



empfohlen für
Sekundarstufe II

Klaus Trimborn, Dipl.-Ing. Claas Niehues, Micaela Krieger-Hauwede

STEM Coding Ultimate

Robotik, KI und moderne Technik realitätsnah
und handlungsorientiert entdecken

Klett MEX

fischertechnik 

STEM Coding Ultimate



2–4 Schülerinnen und Schüler



224 Bauteile inkl. Ersatzteilebeutel und einfacher Rücksortierung



10 Modelle mit 13 Lerneinheiten und 96+ Std. Unterrichtsmaterial



Inkl. TXT 4.0 Controller, 2 × Encodermotor, 2 × Mini-Taster, 5 × LED, Fototransistor, NTC-Widerstand, Spursensor, Ultraschallsensor, USB-Kamera, 9-V-Akku mit USB-C Ladeanschluss und Ladekabel



Noch mehr Infos, Erste-Schritte-Anleitungen und Video-Tutorials finden Sie hier:

→ www.fischertechnik.de/stem-coding-ultimate



Alle Aufgaben und Bauanleitungen sind in der **fischertechnik App „STEM Suite“** integriert.

Einfach die kostenlose App downloaden und loslegen.

Die App ist für iOS, Android, Windows und macOS verfügbar.

IMPRESSUM

fischertechnik 

→ www.fischertechnik.de/schulen

Klett MEX

→ www.klett-mex.de

Projektleitung: Tobias Schlipf, Ann-Christin Walker, fischertechnik GmbH

Umsetzung: Tanja Zunder, Klett MEX GmbH

Autoren: Klaus Trimborn, Dipl.-Ing. Claas Niehues, Micaela Krieger-Hauwede

Redaktion: Micaela Krieger-Hauwede, Leipzig

Gestaltung: Tanja Bregulla, Orangepixel

Didaktisches Konzept, pädagogische Beratung: Klett MEX GmbH

© fischertechnik GmbH, Waldachtal, und Klett MEX GmbH, Stuttgart

Hinweise für Lehrkräfte 4

MODELL 1

Lerneinheit 1  

Analoge Sensoren

Stimmen die Parameter? 8

MODELL 2

Lerneinheit 2  

Spritzenpumpe

Auf die Dosis kommt es an! 13

MODELL 3

Lerneinheit 3  

Tormodell

Zugang gestattet? 18

MODELLE 4 und 5

Lerneinheit 4  

Lerneinheit 5  

Lerneinheit 6  

Parkhaus 1-3

Ist das Parkhaus voll? 24

MODELLE 6 und 7

Lerneinheit 7  

Lerneinheit 8  

Lerneinheit 9 (KI)  

Ampelsteuerung 1-3

Fließt der Verkehr? 31

MODELLE 8 bis 10

Lerneinheit 10  

Lerneinheit 11  

Lerneinheit 12  

Lerneinheit 13 (KI)  

Fahrerloses

Transportfahrzeug 1-4

Sicher unterwegs im Parcours 41

Wo ist was verbaut?

Aktoren und Sensoren 54

BEDEUTUNG DER ICONS

Schwierigkeitsgrad Modell:

 leicht

 mittel

 schwer

Schwierigkeitsgrad Programmierung:

 leicht

 mittel

 schwer

Unterricht:

 mit App

 ohne App

Handlungs- und
problemorientiertes Konstruieren
und Programmieren in der
Sekundarstufe II anschaulich
gemacht!

Hinweise für Lehrkräfte

STEM Coding Ultimate

- Alltagsaufgaben der Informationstechnik und Robotik realisieren
- Kompetenzausrichtung nach dem 4K-Modell:
Kreativität, Kollaboration, Kommunikation und kritisches Denken
- Blockly- und Python-Programmierung mit vielseitig einsetzbarem TXT 4.0 Controller
- 224 Bauteile, inkl. Ersatzteilebeutel und einfacher Rücksortierung



Unterrichtsmaterial
für über 96 Unterrichtsstunden



Anzahl Schülerinnen und Schüler
2–4 pro Baukasten



Zeitaufwand
Alle Grundaufgaben können in
2–4 Doppelstunden gelöst werden.



