



Unterrichtsmodul

Automatisierung in Miniatur

Alle Materialien und Kontaktdaten sind auf den Projektwebseiten und im Erasmus+ Projektprofil zu finden:

<https://sites.google.com/campus.ul.pt/hands-on-remote-language/home>

<https://erasmus-plus.ec.europa.eu/projects/search/details/2020-1-DE02-KA226-VET-008295>

Kernteam Entwicklung

- Marion Pellowski und Lorenz Kampschulte, Deutsches Museum, München, Deutschland
- Pedro Reis, Mónica Baptista, Luís Alexandre da Fonseca Tinoca, Lisbon University, Institute of Education, Lissabon, Portugal
- Wojciech Karcz, Adam Zahler, Anna Strzeszewska-Potyrała, Karolina Klimaszewska, Copernicus Science Center, Warschau, Polen
- Miriam Voß, Mike Kramler, Marion Pellowski, Technische Universität München, München, Deutschland

Haftungsausschluss

Dieses Projekt wurde mit Unterstützung der Europäischen Kommission finanziert. Die Verantwortung für den Inhalt dieser Veröffentlichung trägt allein der Verfasser; die Kommission haftet nicht für die weitere Verwendung der darin enthaltenen Angaben.



Kofinanziert durch das
Programm Erasmus+
der Europäischen Union

Impressum

Autoren:

Marion Pellowski, Technische Universität München und Deutsches Museum,
Mike Kramler, Miriam Voß, Technische Universität München,
beide Institutionen München, Deutschland.

Deutsches Museum, München, Deutschland

Layout & Design: Michał Romański

Druck: Februar 2023



Dieses Werk ist lizenziert unter einer Creative Commons Attribution 4.0 International License. Elemente, die als Zitate gekennzeichnet sind, können anderen Lizenzen unterworfen sein.

Inhalt

Danksagung.....	2
1 Einführung.....	3
Kurzer Überblick über das Modul	3
1.1 Ziele.....	4
1.2 Didaktische Begründung.....	5
1.3 Gemeinschaftsgefühl und digitale Zusammenarbeit.....	7
1.4 Kurze Zusammenfassung.....	8
2 Anbindung an den Lehrplan.....	11
3 Überblick über die Modulsequenz für das Modul „Automatisierung in Miniatur“	12
4 Inhalt: Lernsequenz für das Unterrichtsmodul „Automatisierung in Miniatur“	13
4.1 Einheit 1.a: Einführung in das Thema	13
4.2 Einheit 1.b: Einführung in grundlegende Schaltungen und die Programmierung des Arduino	15
4.3 Einheit 2.a: Online-Simulation.....	18
4.4 Einheit 2.b: Servomotor	19
4.5 Einheit 3.a: Drehscheibe	21
4.6 Einheit 3.b: Lichtschranke	23
4.7 Einheit 4: Erweiterungen des Automatisierungsprozesses.....	25
4.8 Einheit 5: Zusammenarbeit der Schüler:innen	27
4.9 Einheit 6: Digitale Zusammenarbeit.....	28
4.10 Einheit 7: Automatisierung im sozialen Kontext	29
4.11 Einheit 8: Andere Anwendungen der Drehscheibe.....	32
4.12 Allgemeine Hinweise zu Videokonferenzen und praktischer Arbeit in Distanzsituationen	34
5 Möglichkeiten modularer Nutzung	36
Ablauf 1.....	36
Ablauf 2.....	36
6 Anhang: Einblicke in die interaktive multimediale Anleitungsseite	37

Danksagung

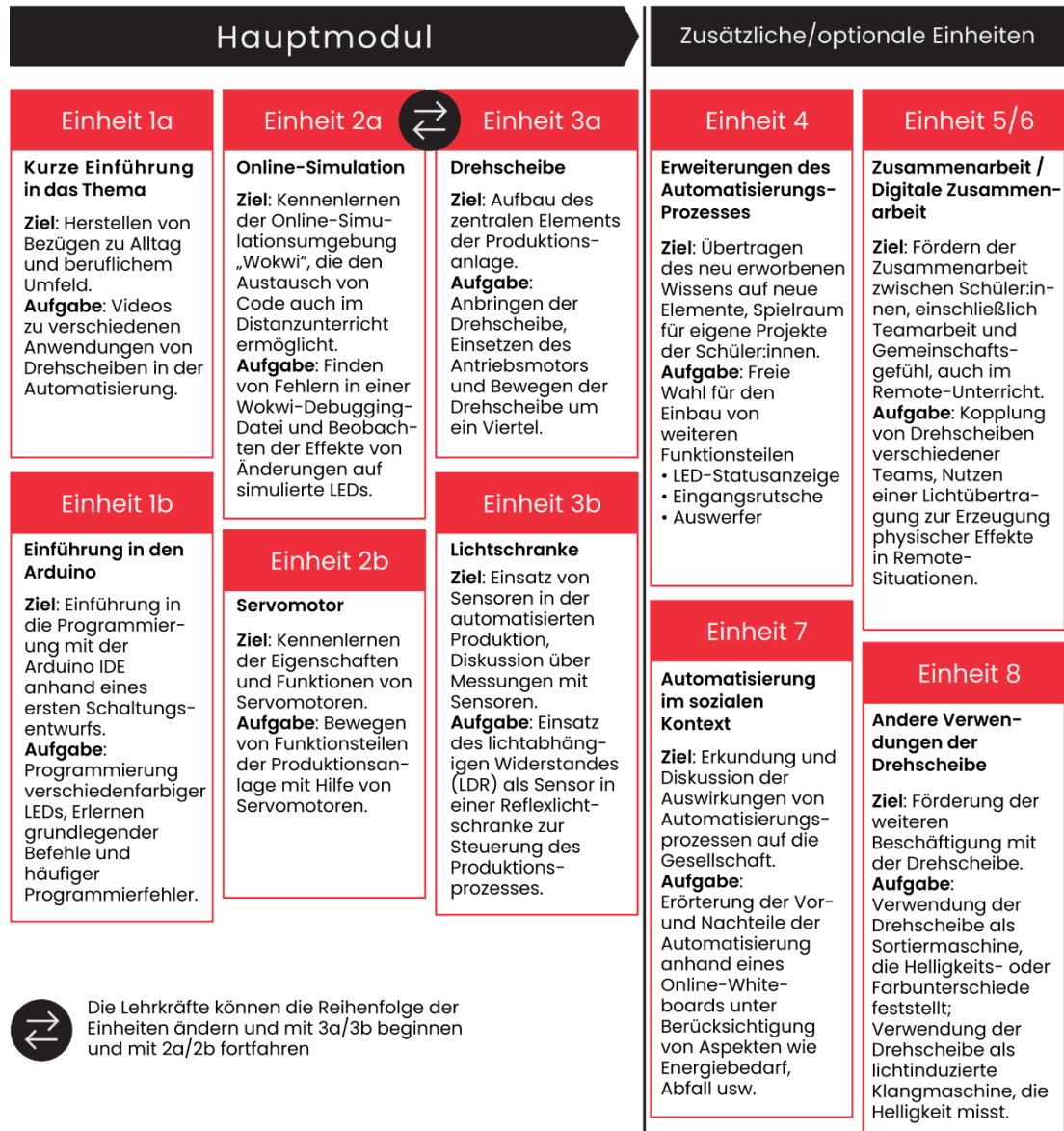
Wir danken den Lehrkräften der Städtischen Berufsschule für Informationstechnik und Fachinformatik Systemintegration München sehr herzlich für die Zusammenarbeit und Beratung und für die konstante und immer konstruktive Unterstützung, die für die Entwicklung dieses Unterrichtsmoduls ausgesprochen hilfreich war. Wir bedanken uns bei den Jugendlichen, die das Modul mit uns getestet haben, für ihr Feedback und ihren Input, der in die Optimierung des Moduls eingeflossen ist. Ein herzlicher Dank gilt ebenso allen Lehrkräften, die sich an unserer Bedarfsanalyse beteiligt haben, für die sehr wertvollen Beiträge und Kommentare zu diesem Projekt.

Kurzer Überblick über das Modul

Unterrichtsmodul

**Automatisierung
in Miniatur**

Die Schüler:innen programmieren eine kleine Produktionsanlage und setzen sie in Gang. Die automatisierte Befüllungsanlage ist aus einfachen Materialien gefertigt und kann von den Teilnehmenden mit nach Hause genommen werden. Die Jugendlichen können als Team zusammenarbeiten, auch in Remote-Situationen.





In diesem Modul ermöglicht eine Mini-Produktionsanlage es Schüler:innen, ein reales mechanisches System zu programmieren. Die Jugendlichen beziehen dabei Effekte ein und lösen Probleme, die in einer rein virtuellen Umgebung nicht auftreten. Damit sie einen funktionierenden Produktionsprozess gestalten können, ist das Zusammenspiel zwischen der Programmierung und den mechanischen Funktionsteilen der Mini-Produktionsanlage zentral. Eine entscheidende Idee dieses Moduls für den Distanzunterricht besteht darin, ein hochwertiges und teures Produkt für den Unterricht durch eine einfachere und kostengünstigere Version zu ersetzen, die Lehrkräfte ihren Schüler:innen mit nach Hause geben können.

Die Jugendlichen lernen am Beispiel der Mini-Produktionsanlage Aufbau und Funktionsweise eines komplexen technischen Systems kennen. Indem sie Servomotoren als Aktoren ansteuern und einen Lichtsensor zur Steuerung des Befüllvorgangs nutzen, programmieren und steuern sie die kleine Fertigungsanlage. Sie programmieren textbasiert einen Mikrocontroller, den Arduino, mit dem sie auch Messwerte erfassen können. Die Schüler:innen testen ihre Programme an dem realen Beispiel der Mini-Automatisierungsanlage und wenden bei ihren Tests Methoden der Fehlersuche an. Dabei müssen sie auch Umgebungseinflüsse berücksichtigen (z. B. den Einfluss von Umgebungslicht auf die Messwerte eines Lichtsensors).

Lehrkräfte können zwischen zwei verschiedenen Niveaus der Programmierung wählen und das Modul so an unterschiedliche Vorkenntnisse anpassen: einem Einstiegsniveau, das den Betrieb der Miniproduktionsanlage mit nur wenigen, einfachen Codezeilen ermöglicht – und einem Niveau für Fortgeschrittene, welches eine Einführung in die objektorientierte Programmierung (Definition von Objekten, Struktur von Objekten, Verwendung von praxisbezogenen Klassen und Objekten) erlaubt. Lehrkräfte können das Modul außerdem für einen Transfer vorhandener Kenntnisse objektorientierter Programmierung auf ein Anwendungsbeispiel nutzen. Die Programmierung des gesamten Automatisierungsprozesses ist jedoch auch ohne Kenntnisse der objektorientierten Programmierung sehr gut möglich.

Automatisierung und deren gesellschaftliche Konsequenzen betreffen viele Berufsschüler:innen direkt, als Teil ihrer künftigen Berufsfelder. In einer abschließenden Unterrichtsstunde werden die Schüler:innen dazu angeregt, über gesellschaftliche Fragen im Zusammenhang mit der Automatisierung zu diskutieren und nachzudenken, um ihre eigene berufliche Zukunft in einem übergeordneten Kontext zu betrachten. Angesichts des gesellschaftlich hoch relevanten Themas setzen sich die Jugendlichen mit verschiedenen Vor- und Nachteilen der Automatisierung für unsere Gesellschaft auseinander, einschließlich ökologischer und arbeitsrelevanter Aspekte.

1.2 Didaktische Begründung



Das Projekt nutzt einfach zu bedienende Online-Tools wie eine Online-Simulationsumgebung und ein Online-Whiteboard, um den Distanzunterricht zu vereinfachen. Die Lehrkräfte können diese Online-Werkzeuge ebenso wie die Hands-on-Experimente sowohl im Präsenz- als auch im Distanzunterricht einsetzen. So lässt sich für verschiedene und wechselnde Unterrichtssituationen eine fortlaufende gemeinsame Basis schaffen.

Für die Einführung in das Thema und die Arbeit mit der Mini-Produktionsanlage sind Videokonferenzen eine geeignete und wichtige Methode, insbesondere zu Beginn. Im weiteren Verlauf bleiben Videokonferenzen von Bedeutung, um Kontakt mit den Schüler:innen zu halten und sie bei ihrer Arbeit zu unterstützen. Dabei sind verschiedene Formen möglich: Videokonferenzen mit der ganzen Klasse, Videokonferenzen mit Teilgruppen in Breakout-Rooms und ein Supportraum, in dem die Lehrkraft zu vorher festgelegten Zeiten für Anfragen bereitsteht. Das Modul setzt Videokonferenzen auch ein, um die Zusammenarbeit zwischen Schüler:innen zu fördern (s. 1.3).

Nach einer Einführung in die Grundlagen können die Jugendlichen mit Hilfe einer multimedialen Anleitungssseite eigenständig weiterarbeiten. Sie können selbst entscheiden, wie sie vorgehen wollen. Die Rolle der Lehrkraft ist es, währenddessen Ansprechpartner für gegebenenfalls auftretende Probleme zu sein. Die Nutzung der multimedialen Anleitungssseite und die dadurch mögliche selbstständige Arbeit der Schüler:innen erleichtert das Vorgehen sowohl in Präsenz- als auch in Remote-Situationen. Die Kombination digitaler Werkzeuge, d. h. der Online-Tools, der verschiedenen Videokonferenz-Formen und der multimedialen

Anleitung, erlauben den Einsatz des Moduls für vollständigen Distanz- ebenso wie für Hybridunterricht. Die Lehrkräfte agieren in diesem Prozess als Lernbegleiter, sei es in Distanz- oder in Präsenzlernformen.

