137 © AlesiaKan/AdobeStock; S. 142/143 © ridjam/AdobeStock; S. 152/153 © smolaw11/AdobeStock; S. 184/185 © petrroudny/AdobeStock.

Wir haben uns bemüht, alle Nutzungsrechte zur Veröffentlichung von Materialien Dritter anzugeben bzw. zu erhalten. Sollten im Einzelfall Nutzungsrechte nicht berücksichtigt sein, bitten wir um Kontaktaufnahme unter info@panama-project.eu.