Anzahl Modelle
10 (mit 13 Lerneinheiten)



Lernziele

- Grundlagen der Informatik und Robotik ausbauen
- Funktionsweise von Aktoren und komplexen Sensoren verstehen
- Grundkenntnisse der neuronalen Netze und KI-Programmierung erwerben



Altersstufe
Sekundarstufe II

◉ LERNMATERIAL

Digitale Kompetenzen und ein Grundverständnis im Bereich des komplexeren Programmierens gehören heute zum Aufbauwissen, das Schülerinnen und Schüler (SuS) in der Sekundarstufe II erhalten müssen. Mit STEM Coding Ultimate kann dieses Aufbauwissen anhand von motivierenden und didaktisch aufbereiteten Alltagsbeispielen erworben werden.

In selbstständiger und durch eine App angeleiteter Weise lernen SuS beginnend mit einer überschaubaren Grundaufgabe über alltagsnahe Anwendungen auch komplexere schaltungstechnische Herausforderun-

gen und Programmieraufgaben zu realisieren. Die beiden Grundaufgaben **Parkhaus 1: Parkhausschranke** und **Ampelsteuerung 1: Lichtsignalanlage** lassen sich flexibel sowohl zu Teamaufgaben in **Parkhaus 2: Einfahrt und Ausfahrt** und **Ampelsteuerung 2: Verkehrskreuzung** als auch zu anspruchsvollen Programmieraufgaben in **Parkhaus 3: Kennzeichenerkennung** und **Ampelsteuerung 3: KI-Phasensteuerung** erweitern.

Vier Lerneinheiten zu drei Fahrzeugmodellen ermöglichen es, je nach Leistungsniveau der SuS auf Fragen zum Thema autonomes Fahren gezielt einzugehen (einfaches Fahr-

training, digitale oder analoge Steuerung eines FTFs, KI-gestütztes Linienfolgen unter Verwendung eines neuronalen Netzes).

Die einzelnen Lerneinheiten in STEM Coding Ultimate widmen sich den Fragen: Wie lernt man, einen vielseitig einsetzbaren TXT 4.0 Controller mit Sensoren und Aktoren auszustatten? Wie kann man zur Lösung komplexer Aufgaben mehrere TXT 4.0 Controller koppeln? Wie kann man eine automatische Spritzenpumpe zur Feindosierung von Medikamenten realisieren? Wie lässt sich ein sensorgesteuertes, automatisches Zugangs-

tor programmieren? Wie kann man eine einfache Parkhausschranke in Teamarbeit zu einem Parkhausystem mit getrennter Einfahrt und Ausfahrt erweitern? Wie lässt sich eine Kennzeichenerkennung mit automatischer Parkpreisberechnung realisieren? Wie funktioniert die Steuerung einer einfachen Lichtsignalanlage? Wie koordiniert man zwei Lichtsignalanlagen zu einer Verkehrskreuzung? Was sind die Grundlagen des autonomen Fahrens? Welche Sensoren kommen dabei zum Einsatz? Wie lassen sich neuronale Netze und KI bei der Ampelsteuerung und dem autonomen Fahren einsetzen?