Automatisierung ist ein wichtiger Teil moderner Gesellschaften und bestimmt einen großen Teil unseres Lebens. Es ist daher einfach, Bezüge zu realen Automatisierungsprozessen und zum Alltag der Schüler:innen herzustellen. Mit dem Automatisierungsprozess sind Bezüge zu verschiedenen Berufsfeldern möglich. Die notwendigen Programmier- und Steuerungsprozesse sind Teil des Lehrplans verschiedener Ausbildungsberufe. Im Vergleich zu industriell gefertigten Produktionsanlagen für Schulen besteht die Mini-Produktionsanlage aus einfachen Materialien, bei denen Einzelteile relativ leicht austauschbar sind. Dies ist von entscheidender Bedeutung, um den Schüler:innen die Materialkits im Distanzunterricht zur Verfügung zu stellen. Darüber hinaus erschien es wichtig, das Set attraktiv zu gestalten, um die Möglichkeit zu eröffnen, dass die Jugendlichen auch in ihrer Freizeit gerne damit weiterarbeiten.

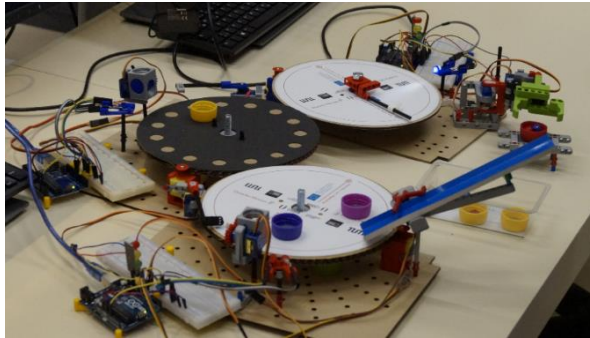
Eine flexible Nutzung der Mini-Produktionsanlage oder ihre Umgestaltung für andere Anwendungen erlauben, den Grad der Komplexität an die jeweilige Zielgruppe anzupassen – und somit auch einen Einsatz der Anlage in einem sehr heterogenen europäischen Kontext.

Für weitere Details zu Entwicklung und Implementation s. Begleitheft „Leitfaden 2023“. Der Kategorisierung im Leitfaden entsprechend (s. Kapitel 2), kombiniert dieses Unterrichtsmodul lokale und reale Experimente (die z. B. einen Versand oder das Abholen von Materialien erfordern) mit Remote-Experimenten und virtuellen Experimenten, die aufgrund ihrer Struktur eine bessere Grundlage für ein konzeptuelles Verständnis bilden können.



In Team-Arbeit Hands-on-Experimente durchzuführen, hat ein großes Potential, das Gemeinschaftsgefühl der Schüler:innen untereinander zu fördern. Einige Aufgaben dieses Moduls sind daher vor allem auf kollaboratives Lernen ausgerichtet.

Dazu gehört insbesondere die Aufgabe, verschiedene der Mini-Produktionsanlagen miteinander zu koppeln und zusammenarbeiten zu lassen.



Die Jugendlichen können ihre Drehscheiben kombinieren, indem sie den Auswerfer nutzen, um einen Becher von einer Drehscheibe auf eine andere zu befördern.

Diese Kopplung der Drehscheiben ist möglich, auch wenn noch keines der Teams einen vollständigen Automatisierungsprozess programmiert hat. In einem solchen Fall nutzen die Schüler:innen einfach auf jeder der Drehscheiben ein anderes Funktionsteil und etablieren einen gemeinsamen Transfer-Mechanismus zwischen den verschiedenen Scheiben.

Mit der Online-Simulationsumgebung können die Schüler:innen ihre Programme auf sehr direktem Weg austauschen und zusammen an den Programmieraufgaben arbeiten. So bleibt eine enge Kooperation selbst im Distanzunterricht möglich.



Dieses Modul enthält zudem eine außergewöhnliche Möglichkeit der Zusammenarbeit auf Distanz: Die Jugendlichen können in Videokonferenzen über eine Lichtübertragung miteinander interagieren. Die Lichtübertragung ermöglicht es, physische Objekte an einem anderen, auch weit entfernten, Standort zu bewegen. Die Lichtübertragung funktioniert folgendermaßen: Der erste Team-Partner leuchtet – z. B. in München – mit einer LED auf die Webcam für die Videokonferenz. Vor dem Laptop des zweiten Team-Partners ist ein Lichtsensor so angebracht, dass er das Signal erkennen kann. Als Reaktion auf das Lichtsignal wird am Standort des zweiten Team-Partners, z. B. in Lissabon, ein Servomotor gedreht, der ein Funktionsteil der Produktionsanlage bewegt.

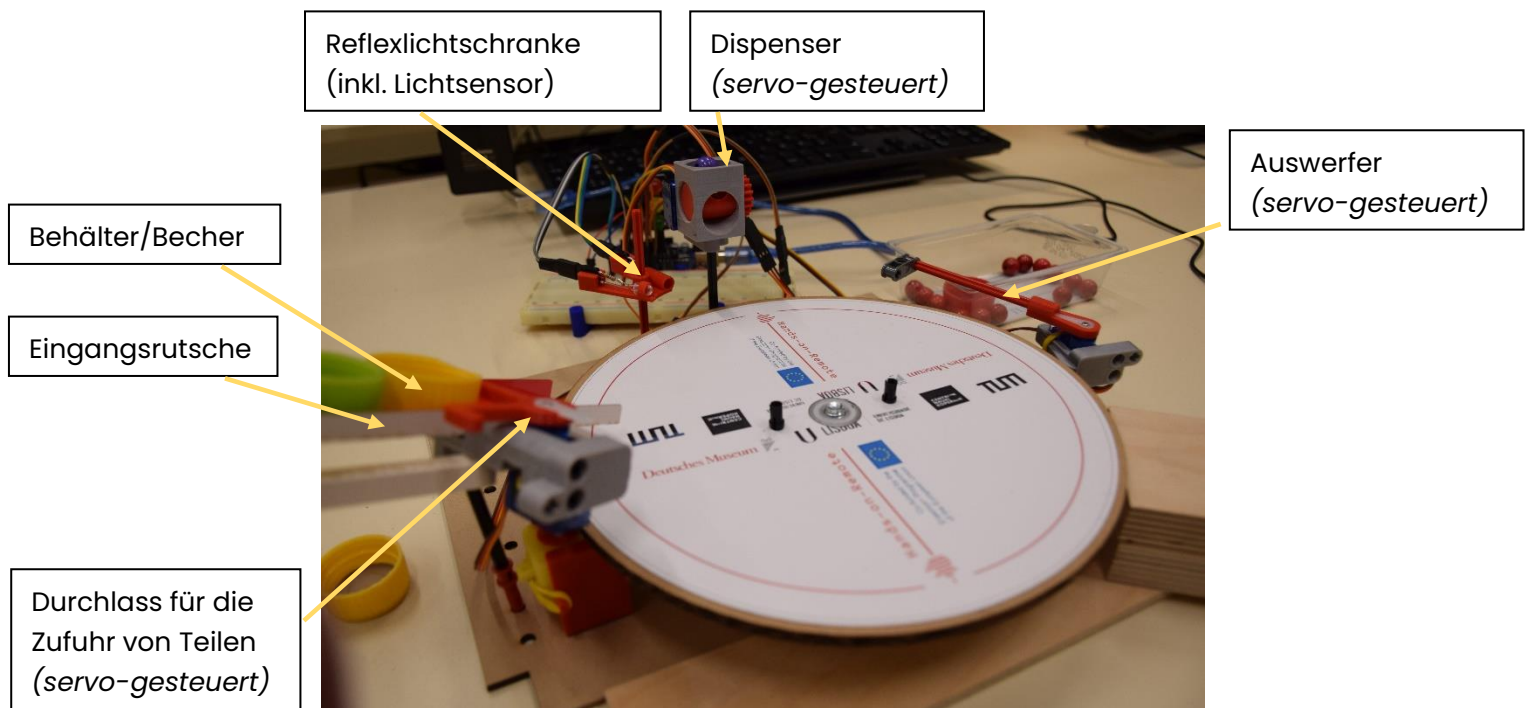
Dann kann der zweite Team-Partner seinerseits das Lichtsignal weitergeben usw. Ziel dieser Kollaborationsform ist es, in Distanzsituationen die motivationalen Vorteile von Team-Arbeit zu nutzen und negativen Effekten einer Isolation entgegenzuwirken.

Es ist Teil des Konzepts, diesen Ansatz einzusetzen, um ein Gefühl der Zusammengehörigkeit zu schaffen, trotz physischer Distanz – und um einfach Freude bei den Jugendlichen auszulösen.

Es ist eine sehr einfache Herangehensweise, die den Vorteil hat, leicht verständlich, low-cost und schnell implementierbar zu sein. Der Ansatz bezieht die Vorerfahrungen der Lehrkräfte mit ein und lässt sich daher auch für Lehrkräfte gut anpassen, die nur wenig Erfahrung mit technischen Medien im Distanzunterricht haben.

1.4 Kurze Zusammenfassung

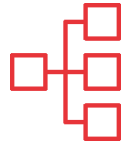
Das Unterrichtsmodul "Automatisierung in Miniatur" ermöglicht es Schüler:innen, eine kleine Produktionsanlage selbst zu programmieren und in Gang zu setzen. Ein Ampelmodul als Statusanzeige, eine Lichtschranke und eine Drehscheibe dienen als Elemente, die zusammen mit verschiedenen beweglichen Funktionsteilen zu einer Mini-Produktionsanlage werden.



Die Schüler:innen können die Drehscheibe als Befüllungsanlage steuern – dabei erkennt die Lichtschranke einen Behälter auf der Drehscheibe, die Statusanzeige zeigt eine Zustandsänderung durch Wechsel von Rot auf Grün an und ein Dispenser wird genutzt, um etwas in einen Behälter zu füllen.

Der Produktionsprozess ist sensorabhängig – eine Reflexlichtschranke erkennt einen vorbeifahrenden Behälter und gibt daraufhin den Impuls zur Befüllung des Bechers.

Struktur der Unterrichtseinheiten



Die Lehrkräfte können verschiedene Abläufe und Ansätze für die Unterrichtsstunden wählen. Die Unterrichtseinheiten bauen aufeinander auf, gestatten aber eine Aufteilung in ein Einstiegs- und ein Fortgeschrittenenprogramm. Einige motivierende Einblicke in die industrielle Anwendung von Drehscheiben versetzen die Klasse zu Beginn in eine fiktive Fabrik. Für ihre Mini-Fabrik können die Teams wählen, was sie in ihrem eigenen fiktiven Szenario "produzieren". Als Einstieg in die Programmierung mit dem Arduino schalten die Schüler zunächst die LEDs der Statusanzeige ein und aus. Dazu arbeiten sie sowohl mit einem realen als auch mit einem virtuellen Ampelmodul. Für die Arbeit mit der Mini-Produktionsanlage starten alle Teams zusammen und setzen z. B. die Drehscheibe in Bewegung, bauen den Dispenser an und programmieren ihn, abhängig von den Signalen des Lichtsensors. Danach können die Schüler:innen entscheiden, welche zusätzlichen Elemente sie ihrem Produktionsprozess hinzufügen wollen.

Programmierung

Der Schwierigkeitsgrad kann auch auf Ebene der Programmierung gewählt werden: Wenn eine Lehrkraft das Modul für eine Einführung nutzen möchte, schreiben die Schüler:innen nur wenige Zeilen Code in ein weitgehend vorgegebenes Programm. Wenn eine Lehrkraft das Modul für fortgeschrittene Programmierungen verwenden möchte, haben die Schüler:innen zusätzliche Aufgaben, z. B. die Definition von Klassen.

Materialien für das Unterrichtsmodul

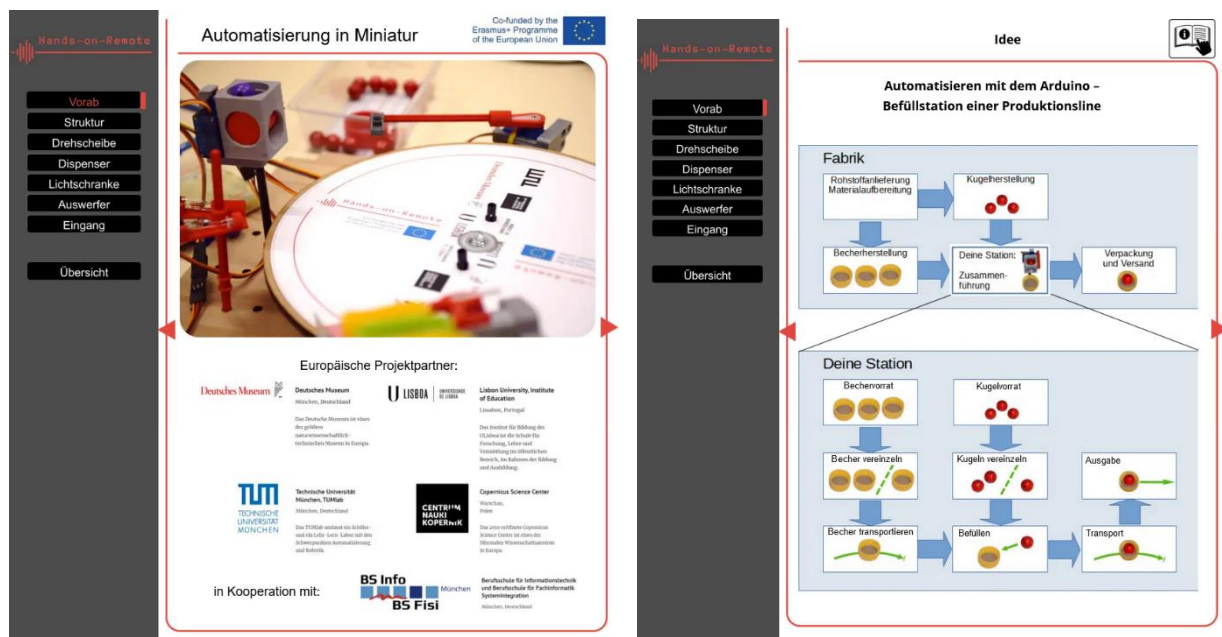
Integraler Bestandteil des Unterrichtsmoduls „Automatisierung in Miniatur“ sind verschiedene Dateien und Materialien, die auf folgenden Seiten herunterzuladen sind:

Download-Seite des Projekts:

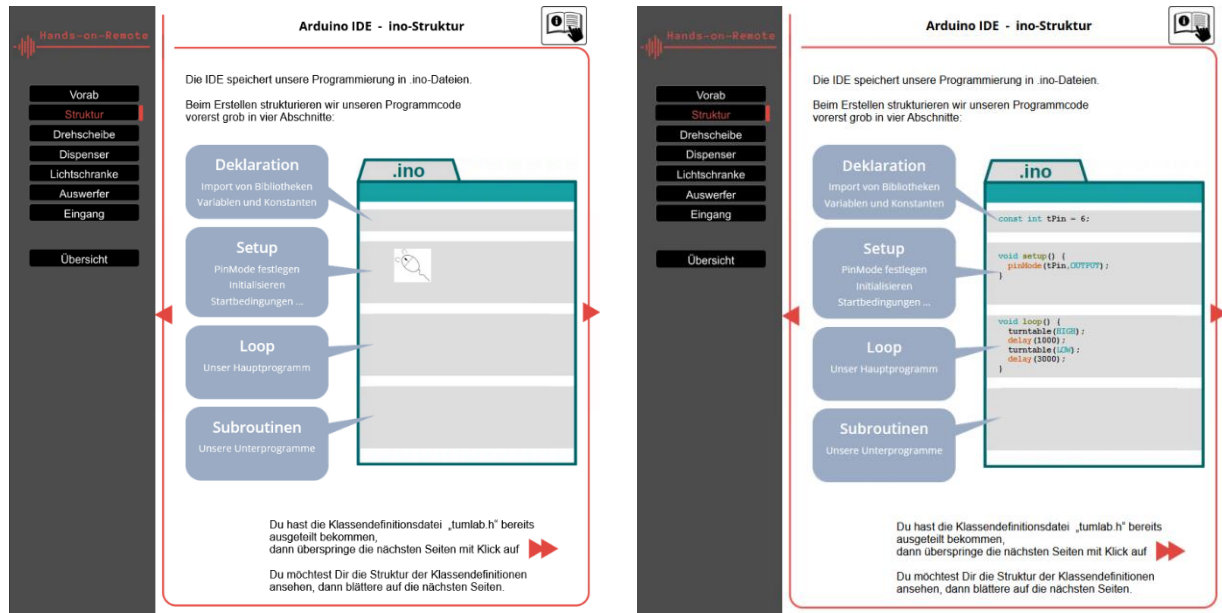
<https://sites.google.com/campus.ul.pt/hands-on-remote-de/downloads>

Projektseite Deutsches Museum: <https://www.deutsches-museum.de/forschung/forschungsinstitut/projekte/detailseite/era-smus-hands-on-remote>

- Präsentationsdatei für die Lehrkraft, *Präsentations-Datei für Lehrkraft (pptx)*, sowie ein Video zum Stromkreis, *Video_Stromkreis (mp4)*
- Interaktive multimediale Anleitungssseite für die Schüler:innen, Schritt-für-Schritt-Anleitung für die selbstständige Arbeit mit der Mini-Produktionsanlage, inklusive Videos und Animationen, *Interaktive Anleitungssseite für Schüler DE (html)*



- Arbeitsblätter für die Einheit „Automatisierung im sozialen Kontext“ und Link zum Online-Whiteboard, *Arbeitsblätter Automatisierung im sozialen Kontext (pdf)*
- 3D-Druck und Lasercut-Dateien für das Materialkit, *Materialkit (Liste, stl- und Lasercut-Dateien)*
- Eine Datei mit häufigen Fehlern und Schwierigkeiten, *Troubeshooting_Automatisierung_in_Miniatur (pdf)*



Einzelschritte zum Bau und zur Programmierung der gesamten Anlage gehen über den Rahmen der vorliegenden Beschreibung des Unterrichtsmoduls hinaus und finden sich in den oben genannten Begleitdateien. Für die Mini-Produktionsanlage sind Aufbau und Programmierung insbesondere in der multimedialen Anleitungssseite dargestellt. Einen ersten Einblick in Aufbau und Gestaltung der Anleitungssseite geben Screenshots im Anhang (s. Kap. 6, Anhang: Einblicke in die multimediale Anleitungssseite). So lässt sich eine bessere Vorstellung gewinnen, wie mit dem Arduino und der Anlage auch remote gearbeitet werden kann.

2 Anbindung an den Lehrplan

Die Gestaltung des Moduls, insbesondere der Mini-Produktionsanlage und der Begleitmaterialien, basiert auf Bedarfsanalysen mit Lehrkräften verschiedener Berufsschulen. Das Modul hat Anknüpfungspunkte zum Lehrplan verschiedener Ausbildungsberufe und zum Unterricht der Fachoberschule. Berufe, für die das Modul eine Rolle spielen kann, sind zum Beispiel Informatiker:in, Elektroniker:in, Automatisierungstechniker:in, Industrieelektriker:in, Softwareentwickler:in. Die Lerneinheit zu den gesellschaftlichen Aspekten der Automatisierung hat einen Bezug zur angewandten Ethik (Technikethik). Schlüsselwörter für die Themen sind z. B. Systeme und Prozesse, komplexe technische Systeme, Steuerungstechnik und objektorientierte Programmierung. Die Anbindungen an den Lehrplan existieren generell für die Jahrgangsstufen 10–13.

3 Überblick über die Modulsequenz für das Modul „Automatisierung in Miniatur“

Einheit 1.a Einführung in die Verwendung von Drehscheiben in der Automatisierung und das Ziel, einen Automatisierungsprozess zu miniaturisieren. (15 min)		Einheit 1.b Einführung in die Arbeit mit dem Arduino anhand eines ersten Schaltungsentwurfs und eines Programmierbeispiels mit verschiedenfarbigen LEDs. (30-45 min)		
Einheit 2.a Einführung einer Online-Simulationsumgebung für die Programmierung des Arduino, um in einer Distanzsituation im Team an der Programmierung arbeiten zu können. Entwurf einer simulierten Schaltung für einen Servomotor und Programmierung des Servomotors. (15 min)		Einheit 2.b Einführung in Servomotoren, einschließlich Schaltung und Programmierung eines realen Servomotors. (30 min)		
Einheit 3.a Aufbau der Drehscheibe, Schaltungen, Programmierung des Getriebemotors. (20 min)		Einheit 3.b Einsatz der Lichtschranke, einschließlich Schaltung, Programmierung und Anzeige der Werte auf dem Plotter. (25 min)		
Einheit 4 Die Schüler:innen arbeiten selbstständig und wählen, mit welchen Elementen sie den Automatisierungsprozess erweitern. (mind. 45 min)				
Kurzversion	Variation	Variation	Variation	Lange Version
Drehscheibe Lichtschranke Dispenser	Drehscheibe Lichtschranke Dispenser LED-Statusanzeige für den Dispenser	Drehscheibe Lichtschranke Dispenser Auswerfer	Drehscheibe Lichtschranke Dispenser Auswerfer Eingangsrutsche	Drehscheibe Lichtschranke Dispenser Auswerfer Eingangsrutsche LED-Statusanzeige für die Aktoren
Einheit 5 Zusammenarbeit der Schüler:innen: Koppeln der Drehscheiben verschiedener Teams. (mind. 45 min)				
Einheit 6 Digitale Zusammenarbeit: Lichtübertragung nutzen, um an entfernten Standorten physische Effekte auszulösen. (flexible Dauer)				
Einheit 7 Automatisierung im sozialen Kontext. (mind. 45 min)				
Einheit 8 Andere Verwendungen der Drehscheibe (s. Tabelle unten) (mind. 45 min)				
Sortiermaschine		Lichtplattenspieler		
Drehscheibe Lichtschranke LED-Statusindikator Helligkeit/Farbe Auswerfer zum Sortieren		Drehscheibe mit Papierauflagen mit Schwarz-Weiß-Mustern oder Graustufen Lichtsensoren als Messkopf Lautsprecher (nicht im Materialkit enthalten)		

4 Inhalt: Lernsequenz für das Unterrichtsmodul „Automatisierung in Miniatur“

4.1 Einheit 1.a: Einführung in das Thema

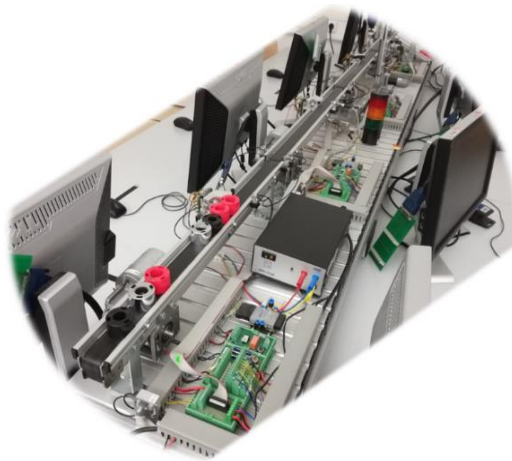


Thema: Verschiedene Praxisbeispiele für Automatisierungsprozesse

Zielsetzung: Einführung in das Thema Automatisierung und das Ziel der Miniaturisierung eines Automatisierungsprozesses.

Dauer: ca. 15 min

Hintergrund/Einführung in die Idee:



Viele Berufsschüler:innen werden in ihrem Berufsleben mit Automatisierungsprozessen zu tun haben – sei es, weil sie in Zukunft selbst Produktionsanlagen programmieren oder weil sie mit solchen Systemen arbeiten werden – zum Beispiel in der Lebensmitteltechnik.

Insgesamt bietet das Thema Automatisierung viele Praxisbezüge aus der realen Welt.

Die Idee der Mini-Automatisierung geht auf eine größere Fertigungsstraße im TUMlab zurück. In diesem Experimentierlabor wird der entsprechende „Produktionsprozess“ von Schüler:innen programmiert und automatisiert. Lehrkräfte haben immer wieder den Wunsch geäußert, ob es nicht auch die Möglichkeit einer Produktionsstraße im kleineren Format gibt, welche die Jugendlichen in der Schule – oder nun auch zu Hause – programmieren können. So entstand die Idee, mit dem Arduino einen Automatisierungsprozess in kleinem Maßstab zu ermöglichen.

Format: Online über Videokonferenz oder im Klassenzimmer



Materialien für die Lehrkraft: PowerPoint-Präsentation, einschließlich Links zu YouTube-Videos

Materialien für die Schüler:innen: Keine

Inhalt:

Aktivitäten der Lehrkräfte

Anhand von Ausschnitten aus verschiedenen YouTube-Videos geben Lehrkräfte Einblicke in reale Praxisbeispiele für automatisierte Industrieanlagen. Die Videos zeigen, welche wichtige Rolle Drehscheiben in Automatisierungsprozessen spielen. Die Lehrkräfte stellen die Idee vor, anstatt eines Förderbandes eine Drehscheibe für den miniaturisierten Automatisierungsprozess einzusetzen. Die Drehscheibe lässt sich in verschiedenen Formen verwenden:

- entweder als Förderscheibe eines Automatisierungssystems mit mehreren Arbeitsschritten
- oder als einzelne Drehscheibe, z. B. als einzelne Bearbeitungsstation einer kompletten Produktionslinie, wie etwa einer Befüllungsstation.

Außerdem zeigt eines der Videos eine selbstgebaute Maschine, die mit Hilfe eines automatisierten Prozesses eine Pizza auf einem Drehteller belegt. Die Schüler:innen können anhand dieser Idee Bezüge zu ihrem eigenen Alltag herstellen.



Aktivitäten der Schüler:innen

Die Jugendlichen sehen sich die Videos an.

4.2 Einheit 1.b: Einführung in grundlegende Schaltungen und die Programmierung des Arduino

Themen: Schaltungsentwurf, grundlegende Programmierbefehle (Arduino IDE), häufige Programmierfehler

Zielsetzung: Die Schüler:innen lernen zum ersten Mal aktiv einen Mikrocontroller kennen. Sie erlernen grundlegende Arbeitsweisen zum Entwurf einer Schaltung und zur Programmierung eines Mikrocontrollers. Für einen einfachen Einstieg arbeiten die Jugendlichen mit LEDs oder dem Ampelmodul, das später als Statusanzeige für die Automatisierungsprozesse dienen kann.

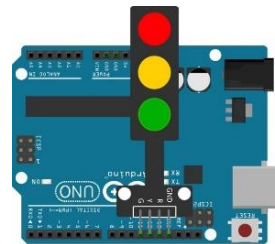
Hintergrund/Einführung in die Idee: Mikrocontroller sind in vielen technischen Alltagsgeräten eingebaut, in denen sie jedoch oft unsichtbar bleiben. Ein Mikrocontroller benötigt für seine Funktionen sowohl eine Schaltung als auch eine Programmierung, was den Lehrkräften die Möglichkeit gibt, beide Themen einzuführen. Für die Schüler:innen ist es motivierend, dass sie mit ihrer Arbeit direkt einen sichtbaren Output erzeugen können. Wenn die Jugendlichen nicht das gewünschte Ergebnis erzielen, können sie selbst oder ihre Lehrkräfte Fehler schnell erkennen und korrigieren.

Dauer: ca. 30 min, mit Einsatz des Videos über Stromkreise und Ausprobieren des DC Virtual Lab, ca. 45 min.



Materialien für die Lehrkraft: PowerPoint-Präsentation

Materialien für die Schüler:innen: Arduino, Arduino IDE, USB-Hub, LEDs oder Ampelmodul. Falls die Lehrkraft diese verwenden möchte: Video über Stromkreise, portable Arduino IDE-Version und Dateien auf einem USB-Stick.



Wichtig zu beachten: Der Arduino wird über einen USB-Hub an den Computer angeschlossen. Bei jedem Aufbau einer Schaltung wird der Arduino vom USB-Hub getrennt. Dieses Vorgehen dient dazu, den Computer bei einem möglichen Kurzschluss zu schützen.

Format: Online per Videokonferenz oder in einem Computerraum an der Schule

Inhalt:

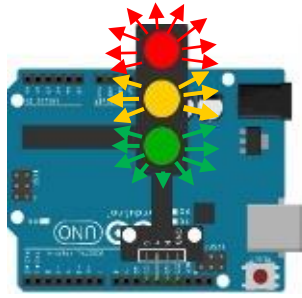
Aktivitäten der Lehrkräfte

- Die Lehrkräfte machen die Schüler:innen mit dem Arduino und den grundlegenden Schritten von Arduino-Projekten vertraut.
- Die Lehrkräfte stellen die textbasierte Programmierumgebung Arduino IDE vor.
- *Abhängig vom Wissensstand der Schüler:innen:*
 - Die Lehrkräfte erinnern die Jugendlichen an den einfachen Stromkreis, mit dem sie eine LED zum Leuchten bringen.
 - Die Lehrkräfte leiten die Schüler:innen beim Aufbau der Schaltung auf dem Arduino für die LEDs oder das Ampelmodul an.
 - Die Lehrkräfte fragen die Jugendlichen nach ihren Fortschritten und bitten darum, die blinkenden LEDs zu zeigen.
 - Die Lehrkräfte helfen den Schüler:innen bei technischen Problemen und nutzen hierfür eine Datei, in der häufige Fehler und Schwierigkeiten aufgeführt sind.
 - Im Distanzunterricht lassen die Lehrkräfte die Jugendlichen in einer Videokonferenz ihren Bildschirm teilen, um bei Programmierproblemen zu helfen.
 - Die Lehrkräfte stellen eine Verbindung zwischen der Programmierung des Ampelmoduls und der Automatisierung her, indem sie ihren Schüler:innen die Verwendung von LEDs bzw. Ampeln als Statusanzeigen in Automatisierungssystemen erklären.



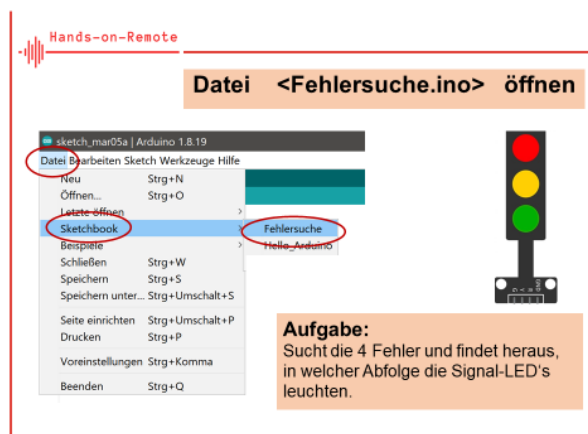
Aktivitäten der Schüler:innen

- Die Jugendlichen sehen sich ein Video über Stromkreise an und bauen online mit dem DC Virtual Lab ihren eigenen Stromkreis (https://phet.colorado.edu/sims/html/circuit-construction-kit-dc-virtual-lab/latest/circuit-construction-kit-dc-virtual-lab_de.html).
- Die Schüler:innen bauen ihre erste Schaltung mit einer LED oder dem Ampelmodul auf dem realen Arduino auf.
- Die Jugendlichen schließen den Arduino an und laden den ersten Code hoch, um eine LED blinken zu lassen.
- Die erste Programmieraufgabe für die Schüler:innen besteht darin, den Programmcode zu ändern, um die Blinkzeit einer LED zu verändern.
- Die Jugendlichen bewältigen technische Probleme anhand einer Datei, in der häufige Fehler und Schwierigkeiten aufgeführt sind.
- Falls nötig, können sie ihre Programmierung mit ihren Lehrkräften besprechen.



- Die Aufgabe der Schüler:innen ist nun, 3 LEDs gleichzeitig blinken zu lassen. Die Lehrkraft kann an dieser Stelle zur objektorientierten Programmierung übergehen: 3 LEDs = mehrere Objekte der gleichen Art. In der objekt-orientierten Programmierung kann man die LEDs daher in einer Klasse zusammenfassen.

Schüler:innen, die bereits über vertiefte Kenntnisse der objektorientierten Programmierung verfügen, können eigene Klassen definieren. Alle anderen können vordefinierte Klassendefinitionen verwenden. Die Verwendung vorgegebener Klassendefinitionen vereinfacht die Programmierung für den gesamten weiteren Prozess.



- Die Schüler:innen vertiefen ihre Kenntnisse der Programmiersprache, indem sie in einer vorgefertigten Fehlersuch-Datei nach typischen Semantik-/Syntaxfehlern suchen.

4.3 Einheit 2.a: Online-Simulation

Die Lehrkräfte können entweder in dieser Reihenfolge fortfahren oder zuerst die Drehscheibe (3.a) und die Lichtschranke (3.b) einführen.

Themen: Kennenlernen und Nutzen einer Online-Simulationsumgebung, die es ermöglicht, häufige Programmierfehler zu erkennen und Programme zwischen Team-Mitgliedern auszutauschen, erste Programmierung eines Servomotors

Zielsetzung: Die Schüler:innen lernen die Online-Simulationsumgebung "Wokwi" kennen und nutzen sie. Dies kann besonders wichtig werden, wenn sie in einer Distanzsituation innerhalb eines Teams Programme austauschen wollen. Sie übertragen Codes zwischen Wokwi und der Arduino IDE.

Hintergrund/Einführung in die Idee: Seitens der Lehrkräfte gab es den Wunsch, die gleichen Methoden im Präsenz- und Online-Unterricht zu verwenden. Lehrkräfte können die Wokwi-Online-Simulation sowohl vor Ort als auch im Distanzunterricht einsetzen. Die Schüler:innen können damit ihr erworbenes Wissen über die Fehlersuche in der Programmierumgebung festigen. Da in der Online-Simulation Effekte direkt sichtbar sind, kann der motivierende Faktor, einen sofortigen Output der eigenen Arbeit zu sehen, für die Jugendlichen auch in der Online-Umgebung beibehalten werden.

Der Übergang von der Simulation mit der Fehlersuch-Datei zum Ampelmodul mit denselben Fehlern ermöglicht es den Schüler:innen, reale Objekte in der Simulation zu erkennen. Darüber hinaus können die Jugendlichen ein neues Element zu ihrem Code hinzufügen und dieses Wissen später auf den realen Arduino übertragen. Außerdem bietet die Online-Simulation große praktische Vorteile für den Austausch von Codes.

Dauer: ca. 15 min



Materialien für die Lehrkraft: PowerPoint-Präsentation

Materialien für die Schüler:innen:

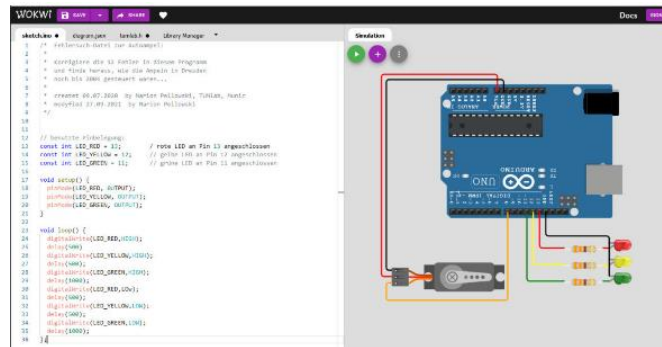
Vorgefertigte Wokwi-Datei zur Fehlersuche
(<https://wokwi.com/projects/328915985532715602>),
Internetverbindung

Format: Die Schüler:innen arbeiten selbstständig mit der Wokwi-Datei, entweder online und mit Unterstützung der Lehrkraft per Videokonferenz oder in einem Computerraum an der Schule.

Inhalt:

Aktivitäten der Lehrkräfte

- Verteilen der Wokwi-Datei und Erklären der Aufgabenstellung



Aktivitäten der Schüler:innen

- Die Jugendlichen finden Fehler in einer Wokwi-Fehlersuch-Datei und können die Auswirkungen ihrer Änderungen auf simulierte LEDs beobachten.
- Die Schüler:innen erweitern die Simulation um einen Servomotor.

4.4 Einheit 2.b: Servomotor

Themen: Kennenlernen eines Servomotors und seiner Funktion, Programmierung der Winkelpositionen eines Servomotors.

Zielsetzung: Die Schüler:innen lernen die Eigenschaften und Funktionen von Servomotoren kennen. In der miniaturisierten Produktionsanlage setzen die Schülerteams Servomotoren als Aktoren ein, um z. B. Zahnräder oder die Schranke für den Durchlass zu bewegen. Generell ist der Einsatz von Servomotoren als Aktoren in der Miniautomatisierung oder im Modellbau üblich.

Hintergrund/Einführung in die Idee: Drei Servomotoren bewegen wesentliche Komponenten der Mini-Fertigungsanlage: Ein Servomotor bildet den Durchlass für die Eingangsrutsche, ein Servomotor ermöglicht die Ausgabe von Kugeln am Dispenser, und ein Servomotor bewegt den Auswerfer. Die Schüler:innen können so ihre bisher erworbenen Programmierkenntnisse zum Servomotor auf verschiedene Anwendungen übertragen.

Wenn die Lehrkraft dies einführen will, können die Schüler:innen lernen, dass verschiedene reale Objekte desselben Typs in der Programmierung verschiedene Objekte derselben Klasse sind – eine wichtige Erkenntnis der objektorientierten Programmierung.

Dauer: ca. 30 Minuten



Materialien für die Lehrkraft: PowerPoint-Präsentation

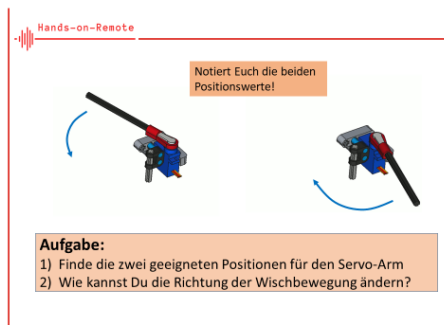
Materialien für die Schüler:innen: Link zur Wokwi-Simulation (<https://wokwi.com/projects/328915985532715602>), Multimediale HTML-Datei, Arduino, Arduino IDE, Servomotoren, Breadboard, Jumper-Kabel. Falls die Lehrkraft diese verwenden möchte: Portable Version der Arduino IDE auf einem USB-Stick.

Format: Online oder in einem Computerraum an der Schule

Inhalt:

Aktivitäten der Lehrkräfte

– Einführung: Was kann ein Servomotor? Wofür wollen wir ihn verwenden?



Aktivitäten der Schüler:innen

- Die Jugendlichen können wählen, ob sie zuerst mit dem Dispenser, der Schranke für den Durchlass oder dem Auswerfer arbeiten möchten. Der Servomotor wechselt nur zwischen zwei Positionen; er schaltet zwischen Ruheposition und aktiver Position um.
- Die Schüler:innen können den erforderlichen neuen Code von der Simulation in die Arduino IDE kopieren, oder auch umgekehrt.
- Die Schüler:innen bauen eine Schaltung mit einem realen Servomotor und dem Arduino und versuchen, geeignete Ruhe- oder Aktivpositionen zu finden.

4.5 Einheit 3.a: Drehscheibe

Themen: Anwenden praktischer Fähigkeiten beim Zusammenbau mechanischer Teile, Erstellen eines weiteren einfachen Schaltkreises und Einfügen neuen Programmcodes, um ein neues Objekt zu erstellen.

Zielsetzung: Die Schüler:innen sammeln praktische Erfahrung beim Aufbau der Drehscheibe und beim Einpassen des Antriebsmotors. Möglicherweise müssen die Jugendlichen beim Zusammenbau kleine mechanische Probleme lösen, z. B. um die richtige Position für den Antriebsmotor zu finden. Sie bauen einen weiteren Schaltkreis mit dem Arduino und fügen neuen Code hinzu, um ein neues Element zu bewegen.

Hintergrund/Einführung in die Idee: Die Drehscheibe ist ein zentrales Element des miniaturisierten Automatisierungsprozesses, da sie den Transport von Objekten zwischen den verschiedenen Stationen des Automatisierungsprozesses ermöglicht.

Format: Die Schüler:innen arbeiten selbstständig mit der Anleitung durch die multimediale HTML-Datei, entweder online und mit Unterstützung der Lehrkraft per Videokonferenz oder in einem Computerraum an der Schule.

Dauer: ca. 20 min



Materialien für die Lehrkraft: PowerPoint-Präsentation, einschließlich Links zu YouTube-Videos

Materialien für die Schüler:innen: Multimediale HTML-Datei, Arduino IDE, Grundplatte, Drehscheibe, Getriebemotor mit Reibrad, Motorhalterung, Gummiring, Schraube mit zwei Muttern als Achse, schwarze Lego-Pins, Arduino, Steckbrett, Jumper-Kabel

Inhalt:

Aktivitäten der Lehrkräfte

- Kurze Einführung in den Aufbau der Drehscheibe anhand der PowerPoint-Präsentation



Aktivitäten der Schüler:innen

- Die Jugendlichen montieren die Drehscheibe und den Antriebsmotor auf die Grundplatte.
- Die Schüler:innen erweitern die Schaltung des Arduino und schließen den Antriebsmotor an.

```
//Create a motor object
Motor xxx(6);
...
xxx.go(HIGH);
...
xxx.go(LOW);
```

- Die Schüler:innen fügen neue Codezeilen für den Antriebsmotor hinzu. Sie erstellen ein Motorobjekt und schalten den Motor ein oder aus. Der Code, der zum Ein- und Ausschalten des Motors benötigt wird, ist dem Code, der zum Blinken einer LED benötigt wird, sehr ähnlich, so dass die Schüler ihr zuvor erworbenes Wissen übertragen können. Ihre Aufgabe ist es, die Drehscheibe um eine Vierteldrehung zu bewegen.