○ PRODUKTBESCHREIBUNG

Anhand von zehn alltagsnahen Modellen und unabhängig voneinander realisierbaren Lerneinheiten lernen SuS kleinschrittig und problemorientiert spannende Aufgabenstellungen aus dem informationstechnischen Bereich zu realisieren. Zwei der Modelle lassen sich zu Teamaufgaben erweitern, bei denen Grundkenntnisse in der Kopplung zweier TXT 4.0 Controller erworben werden können. Darüber hinaus bieten zwei der Modelle Lerneinheiten zum Einsatz neuronaler

Netze und KI. Mithilfe eines vielseitig einsetzbaren TXT 4.0 Controllers, einer Reihe praktischer Sensoren und Aktoren, einer intuitiven App sowie funktional hochwertigen fischertechnik Bausteinen lösen SuS der Sekundarstufe II Aufgaben, die sie aus ihrer eigenen Lebenswelt kennen. Durch den handlungsorientierten und problemorientierten Lernansatz erarbeiten sich SuS auf unterschiedlichen Niveaustufen fachliche, methodische, personelle und soziale Kompetenzen.

○ BILDUNGSPLANBEZÜGE

Die Bezüge zu den Bildungsplänen/Kernlehrplänen beruhen auf dem gemeinsamen Referenzrahmen für Naturwissenschaften, Informatik und Technik (GeRRT). Dieser bildet die Grundlage für eine länderübergreifende Vergleichbarkeit der Technik- und Informatikkompetenzen sowie für die Entwicklung entsprechender Curricula, die Lehrwerkserstellung, die Konzeption von schulischen Unterrichtsmaterialien oder die Entwicklung von außerschulischen Lehr- und Lernkonzepten.

Der gemeinsame Referenzrahmen unterteilt sich in fünf Kompetenzbereiche. Diese fünf Bereiche liegen den Projekten zugrunde.



Technik verstehen:

- Ein komplexes technisches System funktional, hierarchisch und strukturell analysieren und erfassen.
- An einem einfachen Beispiel erkennen, dass jede Technikentwicklung und jeder Technikeinsatz Folgen für die Gesellschaft hat.

Technik nutzen:

- Ein alltagstypisches technisches System zweckentsprechend anhand relevanter technischer Kriterien auswählen.
- Ggf. das Nichtfunktionieren des technischen Systems sachgerecht analysieren und mögliche Handlungsschritte zur Lösung des Problems erkennen und realisieren.

Technik entwickeln:

- Ein technisches Problem durch Bedarfsanalyse und Situationsanalyse in Teilprobleme gliedern und die technische Lösbarkeit der Teilprobleme abschätzen.
- Eigene sachgerechte Lösungen für Teilprobleme, auch auf Basis vorhandener Lösungen, entwickeln bzw. weiterentwickeln.
- Pläne für die Teillösungen durch Anpassung vorhandener Pläne oder eigene Teillösungen auch mit digitalen Werkzeugen entwickeln.
- Mit den Teillösungen die Gesamtlösung unter Verwendung der benötigten (auch digitalen) Werkzeuge, Geräte, Materialien und Maschinen realisieren.

Technik bewerten:

- Eine einfache technisch geprägte Situation nach individuellen Kriterien bewerten.

Technik kommunizieren:

- Technische Darstellungen und Bewertungen für das technische System zielgerichtet beschaffen.
- Eine vorhandene technische Beschreibung und Anleitung für das System in Sprache und/oder Bild lesen und verstehen.
- Das technische System mit standard-orientierten technischen Darstellungen sowie eigene technische Lösungen unter Verwendung der Fachsprache adressatenabhängig beschreiben.

○ WICHTIGSTE LERNZIELE

- Alltagsaufgaben der Informationstechnik und Robotik realisieren.
- Modellvorstellungen auf Alltagsanwendungen übertragen.
- Projekt- und Gruppenarbeit im Team üben und vertiefen.
- Ausbilden von fachlichen, methodischen, personellen und sozialen Kompetenzen.
- Programmabläufe und komplexe informatische Strukturen kennenlernen.
- Blockly- oder Python-Programmierung für einen vielseitig einsetzbaren TXT 4.0 Controller erlernen.
- Funktionsweise und Vernetzungen von Aktoren und Sensoren verstehen und anwenden.
- Vorberuflich in mathematischer, naturwissenschaftlicher, (informations-)technischer Richtung orientieren.