4.6 Einheit 3.b: Lichtschranke

Themen: Schaltungsaufbau für die Reflexlichtschranke, lichtabhängiger Widerstand (LDR) als Helligkeitssensor, Zusammenhang zwischen Widerstandswerten und Lichtintensität, sensorabhängige Steuerung von Prozessen

Zielsetzung: Die Schüler:innen erkennen, dass für die automatisierte Steuerung einer Produktionsanlage Sensoren notwendig sind. Sie lernen die Funktionsweise einer Reflexlichtschranke kennen und verwenden den darin eingebauten Lichtsensor zur Steuerung ihres "Produktionsprozesses".

Dauer: ca. 25 min



Materialien für die Lehrkraft: PowerPoint-Präsentation, einschließlich Link zu YouTube-Video

Materialien für die Schüler:innen: Multimediale HTML-Datei, Arduino IDE, bereits zusammengebaute Drehscheibe, LDR + LED mit Halterung als Reflexlichtschranke, bereits zusammengesteckte Schaltung auf Arduino und Steckbrett, weitere Jumper-Kabel

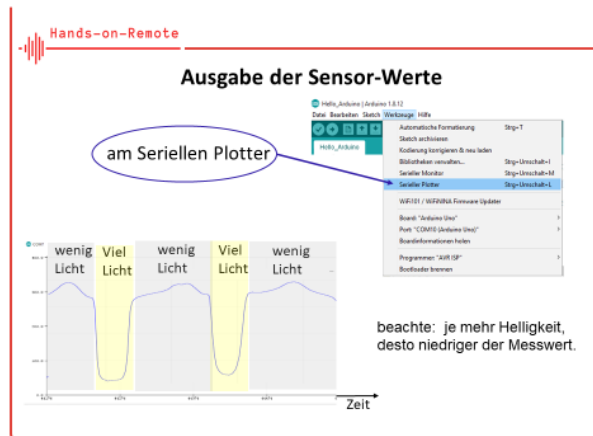
Hintergrund/Einführung in die Idee: Lichtschranken sind häufig verwendete Elemente im Alltag, z. B. beim Schließen von S-Bahn-Türen. Mit der Mini-Produktionsanlage lernen die Jugendlichen eine mögliche Anwendung für Automatisierungsprozesse kennen. Sie erfahren, wie der lichtabhängige Widerstand funktioniert und wie Messwerte schon durch kleine Helligkeitsabweichungen beeinflusst werden können. Die Lehrkraft kann diese Messungen nutzen, um den Schüler:innen die Schwierigkeiten einer genauen und aussagekräftigen Messung vor Augen zu führen und mit den Teams am Beispiel ihrer Mini-Fertigungsanlage zu diskutieren, was dies für sensorabhängige Prozesse bedeutet.

Format: Die Schüler:innen arbeiten selbstständig mit der Anleitung durch die multimediale HTML-Datei, entweder online und mit Unterstützung der Lehrkraft per Videokonferenz oder in einem Computerraum in der Schule. In einer Online-Umgebung können die Schüler:innen in Break-Out-Rooms zusammenarbeiten.

Inhalt:

Aktivitäten der Lehrkräfte

- Die Lehrkraft zeigt einen typischen Einsatz von Reflex-Lichtschränken in Automatisierungsprozessen.
- Die Lehrkraft stellt den lichtabhängigen Widerstand (LDR) als Helligkeitssensor vor.



- Die Lehrkraft unterstützt die Schüler:innen bei der Verwendung der Lichtschranke und der Interpretation der Messwertkurven.
- Die Lehrkräfte stehen für Fragen der Schüler:innen zur Verfügung, z. B. per Videokonferenz.



Aktivitäten der Schüler:innen

- Die Jugendlichen lernen anhand der multimedialen Anleitung die Eigenschaften einer Reflexlichtschranke kennen.
- Die Schüler:innen montieren die Reflexlichtschranke auf die Grundplatte.
- Die Jugendlichen erweitern die Arduino-Schaltung und schließen den LDR/die Reflexlichtschranke an den Arduino an.
- Die Schüler:innen lassen sich die vom LDR erhaltenen Sensorwerte mit dem Arduino-Plotter anzeigen und interpretieren die Grafiken. So können sie Veränderungen der Werte bei verschiedenen Lichtverhältnissen nachvollziehen.
- Die Schüler:innen verwenden die Reflexlichtschranke, um zu erkennen, ob ein Becher vor dem Lichtsensor steht. Je nachdem, ob dies der Fall ist, können sie die Drehscheibe anhalten oder starten.
- Die Schüler:innen fügen neue Codezeilen für die Lichtschranke hinzu, insbesondere den Wartebefehl "waitForCan" - nur wenn der Sensor einen Becher erkennt, wird das Programm fortgesetzt.

4.7 Einheit 4: Erweiterungen des Automatisierungsprozesses

Themen: Transfer des zuvor erworbenen Wissens auf neue Elemente

Zielsetzung: Die Schüler:innen arbeiten selbstständig und können wählen, welche Komponenten für die Drehscheibe sie in ihren Automatisierungsprozess einbauen. Dabei können sie bereits erworbene Kenntnisse über Programmierung, Schaltungen oder Automatisierungsprozesse übertragen und auf ihre neuen Aufgaben anwenden. Sie erhalten so außerdem einen oft gewünschten Freiraum für eigene Experimente.

Dauer: flexibel, mind. 45 Minuten



Materialien für die Lehrkraft: Falls erforderlich: Datei, in der häufige Fehler und Schwierigkeiten aufgeführt sind.

Materialien für die Schüler:innen: Multimediale HTML-Datei, Arduino IDE, bereits zusammengebaute Drehscheibe, LDR + LED mit Halterung als Reflexlichtschranke, Arduino, Steckbrett, Jumper-Kabel, als Material zur Auswahl für die Schüler:innen: Dispenser, Auswerfer, Eingangsrutsche, LEDs oder Ampelmodul

Hintergrund/Einführung in die Idee: Die Schüler:innen können ihren Automatisierungsprozess mit mehreren Funktionselementen aufbauen.



Format: Die Schüler:innen arbeiten selbstständig mit der Anleitung durch die multimediale HTML-Datei, entweder online und mit Unterstützung der Lehrkraft per Videokonferenz oder in einem Computerraum an der Schule. In einer Online-Umgebung können die Schüler:innen in Breakout-Rooms zusammenarbeiten.

Inhalt

Aktivitäten der Lehrkräfte

Die Lehrkräfte stehen für Fragen der Schüler:innen zur Verfügung, z. B. über Videokonferenzen.



Aktivitäten der Schüler:innen

Die Jugendlichen arbeiten mit der multimedialen HTML-Datei, um weitere Elemente zu ihrem Automatisierungsprozess hinzuzufügen. Von der Übersichtsseite der multimedialen HTML-Datei aus können die Schüler:innen durch Anklicken zu den verschiedenen Komponenten gelangen, die sie interessieren. Die Jugendlichen können selbst wählen, welche Elemente sie dem Automatisierungsprozess hinzufügen möchten.

Kurzversion	Variation	Variation	Variation	Lange Version
Drehscheibe Lichtschranke Dispenser	Drehscheibe Lichtschranke Dispenser LED-Statusanzeige für den Dispenser	Drehscheibe Lichtschranke Dispenser Auswerfer	Drehscheibe Lichtschranke Dispenser Auswerfer Eingangsrutsche	Drehscheibe Lichtschranke Dispenser Auswerfer Eingangsrutsche LED-Statusanzeige für die Akteure

Diese Einheit kann flexibel in verschiedenen Varianten (s. Tabelle oben) durchgeführt werden, je nach zeitlichen und inhaltlichen Anforderungen der Lehrkraft. Mit den verschiedenen Varianten können die Lehrkräfte auch unterschiedliche Programmierkenntnisse und Vorerfahrungen der Schüler:innen berücksichtigen.



Themen: Erweiterung der Programmierkenntnisse, Teamwork und Kollaboration

Zielsetzung: Kollaboratives Arbeiten der Schüler:innen

Dauer: mind. 45 min



Materialien für die Lehrkraft: Keine

Materialien für die Schüler:innen: In den Teams bereits aufgebaute Miniproduktionsanlagen

Notwendiges Vorwissen: Die Schüler-Teams müssen mindestens eines der Funktionsteile des Automatisierungsprozesses programmiert haben. Idealerweise haben Schüler:innen in verschiedenen Teams verschiedene Funktionsteile programmiert.

Format: In einem Computerlabor an der Schule

Inhalt

Aktivitäten der Lehrkräfte

Die Lehrkraft initiiert und unterstützt die Kollaboration der Schüler:innen und ihr Teamwork.



Aktivitäten der Schüler:innen

Die Aufgabe der Jugendlichen ist es, ihre Drehscheiben teamübergreifend zu koppeln. Die Schüler:innen entscheiden, wie sie für die Kopplung der Drehscheiben vorgehen wollen und programmieren zusammen die notwendigen Programmierschritte.



Themen: Messungen mit Sensoren, sensorabhängiges Auslösen verschiedener Aktoren des Automatisierungsprozesses, auch an entfernten Standorten

Zielsetzung: Ein Gemeinschaftsgefühl zwischen den Schüler:innen fördern

Dauer: Flexibel



Materialien für die Lehrkraft: Keine

Materialien für die Schüler:innen: Lichtreflexschranke, LED

Notwendiges Vorwissen: Die Schüler:innen sollten vorab einen sensorabhängigen Prozess, einschließlich der sensorabhängigen Auslösung eines Aktors, programmiert haben (Einheit 3.b).

Format: Online oder in einem Computerlabor an der Schule

Inhalt:

Aktivitäten der Lehrkräfte

Die Lehrkräfte machen die Schüler:innen mit der Idee eines Kommunikationsprozesses per Lichtübertragung vertraut und unterstützen die Jugendlichen bei der Ausrichtung ihrer Sensoren.



Aktivitäten der Schüler:innen

Die Jugendlichen programmieren einen Aktor des Automatisierungsprozesses so, dass er sich bewegt, wenn der Lichtsensor einen Anstieg der Helligkeit erfasst. Die Team-Mitglieder befinden sich zusammen in einer Videokonferenz. Der erste Schüler positioniert eine LED vor der Webcam für die Videokonferenz. Der zweite Schüler positioniert die Reflexlichtschranke vor seiner eigenen Videokonferenzansicht so, dass der Lichtsensor das Lichtsignal der LED des ersten Schülers erkennt. Wenn der erste Schüler das Licht der LED einschaltet, erkennt der Lichtsensor beim zweiten Schüler einen Anstieg der Helligkeit und setzt dort den Aktor in Bewegung. Der Teampartner kann das Lichtsignal nun seinerseits an einen dritten Schüler weitergeben usw.

4.10 Einheit 7: Automatisierung im sozialen Kontext

Die Lehrkräfte können entweder in dieser Reihenfolge fortfahren oder zunächst andere Anwendungen der Drehscheibe (Einheit 8) einführen.

Themen: Konsumentenverhalten, Abhängigkeit von produzierten Gütern, Nebeneffekte des Konsums (Energiebedarf, Abfall, Transport, etc.), Vor- und Nachteile von Automatisierungsprozessen

Zielsetzung: Die Schüler:innen erkunden, welche unterschiedlichen Auswirkungen Automatisierungsprozesse in der Gesellschaft haben können. Die Jugendlichen reflektieren ihr eigenes Konsumverhalten.

Dauer: flexibel, mind. 45 min



Materialien für die Lehrkraft: Zugang zum Online-Whiteboard – der Zugang wird von der Lehrkraft an die Schüler:innen verteilt. Fragekarten, welche die Lehrkraft an die Schüler:innen austeilen kann.

Materialien für die Schüler:innen: Fragekarten, Online-Whiteboard und Internetverbindung, falls die Lehrkraft diese nutzen möchte: bereits aufgebaute Drehscheiben

Format: Online oder in einem Computerraum an der Schule

Inhalt:

Aktivitäten der Lehrkräfte

Die Lehrkraft teilt die Fragekarten an die Schüler:innen aus und/oder leitet eine Gruppendiskussion.

Die Lehrkraft gestaltet die Einführung und Überleitung zum neuen Thema mit einer Gratulation für die Leistung der Schüler:innen und der Frage, wie sie sich nach Beendigung ihrer Aufgaben gerne verwöhnen würden. Wenn die Lehrkraft eine Antwort direkt umsetzen kann, kann sie diese gerne in ihren Unterricht einbauen.



Herzlichen Glückwunsch zu Eurer ersten automatisierten Fertigung!

Jetzt habt Ihr richtig was geschafft, jetzt könnt Ihr Euch etwas Schönes gönnen...

Was würdet Ihr Euch denn im Augenblick gern gönnen?

Diskutiert gern im Team, was Euch so einfällt und wenn Ihr mögt, notiert es hier...



Gibt es bei Euren Ideen, was Ihr Euch jetzt gern gönnen würdet, etwas, das ohne Konsum und ohne Produkte auskommt?

Könnt Ihr Euch vorstellen, Euch etwas zu gönnen, ohne dafür etwas zu brauchen oder verbrauchen? Was wäre das?

Die Lehrkraft bespricht die Antworten mit den Jugendlichen und unterscheidet, ob der Konsum von Produkten zu den vorgeschlagenen Belohnungen gehört oder nicht. Wenn alle Antworten auf Konsum basieren, teilt die Lehrkraft die zweite Karte aus, auf der gefragt wird, ob auch eine Belohnung ohne Konsum möglich wäre. Die Lehrkraft diskutiert mit den Schüler:innen über die Nebenwirkungen eines hohen Konsums (Energiekosten, Abfall, Transport etc.).

Je nach Diskussionsverlauf und gewähltem Schwerpunkt stellt die Lehrkraft weitere Fragen. Die Lehrkraft macht die Schüler:innen mit dem verwendeten Online-Whiteboard vertraut und erklärt ihnen ihre Aufgabe. Die Lehrkraft kann die Jugendlichen ein Thema wählen lassen, das sie vertiefen möchten. Wenn die Lehrkraft dies ausprobieren möchte, kann sie eine Papierunterlage mit Themenvorschlägen für die Drehscheibe verwenden und die Schüler:innen auf diese Weise ihr Thema wählen lassen.





Aktivitäten der Schüler:innen

- Die Jugendlichen diskutieren, was sie sich gern gönnen würden und reflektieren zusammen mit der Lehrkraft ihr Konsumverhalten.
- Die Schüler:innen können beim Betrachten ihrer selbst gebauten Mini-Produktionsanlage feststellen, dass diese auch selbst etwas „Produziertes“ ist – welche Teile stammen bei der Anlage von einem Automatisierungsprozess, welche sind selbst hergestellt?
- Auf dieser Grundlage können die Schüler:innen diskutieren, welche Vor- und Nachteile Automatisierungsprozesse haben. Sie können auch überlegen, mit welchen Vor- und Nachteilen selbst hergestellte Materialien einhergehen. Die Jugendlichen können ihre spontanen Ideen und Gedanken, die sie mit dem Thema Automatisierung und deren gesellschaftlichen Konsequenzen verbinden, auf einem Online-Whiteboard sammeln. Sie können das Whiteboard um weitere Materialien und Verknüpfungen ergänzen.
- Die Jugendlichen können sich nun ein bestimmtes Thema aussuchen, das sie vertiefen möchten, und es in Teams diskutieren.

4.11 Einheit 8: Andere Anwendungen der Drehscheibe

Themen: Helligkeitswerte verwenden, um Objekte zu unterscheiden oder Töne abzuspielen.

Zielsetzung: Sowohl die Jugendlichen als auch die Lehrkräfte brachten die Idee einer anderen Anwendung für die Materialien ein, um mit dem gleichen Materialkit weiterarbeiten und etwas Neues ausprobieren zu können. Einige von ihnen waren sehr motiviert, sich in ihrer Freizeit weiter mit dem Materialkit zu beschäftigen. Die Schüler:innen können mit den zusätzlichen Anwendungen selbstständig weiterarbeiten.

Gleichzeitig können diese anderen Anwendungen Alternativen darstellen, wenn Lehrkräfte und Schüler:innen nicht zu tief in die Programmierung einsteigen wollen.

Dauer: flexibel, mind. 45 min



Materialien für die Lehrkraft: Falls erforderlich: Drucker zum Ausdrucken der von den Schülern erstellten Papiervorlagen mit verschiedenen Graustufen

Materialien für die Schüler:innen: Arduino, Steckbrett, Jumper-Kabel, Arduino IDE, bereits montierte Drehscheibe, Lichtschranke

Zusätzlich für die Sortiermaschine: Auswerfer und LED-Statusanzeige

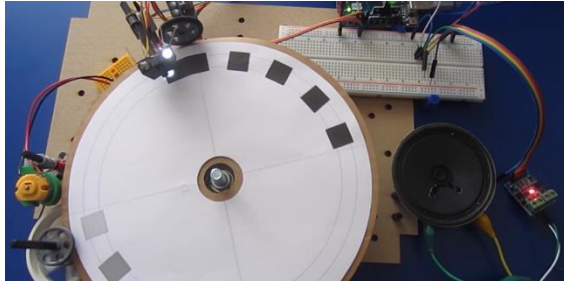
Zusätzlich für den Lichtplattenspieler: Papier auf Karton in Scheibengröße, um das Helligkeitsmuster fest auf den Drehteller zu legen, einen Drucker, um das Schwarz-Weiß-Muster oder die Graustufen auf die Papierscheibe zu drucken, einen Summer oder einen Lautsprecher

Sortiermaschine	Lichtplattenspieler
Drehscheibe Lichtschranke LED-Statusindikator Helligkeit/Farbe Auswerfer zum Sortieren	Drehscheibe mit Papierauflagen mit Schwarz-Weiß-Mustern oder Graustufen Lichtsensoren als Messkopf Lautsprecher (nicht im Materialkit enthalten)

Format: Selbstständige Arbeit der Schüler:innen

Inhalt:

- Sortiermaschine: Mit dem Auswerfer verschiedene Arten von Objekten sortieren, die sich z. B. in Helligkeit oder Farbe unterscheiden.



- Lichtplattenspieler: Legt man ein Papier mit einem Schwarz-Weiß-Muster auf die Drehscheibe, lassen sich mit Hilfe des Lichtsensors die Helligkeitsinformationen in Töne und einfache Rhythmen für ein simples Tonausgabegerät umwandeln. Für diesen Zweck lassen sich z. B. auch gedruckte Graustufenskalen verwenden.



Aktivitäten der Schüler:innen

- Die Jugendlichen programmieren die Mini-Produktionsanlage als Sortiermaschine und unterscheiden Becher nach Farbe oder Helligkeit.
- Die Schüler:innen entwerfen Graustufen-Muster für die Drehscheibe. Der Arduino erkennt die Helligkeitswerte und spielt die dazugehörigen Töne ab. Die Jugendlichen können in der Programmierung auch selbst Töne zu Helligkeitswerten zuordnen.

4.12 Allgemeine Hinweise zu Videokonferenzen und praktischer Arbeit in Distanzsituationen

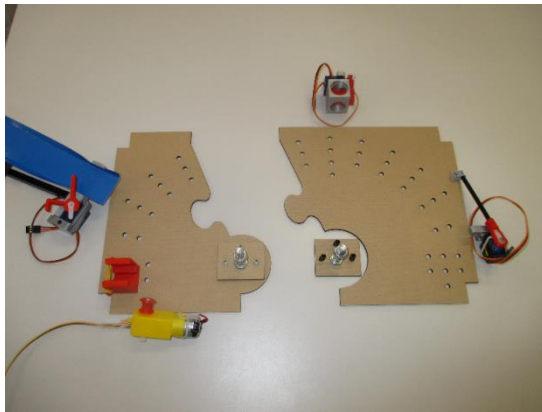
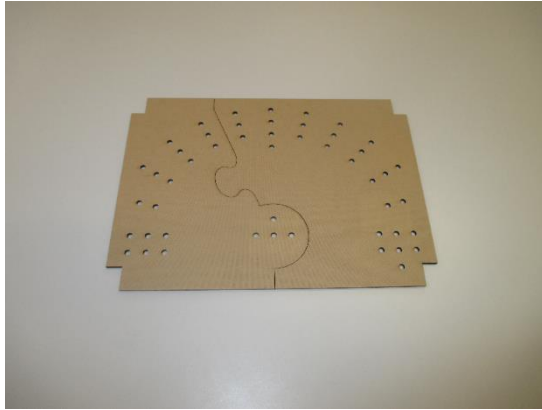
Datenschutz

Die Schule sollte Fragen zur notwendigen Datenverarbeitung bei Videokonferenzen – z. B. die Nutzung der Bildschirmfreigabe oder der Einsatz der Kamera – im Einklang mit der Datenschutz-Grundverordnung klären. In enger Absprache mit den Schüler:innen können die Lehrkräfte den Unterricht ohne eingeschaltete Kamera durchführen. Wenn die Schüler:innen ihre Kamera einschalten, ist es jedoch viel einfacher Fortschritte zu sehen und es kann das Gruppengefühl erheblich verbessern. Eine Möglichkeit, welche die Lehrkräfte anbieten können, besteht darin, die Kamera nur auf die Experimente richten zu lassen, ohne dass die Jugendlichen selbst im Bild sein müssen – diese Option steht aufgrund von Fragen der Kameratechnik und der Platzierung der Experimente jedoch nicht allen zur Verfügung.

Verwaltung der Materialien

Der erste Teil der Projektsequenz, d. h. die Abschnitte 1.a Einführung in das Thema bis 2.b Servomotor, kann problemlos in einer Online-Lernumgebung durchgeführt werden. Hierfür benötigen die Lehrkräfte ein geeignetes Videokonferenzsystem, welches ihnen ermöglicht, eine Online-Präsentation zu zeigen und ggf. Breakout Rooms einzurichten, in denen die Schüler:innen selbstständig in Teams arbeiten können. Der Anteil der Hands-on-Materialien, welche die Jugendlichen in diesem Part für ihre Arbeit zu Hause benötigen, besteht aus einem Arduino, einem Steckbrett, Jumper-Kabeln, einem Ampelmodul oder LEDs und einem Servo – also einer insgesamt relativ überschaubaren Ausstattung. In einer vollständigen Distanzsituation erfordert dies jedoch, dass alle Jugendlichen in einer Klasse ein solches Set erhalten. Ist dies nicht möglich, kann die Lehrkraft stattdessen die Arbeit mit der Online-Simulationsumgebung Wokwi vertiefen. Dies kann jedoch zu Lasten des Erlernens der für spätere Einheiten notwendigen Kompetenzen im Umgang mit einem Mikrocontroller gehen.

Was die Drehscheibe betrifft, ist die Umsetzung etwas komplizierter. Idealerweise erhalten alle Schüler:innen einen kompletten Materialsatz, mit dem sie zu Hause arbeiten können. Wenn dies nicht möglich ist (sei es aufgrund der Kosten, des Aufwands für die Materialproduktion oder der Schwierigkeit, die Materialien auszuleihen oder zu versenden), ist eine Alternative nötig.



Eine Alternative kann z. B. sein, die Grundplatte für die Drehscheibe zu teilen und den Schüler:innen verschiedene Teile zu geben, damit sie als Team zusammenarbeiten und so den Automatisierungsprozess gemeinsam gestalten können.

5 Möglichkeiten modularer Nutzung

Je nach gegebenen Voraussetzungen, kann es hilfreich sein, das Unterrichtsmodul auch in anderer Form nutzen zu können. Im Folgenden werden zwei alternative Abläufe aufgezeigt; es gibt darüber hinaus viele verschiedene andere Möglichkeiten.

Ablauf 1

Im Zentrum steht bei Ablauf 1 weiterhin der Automatisierungsprozess. Allerdings reduziert sich der Prozess auf den Transport der Becher mit der Drehscheibe, den Einsatz des Servomotors für den Dispenser und eine sensorabhängige Steuerung des Produktionsprozesses.

1.1 Einführung in die Verwendung von Drehscheiben in der Automatisierung und in das Ziel der Miniaturisierung eines Automatisierungsprozesses.
1.2 Einführung in die Arbeit mit dem Arduino anhand eines ersten Schaltungsentwurfs und Programmierbeispiels mit verschiedenfarbigen LEDs.
2.1 Einführung zu Servomotoren, einschließlich Schaltung und Programmierung eines realen Servomotors. Verwendung des Servomotors für den Dispenser.
2.2 Aufbau der Drehscheibe, Schaltung und Programmierung des Antriebsmotors.
2.3 Einsatz der Lichtschranke, einschließlich der Schaltung, Programmierung und Anzeige der Messwerte mit dem Plotter. Verwendung des Lichtsensors zur sensorabhängigen Steuerung des „Produktionsprozesses“.
3. Automatisierung im gesellschaftlichen Kontext

Ablauf 2

Mit dem Ablauf 2 wird die Drehscheibe ganz anders genutzt, nämlich ausschließlich als Lichtplattenspieler. Es sind daher nur die grundlegenden Schritte notwendig, um mit dem Arduino zu arbeiten, die Drehscheibe in Bewegung zu bringen und die Lichtschranke einzusetzen.

1.1 Einführung in ein selbstgewähltes Thema
1.2 Einführung in die Arbeit mit dem Arduino anhand eines ersten Schaltungsentwurfs und Programmierbeispiels mit verschiedenfarbigen LEDs
2.1 Aufbau der Drehscheibe, Schaltung und Programmierung des Antriebsmotors.
2.2 Einsatz der Lichtschranke, einschließlich der Schaltung, Programmierung und Anzeige der Messwerte mit dem Plotter.
3. Einsatz der Drehscheibe als Lichtplattenspieler

6 Anhang: Einblicke in die interaktive multimediale Anleitungsseite

Die Screenshots zeigen für einen ersten und schnellen Eindruck beispielhaft die Schritt-für-Schritt-Anleitung mit Bildern der Animationen, Videos und Mouseover-Effekte für Aufbau, Verkabelung und Programmierung der Eingangsruutsche und Lichtreflexschranke. Um einen besseren Eindruck zu bekommen, empfiehlt es sich ggf., auf eine größere Ansicht zu zoomen.



Hands-on-Remote

- Vorab
- Struktur
- Drehscheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang
- Übersicht

Hardware - Eingangsruutsche montieren

Eingangsruutsche montieren:

1. Befestige die Eingangsruutsche mit dem Standfuß in der Grundplatte.
2. Schiebe die Ruutsche in die Führung.
3. Der Lego-Arm (Lochbalken) dient als Stütze für die Ruutsche.

Hands-on-Remote

- Vorab
- Struktur
- Drehscheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang
- Übersicht

Verkabelung - Eingangsruutsche

Bitte vor jedem Ändern der Verkabelung unbedingt den Arduino vom Computer trennen!

Arbeiten am Arduino nur in stromlosem Zustand!

Nun muss der Servo der Eingangsruutsche mit dem Arduino verbunden werden:

Nutze die Versorgungsleisten am Steckbrett (rot und blau):	am Arduino:	am Steckbrett:
rotes Kabel:	V in - Pin	rote Schiene
blaues Kabel:	GND - Pin	blaue Schiene

Schließe den Servo der Eingangsruutsche (=Gate) an:

Orange	-> Pin 9
Rot	-> V in - Schiene
Braun	-> GND - Schiene

Fahre mit der Maus über das Bild...