HINWEISE FÜR DIE ARBEIT MIT DEN MATERIALIEN

Die grundsätzlichen Arbeits- und Verbindungstechniken von fischertechnik sowie Funktionsweise und Aufbau eines Programms werden in der ersten einführenden Lerneinheit zu den analogen Sensoren wiederholt. Dreizehn Lerneinheiten zu zehn Modellen stehen zur Verfügung. Drei der Lerneinheiten lassen sich zu Teamaufgaben bzw. komplexen Programmieraufgaben erweitern. Mit den Materialien lassen sich auch eigene Lösungen zu Themenstellungen aus dem Alltag realisieren. Zu allen Modellen werden Informationen zu Umsetzung, Medien, Organisationsformen oder zur Differenzierung angeboten.

Die Materialien für Lehrpersonen sind nach dem folgenden allgemeinen Muster aufgebaut:

- Leitfragen zum Thema,
- Darstellung der Unterrichtsidee,
- Beschreibung der Aufgabe,
- Alltagsbezug,
- Fächerbezug im Rahmen des Curriculums,
- Unterrichtsverlauf, ggf. in aufeinander aufbauende Lerneinheiten unterteilt,
- methodisch-didaktische Hinweise mit Differenzierungsmöglichkeiten und motivationalen Aspekten,
- Programmierkenntnisse,
- Hinweise zu Zusatzmaterialien,
- Funktionsbeschreibungen der einzelnen Modelle sowie
- Materiallisten.

Auch die Aufgaben für die SuS sind nach einem wiederkehrenden Muster aufgebaut:

- Storytelling zur Einführung,
- einführende und vertiefende Aufgaben in das Thema,
- Konstruktions- und Bauphase des Projekts,
- Anleitung zur Übertragung des Programmcodes an den TXT 4.0 Controller,
- ausführliche Entwicklung und Erläuterung des Programmcodes,
- parallel dazu schrittweise Inbetriebnahme des Modells,
- Hilfestellung zur Fehlersuche,
- Differenzierungsmöglichkeiten sowie
- ein Abschnitt mit Infos zu verwandten Themen oder Kuriositäten zum Abschluss des Projekts.



Noch mehr Infos,
Erste-Schritte-Anleitungen
und Video-Tutorials
finden Sie hier:
→ [www.fischertechnik.de/
stem-coding-ultimate](http://www.fischertechnik.de/stem-coding-ultimate)

HINWEISE FÜR DIE ARBEIT MIT DER APP

Alle Aufgaben und Bauanleitungen sind in der fischertechnik App „STEM Suite“ integriert.

Einfach die kostenlose App downloaden und loslegen. Die App ist für iOS, Android, Windows und macOS verfügbar.

Die Projekte und Aufgaben in der App werden nach dem oben genannten Muster von den SuS selbstständig bearbeitet. Die Aufgaben und Lösungswege werden dabei so angeboten, dass alle SuS zu einem funktionierenden Ergebnis (Modell) kommen und ein Erfolgserlebnis haben.

Analoge Sensoren

Stimmen die Parameter?



LEITFRAGEN:

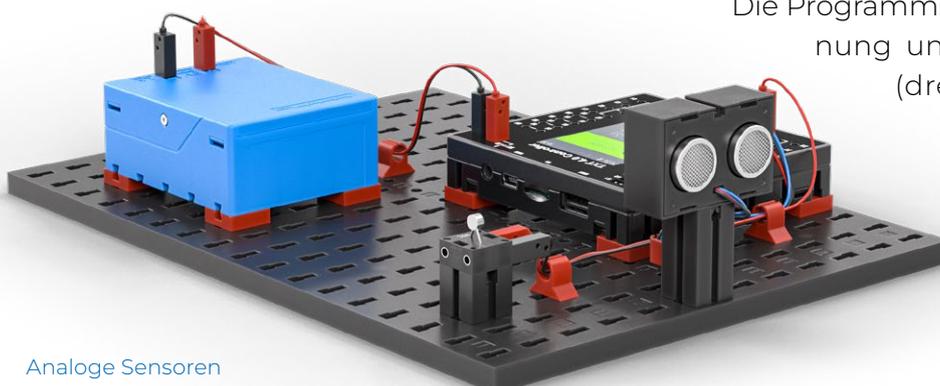
- Wie kann man mithilfe der Sensoren Widerstände, Temperatur und Abstände messen? (*Kommunikation und Information*)
- Mit welchen Sensoren lässt sich die Produktion sinnvoll überwachen? (*Kollaboration*)
- Unter welchen Bedingungen soll das System über Abweichungen von der Norm informieren? (*kritisches Denken*)
- Was ist zu berücksichtigen, damit die Sensorik für verschiedene Anwendungsszenarien genutzt werden kann und das System möglichst robust funktioniert? (*Kreativität*)

○ DIE UNTERRICHTSIDEE AUF EINEN BLICK

Klassenstufe:	11–13
Zeitaufwand:	2 Doppelstunden (erweiterbar bis zu 3 DS)
Schwierigkeitsgrad:	Modell  Programmierung 
Modellart:	Tischmodell mit analogen Sensoren

○ MODELLBESCHREIBUNG / AUFGABE

Die Schülerinnen und Schüler (SuS) planen und realisieren einen Versuchsaufbau für die Experimente mit analogen Sensoren. Ausgehend von der einfachen Widerstandsanzeige im Schnittstellentest entwickeln sie eine eigene grafische Anzeige auf dem Display des TXT 4.0 Controllers und messen Widerstand und Temperatur.



Analoge Sensoren

Die Programmierung wird um die Berechnung und Anzeige von Farbcodes (drei- oder vierstellig, Differenzierung 1 und 2) für Schichtwiderstände erweitert.

ALLTAGSBEZUG

Die Messung von Widerständen und Temperaturen ist den SuS aus dem naturwissenschaftlichen Unterricht bzw. ihrem Alltag bekannt.

Insbesondere die Einbettung in einen realitätsnahen Anwendungskontext weist einen hohen motivationalen Wert auf.

Eine Integration der Thematik in die vorberufliche Orientierung könnte im Hinblick auf informationstechnische Berufsfelder erfolgen. Hier wird das automatisierte Erfassen physikalischer Größen in vielen Bereichen genutzt. In besonderer Weise wird die automatisierte Erwärmung und/oder Befüllung nicht nur in der Industrie genutzt, sondern auch verstärkt im häuslichen Umfeld.

FÄCHERBEZUG

Informatik:	Fortgeschrittene Programmierung, Bedingungsschleifen, Funktionen, grafische Anzeigen
Physik:	Widerstand im Stromkreis, Temperaturen, Messungen von Strecken
Technik:	Stabiles Bauen, Konstruktionstechnik
Mathematik:	Berechnung von Termen, Rechnen mit Einheiten

UNTERRICHTSVERLAUF

Einführungsphase



Unterrichtsgespräch

- Bekanntgeben des Themas, ggf. Video „Befüllvorgang eines Additivgebindes“ zeigen:
→ www.youtube.com/watch?v=k5pr1E8ihe4
- Abfragen, was die wesentlichen Merkmale der Qualitätssicherung bei der Abfüllung/Produktion von Additiven sind.
- Messgrößen und Szenarien abfragen, in denen automatisch messende Befüllsysteme eingesetzt werden.
- Einsatzmöglichkeiten der gesammelten Szenarien diskutieren (z. B. Kaffeemaschine, Küchenmaschine, Spülmaschine).
- Anforderungen an den Versuchsaufbau ermitteln.



ggf. Hilfestellung

- Sensoren und Bauteile aus dem Baukasten zeigen, wenn nötig, Präsentationsmedien einsetzen.

Planungsphase



Unterrichtsgespräch

- Die Vorgehensweise zum Bau des Modells und die zu erzielende Funktion werden gemeinsam erarbeitet.
- Die Arbeitsschritte der App werden vorgegeben bzw. besprochen.



Partner- oder Einzelarbeit

- Die SuS machen sich mit der App vertraut und laden die entsprechende Aufgabe.
- Die SuS definieren sinnvolle Funktionen einer Messung mithilfe der an den TXT 4.0 Controller angeschlossenen Sensoren.
- Die SuS erstellen mittels App die Anforderungsliste für den zu bauenden Versuch.



Optional:
Partner- oder Gruppenarbeit

- Die SuS skizzieren die möglichen Aufbauten der Messung mit den Sensoren (Widerstände, Ultraschallsensor).
- Die SuS diskutieren die Ergebnisse in der Gruppe und legen sich auf ein Design fest.

Konstruktionsphase



Partner- oder Einzelarbeit

- Die SuS nutzen die App zum Versuchsaufbau. Die App führt kleinschrittig durch den Aufbau.

Programmierphase



Partner- oder Einzelarbeit

- Die SuS schreiben das Programm für die mit dem Versuchsaufbau durchzuführenden Messungen. Die App führt hier in aufeinander aufbauenden Schritten durch die Programmieraufgabe; Hilfe wird in der App angeboten.
- Das Programm wird nach jedem Differenzierungsschritt auf den TXT 4.0 Controller übertragen.

Experimentier- und Testphase



Partner- oder Gruppenarbeit

- Die Anlage wird in Betrieb genommen.
- Mögliche Störungen im Funktionsablauf müssen gefunden und eliminiert werden.
- Eventuelle Fehlersuche ist mithilfe von Vorschlägen in der App möglich.
- Eventuelle Optimierungen bei der Hardware und der Programmierung können vorgenommen werden.

Abschlussphase



Optional:
Vorstellung und Zuteilung der Differenzierungen

- Für die Differenzierung infrage kommende SuS werden ggf. durch die Lehrperson angesprochen. Dabei wird die grafische Darstellung der Farbcodierung von Widerständen vorgestellt.
- Die App bietet konkrete Ideen für die interessierten SuS, darunter die Anzeige von drei- oder vierstelligen Farbcodes auf dem Display des TXT 4.0 Controllers.



Diskussion im Plenum

- Nachbesprechung des Projekts im Klassenverbund
- Klärung von zukünftigen Einsatzmöglichkeiten im Alltag

METHODISCH-DIDAKTISCHE HINWEISE

Differenzierungsmöglichkeiten

Je nach Dauer der Unterrichtsreihe und der Stärke der SuS kann

- das Ergebnis der Widerstandsmessung nicht nur als Zahl, sondern auch auf einer Skalanzeige sowie als drei- oder vierstelliger Farbcode ausgegeben werden,
- der Versuchsaufbau verändert werden, um Temperaturmessungen an gewählten Objekten durchführen zu können,
- der Versuchsaufbau geändert werden, um Füllstände von Trinkgefäßen bei der Befüllung messen und anzeigen zu können.

Motivationale Aspekte

Die Themen „automatische Befüllung“ und „Qualitätsüberwachung“ sind allen SuS aus dem Alltag bekannt. In vielen Haushalten gehören Kaffee- und Waschmaschinen neben vielen weiteren smarten Anwendungen längst zum Alltag. Teilautonom kochende Küchenmaschinen finden immer häufiger Anwendung bei der Zubereitung von Gerichten.



PROGRAMMIERKENNTNISSE

- Programmstart
- Dauerschleife **wiederhole dauerhaft (repeat forever)**
- Einbindung von Sensoren
- Programmierung der Anzeige auf dem Display des TXT 4.0 Controllers
- Schleife **wiederhole solange (repeat while)**
- Befehl **warte (wait)**
- Nutzen von Variablen und deren Veränderung
- Arbeit mit Unterprogrammen

ZUSATZMATERIALIEN

Zum Download optional:

- Stromlaufplan
- Bauanleitung

- Wenn vorhanden, könnte für die Einführungsphase in das Thema ein Video genutzt werden.
- Zeichenmedien (Papier, Whiteboard oder Projektionsfläche).