Hands-on-Remote

- Vorab
- Struktur
- Drehscheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang
- Übersicht

Verkabelung - Eingangsruutsche

Bitte vor jedem Ändern der Verkabelung unbedingt den Arduino vom Computer trennen!

Arbeiten am Arduino nur in stromlosem Zustand!

Nun muss der Servo der Eingangsruutsche mit dem Arduino verbunden werden:

Nutze die Versorgungsleisten am Steckbrett (rot und blau):	am Arduino:	am Steckbrett:
rotes Kabel:	V in - Pin	rote Schiene
blaues Kabel:	GND - Pin	blaue Schiene

Schließe den Servo der Eingangsruutsche (=Gate) an:

Orange	-> Pin 9
Rot	-> V in - Schiene
Braun	-> GND - Schiene

Hands-on-Remote

- Vorab
- Struktur
- Drehscheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang
- Übersicht

Hardware - Gate-Servo

Die Schranke der Eingangsruutsche (= das Gate) wird von einem Servo bewegt.

Servos sind kleine Getriebemotoren, die ...

Fahre mit der Maus über die Bilder...

Für die Schranke brauchen wir zwei Positionen:

- > geschlossene Schranke (= Ruheposition: hält alle Becher zurück)
- > offene Schranke (= Durchlassposition: ein Becher rutscht auf die Scheibe)

Der gewinkelte Zeiger...

Zum Starten der Animation ins Bild klicken...

Die Gradzahlen der beiden Positionen müssen durch Ausprobieren ermittelt werden.

Übrigens: Die Ruheposition hat einen größeren Winkelwert als die Durchlassposition.

Hands-on-Remote

- Vorab
- Struktur
- Drehscheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang
- Übersicht

Hardware - Gate-Servo

Die Schranke der Eingangsruutsche (= das Gate) wird von einem Servo bewegt.

Servos sind kleine Getriebemotoren, die sich meist nur eine halbe Umdrehung, zwischen 0° und 180° bewegen können. Man schickt dem Servo die gewünschte Gradzahl und der Servo fährt die Position an.

Für die Schranke brauchen wir zwei Positionen:

- > geschlossene Schranke (= Ruheposition: hält alle Becher zurück)
- > offene Schranke (= Durchlassposition: ein Becher rutscht auf die Scheibe)

Der gewinkelte Zeiger...

Zum Starten der Animation ins Bild klicken...

Die Gradzahlen der beiden Positionen müssen durch Ausprobieren ermittelt werden.

Übrigens: Die Ruheposition hat einen größeren Winkelwert als die Durchlassposition.

Hands-on-Remote

- Vorab
- Struktur
- Drehscheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang
- Übersicht

Hardware - Gate-Servo

Die Schranke der Eingangsruutsche (= das Gate) wird von einem Servo bewegt.

Servos sind kleine Getriebemotoren, die ...

Fahre mit der Maus über die Bilder...

Für die Schranke brauchen wir zwei Positionen:

- > geschlossene Schranke (= Ruheposition: hält alle Becher zurück)
- > offene Schranke (= Durchlassposition: ein Becher rutscht auf die Scheibe)

Der gewinkelte Zeiger der Schranke verhindert, dass beim Öffnen der Schranke weitere Becher durchrutschen.

Zum Starten der Animation ins Bild klicken...

Die Gradzahlen der beiden Positionen müssen durch Ausprobieren ermittelt werden.

Übrigens: Die Ruheposition hat einen größeren Winkelwert als die Durchlassposition.

Hands-on-Remote

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

Programmieren: Gate steuern

Arbeite im ino-Tab.
Aufgabe: Programmiere den Gate-Servo so, dass er einen Becher ausgibt. Finde dazu die passenden Winkelwerte für die Ruhe- und Durchlass-Position.

1. Binde mit der #include-Anweisung die „h-Datei“ ein...
2. Erzeuge ein Servo-Objekt, benenne es und ...
3. Im setup() ...
4. Drehe in der loop() die Schranke des Gates ...
5. Kommentiere Dein Programm ...

```

#include <Servo.h>

ServoMotor name(pin_pos0_pos1);

void setup() {
  name.attach(pin_pos0_pos1);
  name.write(0);
}

void loop() {
  name.write(180);
}

```

Erweiterungs-Aufgabe:
Eine LED soll die Aktivität des Gates (Gesamtvorgang Drehen auf Durchlass-Position und wieder zurück in Ruhe-Position) anzeigen.

Hands-on-Remote

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

Programmieren: Gate steuern

Arbeite im ino-Tab.
Aufgabe: Programmiere den Gate-Servo so, dass er einen Becher ausgibt. Finde dazu die passenden Winkelwerte für die Ruhe- und Durchlass-Position.

1. Binde mit der #include-Anweisung die „h-Datei“ ein...
2. Erzeuge ein Servo-Objekt, benenne es und ...
3. Im setup() ...
4. Drehe in der loop() die Schranke des Gates ...
5. Kommentiere Dein Programm ...

```

#include <Servo.h>

ServoMotor name(pin_pos0_pos1);

void setup() {
  name.attach(pin_pos0_pos1);
  name.write(0);
}

void loop() {
  name.write(180);
}

```

Erweiterungs-Aufgabe:
Eine LED soll die Aktivität des Gates (Gesamtvorgang Drehen auf Durchlass-Position und wieder zurück in Ruhe-Position) anzeigen.

Hands-on-Remote

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

Programmieren: Gate steuern

Arbeite im ino-Tab.
Aufgabe: Programmiere den Gate-Servo so, dass er einen Becher ausgibt. Finde dazu die passenden Winkelwerte für die Ruhe- und Durchlass-Position.

1. Binde mit der #include-Anweisung die „h-Datei“ ein...
2. Erzeuge ein Servo-Objekt, benenne es und übergebe in den Klammern () die Pinnummern, an der Du den Servo angesteckt hast und die beiden Winkelpositionen.
3. Im setup() ...
4. Drehe in der loop() die Schranke des Gates ...
5. Kommentiere Dein Programm ...

```

#include <Servo.h>

ServoMotor gate(9, 90, 270);

void setup() {
  gate.attach(9);
  gate.write(90);
}

void loop() {
  gate.write(270);
  delay(1000);
  gate.write(90);
  delay(1000);
}

```

Erweiterungs-Aufgabe:
Eine LED soll die Aktivität des Gates (Gesamtvorgang Drehen auf Durchlass-Position und wieder zurück in Ruhe-Position) anzeigen.

Hands-on-Remote

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

Programmieren: Gate steuern

Arbeite im ino-Tab.
Aufgabe: Programmiere den Gate-Servo so, dass er einen Becher ausgibt. Finde dazu die passenden Winkelwerte für die Ruhe- und Durchlass-Position.

1. Binde mit der #include-Anweisung die „h-Datei“ ein...
2. Erzeuge ein Servo-Objekt, benenne es und ...
3. Im setup() kannst Du den Servo mit der Methode go() in seine Ruheposition bringen.
4. Drehe in der loop() die Schranke des Gates ...
5. Kommentiere Dein Programm ...

```

#include <Servo.h>

ServoMotor gate(9, 90, 270);

void setup() {
  gate.attach(9);
  gate.go(90);
}

void loop() {
  gate.go(270);
  delay(1000);
  gate.go(90);
  delay(1000);
}

```

Erweiterungs-Aufgabe:
Eine LED soll die Aktivität des Gates (Gesamtvorgang Drehen auf Durchlass-Position und wieder zurück in Ruhe-Position) anzeigen.

Hands-on-Remote

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

Programmieren: Gate steuern

Arbeite im ino-Tab.
Aufgabe: Programmiere den Gate-Servo so, dass er einen Becher ausgibt. Finde dazu die passenden Winkelwerte für die Ruhe- und Durchlass-Position.

1. Binde mit der #include-Anweisung die „h-Datei“ ein...
2. Erzeuge ein Servo-Objekt, benenne es und ...
3. Im setup() ...
4. Drehe in der loop() die Schranke des Gates mit der Methode goSlow() in die Durchlassposition und wieder zurück, indem Du die Werte HIGH bzw. LOW überbist.
5. Kommentiere Dein Programm ...

```

#include <Servo.h>

ServoMotor gate(9, 90, 270);

void setup() {
  gate.attach(9);
  gate.go(90);
}

void loop() {
  gate.goSlow(HIGH);
  delay(1000);
  gate.goSlow(LOW);
  delay(2000);
}

```

Erweiterungs-Aufgabe:
Eine LED soll die Aktivität des Gates (Gesamtvorgang Drehen auf Durchlass-Position und wieder zurück in Ruhe-Position) anzeigen.

Hands-on-Remote

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

Programmieren: Gate steuern

Arbeite im ino-Tab.
Aufgabe: Programmiere den Gate-Servo so, dass er einen Becher ausgibt. Finde dazu die passenden Winkelwerte für die Ruhe- und Durchlass-Position.

1. Binde mit der #include-Anweisung die „h-Datei“ ein...
2. Erzeuge ein Servo-Objekt, benenne es und ...
3. Im setup() ...
4. Drehe in der loop() die Schranke des Gates ...
5. Kommentiere Dein Programm: Mehrzeilige Kommentare beginnen mit /* und enden mit */ // leitet einzeiligen Kommentar ein.

```

#include <Servo.h>

ServoMotor gate(9, 90, 270);

void setup() {
  gate.attach(9);
  gate.go(90);
}

void loop() {
  // Schranke in Aktion:
  gate.goSlow(HIGH);
  delay(1000);
  gate.goSlow(LOW);
  delay(2000);
}

```

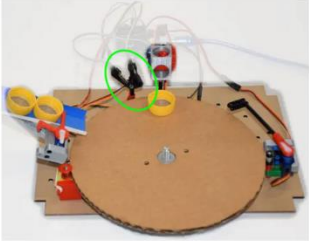
Erweiterungs-Aufgabe:
Eine LED soll die Aktivität des Gates (Gesamtvorgang Drehen auf Durchlass-Position und wieder zurück in Ruhe-Position) anzeigen.

Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschranke
- Auswerfer
- Eingang
- Übersicht

Übersicht

Automatisieren mit dem Arduino - Befüllstation einer Produktionsline



Bewege die Maus über die Kästchen

```

graph LR
    A[Becher vereinzeln] --> B[Becher transportieren]
    B --> C[Befüllen]
    C --> D[Transport]
    D --> E[Ausgabe]
    
```

Aufgabe: Klicke auf ein Symbol und springe ins zugehörige Kapitel.
Bist Du mit allem fertig? Dann klicke hier:


Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschranke
- Auswerfer
- Eingang
- Übersicht

Hardware - LED und LDR

Ein optischer Sensor, z.B. unser LDR, und eine Lichtquelle, z.B. unsere LED, arbeiten zusammen und werden als Lichtschranke bezeichnet. Im Gegensatz zu anderen Lichtschranken wird bei uns der Lichtstrahl nicht unterbrochen sondern reflektiert: Reflex-Lichtschranke.

LED und LDR zielen auf den selben Punkt.




Die Lichtschranke besteht aus zwei aktiven Komponenten, die ...

Fährt ein Becher in den Lichtkegel der LED so ...

Die LED beleuchtet ...
Der LDR misst ...

Beachte den LDR ↓




Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschranke
- Auswerfer
- Eingang
- Übersicht

Hardware - LED und LDR

Ein optischer Sensor, z.B. unser LDR, und eine Lichtquelle, z.B. unsere LED, arbeiten zusammen und werden als Lichtschranke bezeichnet. Im Gegensatz zu anderen Lichtschranken wird bei uns der Lichtstrahl nicht unterbrochen sondern reflektiert: Reflex-Lichtschranke.

LED und LDR zielen auf den selben Punkt.




Die Lichtschranke besteht aus zwei aktiven Komponenten, die wir beide ansteuern, bzw. auslesen müssen.

Fährt ein Becher in den Lichtkegel der LED so ...

Die LED beleuchtet den Raum vor dem LDR. Der LDR misst die Helligkeit vor sich und gibt ein analoges, der Helligkeit entsprechendes Signal aus.

Beachte den LDR ↓



Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschranke
- Auswerfer
- Eingang
- Übersicht

Hardware - LED und LDR

Ein optischer Sensor, z.B. unser LDR, und eine Lichtquelle, z.B. unsere LED, arbeiten zusammen und werden als Lichtschranke bezeichnet. Im Gegensatz zu anderen Lichtschranken wird bei uns der Lichtstrahl nicht unterbrochen sondern reflektiert: Reflex-Lichtschranke.

LED und LDR zielen auf den selben Punkt.



Die Lichtschranke besteht aus zwei aktiven Komponenten, die ...

Fährt ein Becher in den Lichtkegel der LED so reflektiert er ihr Licht und der LDR registriert die gestiegene Helligkeit.

Die LED beleuchtet ...
Der LDR misst ...

Beachte den LDR ↓



Hands-on-Remote


- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschranke
- Auswerfer
- Eingang
- Übersicht

Hardware - Lichtschranke

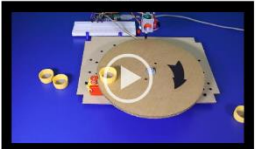
Nächster Schritt im mechanischen Aufbau: die Reflex-Lichtschranke.

Der Becher soll zuverlässig vor dem Dispenser stoppen. Dazu wird seine Position von einem optischen Sensor erkannt.

Fahre mit der Maus über das Bild...



Im Folgenden muss zunächst die Lichtschranke montiert werden, dann sowohl LDR (Lichtsensor) als auch LED (Lichtquelle) an den Arduino angeschlossen und schließlich der Arduino programmiert werden.




Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschranke
- Auswerfer
- Eingang
- Übersicht

Hardware - Lichtschranke montieren

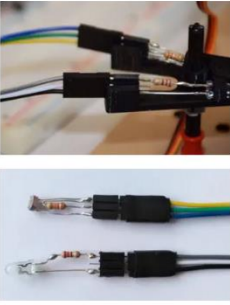
Reflex-Lichtschranke montieren: Mit einem Standfuß kann die Lichtschranken-Halterung in der Grundplatte befestigt werden.

Über den Achsstopper kann die Lichtschranke auf die passende Höhe gebracht werden.



LDR und LED sind bereits mit Widerstand verlötet ...

Beachte: das einzeln durchgehende Beinchen ...



Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang

Übersicht

Verkabelung - Lichtschränke



Bitte vor jedem Ändern der Verkabelung unbedingt den Arduino vom Computer trennen!

Arbeiten am Arduino nur in stromlosem Zustand!

Nun müssen LDR (Lichtsensoren) und LED (Lichtquelle) mit dem Arduino verbunden werden.

Nutze wieder die Versorgungsleisten (rot und blau) am Steckbrett:
am Arduino:
 rotes Kabel: V in - Pin → rote Schiene
 blaues Kabel: GND - Pin → blaue Schiene

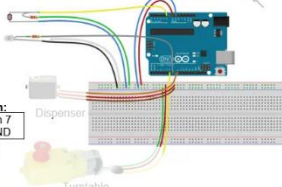
1. LDR anschließen:

Gelb → Pin A0
 Grün → V in - Schiene
 Blau → GND - Schiene

Fahre mit der Maus über das Bild...

2. LED anschließen:

Grau → Pin 7
 Schwarz → GND
 (das weiße Kabel wird bei der LED nicht benötigt)



Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang

Übersicht

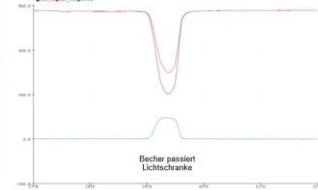
Messwerte - reflektiertes und Umgebungslicht



Geht der Lichtstrahl der LED ins Leere (kein Becher vor der Lichtschränke), so wird ihr Licht nicht reflektiert, der Lichtsensor sieht nur das Umgebungslicht.

Steht ein Becher im Lichtstrahl, addiert sich das vom Becher reflektierte Licht der LED zum Umgebungslicht, der Lichtsensor misst mehr Helligkeit.

Achtung: Je mehr Helligkeit, desto niedriger der Messwert. Dies liegt an der Art der Schaltung von Vorwiderstand und Lichtsensor.



Blaue Linie: Messwerte mit eingeschalteter LED
 Rote Linie: Messwerte mit ausgeschalteter LED
 Grüne Linie: Differenz der beiden Messwerte

Nur wenn ein Becher vor der Lichtschränke vorbeifährt und Licht reflektiert, unterscheiden sich die beiden Werte deutlich. Der Unterschied ist hier grün dargestellt. Diesen Unterschied werden wir aus.

Übersteigt der Unterschied den von uns gesetzten Wert, gilt ein Becher als erkannt.

Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang

Übersicht

Programmierung: Die Klasse „ReflexSensor“



ReflexSensor

- name: String
- pinLEDnumber: int
- pinLDRnumber: int
- trigger: int
- ReflexSensor(pinLED, pinLDR, trigger)
- ReflexSensor(pinLDR)
- showLDR(void): void
- showValues(void): void
- waitForCan(void): void
- detectCan(void): bool



Fahre mit der Maus über die Textzeilen...

Die Klasse „ReflexSensor“ erwartet bei der Objekterstellung neben dem Namen die Pinnummern, an denen LED und LDR angeschlossen sind, und den Schwellwert, ab dem ein Becher als erkannt gilt.
 Man kann auch den LDR alleine nutzen, dann ist ...

Mit den Methoden showLDR() und showValues() können ...

Methode „waitForCan()“ ...

Aufgabe: Erstelle in der Klassendefinition im h-Tab die Klasse ReflexSensor.

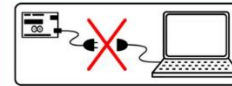
oder Du hast die Klassendefinitionsdatei „tumlabs.h“ bereits ausgeleitet bekommen, dann überspringe die nächsten Seiten mit Klick auf

Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang

Übersicht

Verkabelung - Lichtschränke



Bitte vor jedem Ändern der Verkabelung unbedingt den Arduino vom Computer trennen!

Arbeiten am Arduino nur in stromlosem Zustand!

Nun müssen LDR (Lichtsensoren) und LED (Lichtquelle) mit dem Arduino verbunden werden.

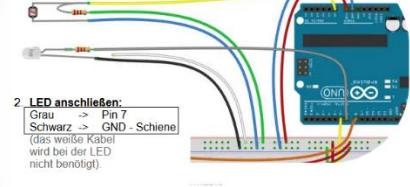
Nutze wieder die Versorgungsleisten (rot und blau) am Steckbrett:
am Arduino:
 rotes Kabel: V in - Pin → rote Schiene
 blaues Kabel: GND - Pin → blaue Schiene

1. LDR anschließen:

Gelb → Pin A0
 Grün → V in - Schiene
 Blau → GND - Schiene

2. LED anschließen:

Grau → Pin 7
 Schwarz → GND
 (das weiße Kabel wird bei der LED nicht benötigt)



Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang

Übersicht

Programmierung: Die Klasse „ReflexSensor“



ReflexSensor

- name: String
- pinLEDnumber: int
- pinLDRnumber: int
- trigger: int
- ReflexSensor(pinLED, pinLDR, trigger)
- ReflexSensor(pinLDR)
- showLDR(void): void
- showValues(void): void
- waitForCan(void): void
- detectCan(void): bool



Die Klasse „ReflexSensor“ erwartet bei der Objekterstellung ...

Man kann auch den LDR alleine nutzen; Dann ist ...

Mit den Methoden showLDR() und showValues() können ...

Methode „waitForCan()“ ...

Aufgabe: Erstelle in der Klassendefinition im h-Tab die Klasse ReflexSensor.

oder Du hast die Klassendefinitionsdatei „tumlabs.h“ bereits ausgeleitet bekommen, dann überspringe die nächsten Seiten mit Klick auf

Hands-on-Remote

- Vorab
- Struktur
- Dreh Scheibe
- Dispenser
- Lichtschränke
- Auswerfer
- Eingang

Übersicht

Programmierung: Die Klasse „ReflexSensor“



ReflexSensor

- name: String
- pinLEDnumber: int
- pinLDRnumber: int
- trigger: int
- ReflexSensor(pinLED, pinLDR, trigger)
- ReflexSensor(pinLDR)
- showLDR(void): void
- showValues(void): void
- waitForCan(void): void
- detectCan(void): bool



Fahre mit der Maus über die Textzeilen...

Die Klasse „ReflexSensor“ erwartet bei der Objekterstellung neben dem Namen die Pinnummern, an denen LED und LDR angeschlossen sind, und den Schwellwert, ab dem ein Becher als erkannt gilt.
 Man kann auch den LDR alleine nutzen; Dann ist neben dem Namen nur die Pinnummer des LDR nötig.

Mit den Methoden showLDR() und showValues() können ...

Methode „waitForCan()“ ...

Aufgabe: Erstelle in der Klassendefinition im h-Tab die Klasse ReflexSensor.

oder Du hast die Klassendefinitionsdatei „tumlabs.h“ bereits ausgeleitet bekommen, dann überspringe die nächsten Seiten mit Klick auf

Hands-on-Remote

Programmierung: Die Klasse „ReflexSensor“

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

ReflexSensor

- name: String
- pinLEDnumber: int
- pinLDRnumber: int
- trigger: int

- ReflexSensor(pinLED, pinLDR, trigger)
- ReflexSensor(pinLDR)
- showLDR(void): void
- showValues(void): void
- waitForCan(void): void
- detectCan(void): bool

Fahre mit der Maus über die Textzeilen...

Die Klasse „ReflexSensor“ erwartet bei der Objekterstellung ...

Man kann auch den LDR alleine nutzen, Dann ist ...

Mit den Methoden showLDR() und showValues() können die Messwerte, die der LDR zurückgibt, grafisch angezeigt werden.

Methode „waitForCan()“ ...

```

// Licht-Sensor-Objekte erzeugen:
ReflexSensor xxx (A0);
ReflexSensor yyy (7, A0, 30);

void setup() {
  xxx.showLDR(); //
  yyy.showValues();
}

void loop() {
  xxx.waitForCan(); // wartet auf Becher
  if (xxx.detectCan() == HIGH) {
    //
  }
}

```

Aufgabe: Erstelle in der Klassendefinition im h-Tab die Klasse ReflexSensor.

oder Du hast die Klassendefinitionsdatei „umlab.h“ bereits ausgeteilt bekommen, dann überspringe die nächsten Seiten mit Klick auf

Hands-on-Remote

Programmierung: Die Klasse „ReflexSensor“

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

ReflexSensor

- name: String
- pinLEDnumber: int
- pinLDRnumber: int
- trigger: int

- ReflexSensor(pinLED, pinLDR, trigger)
- ReflexSensor(pinLDR)
- showLDR(void): void
- showValues(void): void
- waitForCan(void): void
- detectCan(void): bool

Fahre mit der Maus über die Textzeilen...

Die Klasse „ReflexSensor“ erwartet bei der Objekterstellung ...

Man kann auch den LDR alleine nutzen, Dann ist ...

Mit den Methoden showLDR() und showValues() können ...

Die Klasse „ReflexSensor“ stellt die Methode „waitForCan()“ zur Verfügung, die den Programmablauf solange anhält, bis ein Becher erkannt wird.

Aufgabe: Erstelle in der Klassendefinition im h-Tab die Klasse ReflexSensor.

oder Du hast die Klassendefinitionsdatei „umlab.h“ bereits ausgeteilt bekommen, dann überspringe die nächsten Seiten mit Klick auf

Hands-on-Remote

Programmieren: Lichtschranke einrichten

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

Die Empfindlichkeit der Lichtschranke wird über den Schwellwert festgelegt. Um den Schwellwert herauszufinden, kannst Du Dir die Messwerte des Lichtsensors in der Plottergrafik anzeigen lassen.

Arbeite im ino-Tab.

- 1) Erzeuge ein neues Objekt...
- 2) Ermittle den Schwellwert: ...
- 3) Öffne den Seriellen Plotter ...
- 4) und bestimme ...
- 5) Beachte ...
- 6) Ergänze nun ...

```

// Erweiterung des Programms
//
ReflexSensor lightBarrier(7,A0,30);
void setup() {
  ...
  lightBarrier.showValues();
  // nach der Messung diese Zeile
  // wieder kommentieren!
}
void loop() {
  ...
}

```

Hands-on-Remote

Programmieren: Lichtschranke einrichten

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

Die Empfindlichkeit der Lichtschranke wird über den Schwellwert festgelegt. Um den Schwellwert herauszufinden, kannst Du Dir die Messwerte des Lichtsensors in der Plottergrafik anzeigen lassen.

Arbeite im ino-Tab.

- 1) Erzeuge ein neues Objekt der Klasse ReflexSensor mit Namen, Pins und Schwellwert. Verwende erstmal den Wert 30.
- 2) Ermittle den Schwellwert: ...
- 3) Öffne den Seriellen Plotter ...
- 4) und bestimme ...
- 5) Beachte ...
- 6) Ergänze nun ...

Als Schwellwert bitte den Schätzwert 30 eingeben und im Schritt (4) anpassen.

```

// Erweiterung des Programms
//
ReflexSensor lightBarrier(7,A0,30);
void setup() {
  ...
}
void loop() {
  ...
}

```

Hands-on-Remote

Programmieren: Lichtschranke einrichten

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

Die Empfindlichkeit der Lichtschranke wird über den Schwellwert festgelegt. Um den Schwellwert herauszufinden, kannst Du Dir die Messwerte des Lichtsensors in der Plottergrafik anzeigen lassen.

Arbeite im ino-Tab.

- 1) Erzeuge ein neues Objekt...
- 2) Ermittle den Schwellwert: ...
- 3) Öffne den Seriellen Plotter Lasse Becher die Lichtschranke passieren
- 4) und bestimme ...
- 5) Beachte ...
- 6) Ergänze nun ...

```

// Erweiterung des Programms
//
ReflexSensor lightBarrier(7,A0,30);
void setup() {
  ...
}
void loop() {
  ...
}

```

Hands-on-Remote

Programmieren: Lichtschranke einrichten

Vorab
Struktur
Drehseibe
Dispenser
Lichtschranke
Auswerfer
Eingang
Übersicht

Die Empfindlichkeit der Lichtschranke wird über den Schwellwert festgelegt. Um den Schwellwert herauszufinden, kannst Du Dir die Messwerte des Lichtsensors in der Plottergrafik anzeigen lassen.

Arbeite im ino-Tab.

- 1) Erzeuge ein neues Objekt...
- 2) Ermittle den Schwellwert: ...
- 3) Öffne den Seriellen Plotter ...
- 4) und bestimme aus der Plottergrafik einen geeigneten Schwellwert und ersetze die 30 durch ihn.
- 5) Beachte ...
- 6) Ergänze nun ...

Schwellwert

Becher passieren Lichtschranke

```

// Erweiterung des Programms
//
ReflexSensor lightBarrier(7,A0,30);
void setup() {
  ...
}
void loop() {
  ...
}

```



**CENTRUM
NAUKI
KOPERNIK**



Technische Universität München

Alle Materialien sind erhältlich unter

<https://erasmus-plus.ec.europa.eu/projects/search/details/2020-1-DE02-KA226-VET-008295>

Das Projekt Hands-on Remote wurde im Rahmen des Programms Erasmus+ KA226 Partnerships for Digital Education Readiness finanziert (2020-1-DE02-KA226-VET-008295).

Kofinanziert durch das
Programm Erasmus+
der Europäischen Union