—○ FUNKTIONEN DES MODELLS UND DEREN TECHNISCHE LÖSUNGEN

Funktion der Sensoren/Aktoren	Technische Lösung
Messung von Widerständen	Messen der Leitfähigkeit (Qualitätskriterium) des Additivs
Messung von Temperaturen	Überwachung der Befülltemperatur
Messung von Abständen	Überwachung der Befüllmengen
Differenzierungsmöglichkeiten für weiterführenden Unterricht	<ul style="list-style-type: none"> • Ergänzen der Ziffernanzeige durch eine Skalenanzeige • Einbindung von Farbcodes und/oder anderen Widerstandstypen • Ausgabe eines Alarms bei Unter- oder Übertemperatur • Messung von Abständen und Berechnung der Differenz (Minimum – Maximum)

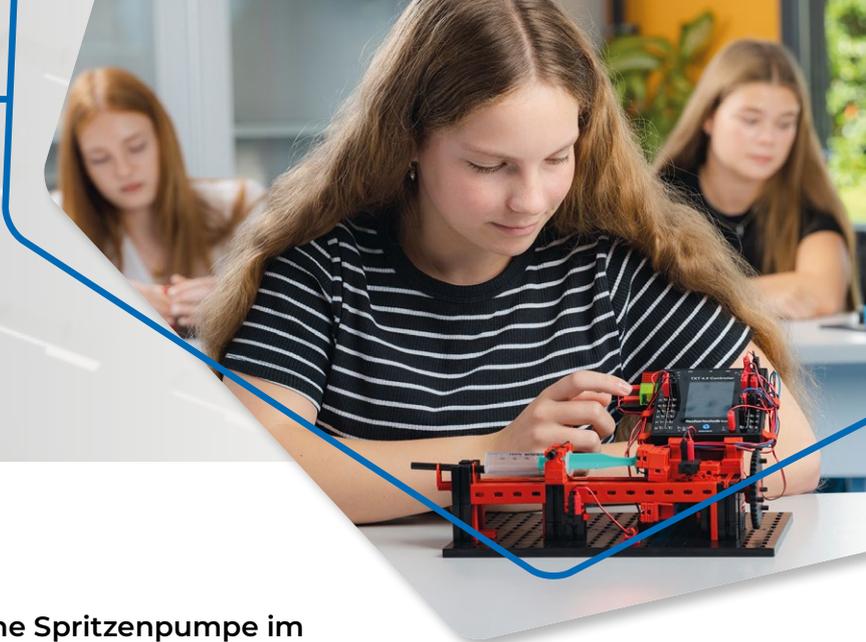
—○ MATERIALLISTE

Sensoren	Funktion
1 On/Off-Taster am TXT 4.0 Controller	Einschalten/Start der Messung/ Stopp der Messung
1 NTC-Widerstand	Widerstandsmessung, Temperaturbestimmung
1 Ultraschallsensor	Abstandsmessung

Aktoren	Funktion
1 Display des TXT 4.0 Controllers	Visualisierung der Messergebnisse (in Form von Ziffern oder als Grafik)

Spritzenpumpe

Auf die Dosis kommt es an!



LEITFRAGEN:

- Wo ist eine automatische medizinische Spritzenpumpe im Alltag einsetzbar? (*Kommunikation*)
- Welche Funktionen muss die Steuerung der Spritzenpumpe sinnvollerweise erfüllen? (*Kollaboration*)
- Unter welchen Bedingungen soll das System die Spritze bewegen? (*kritisches Denken*)
- Was ist zu berücksichtigen, damit die Steuerung an verschiedenen Standorten genutzt werden kann und das System möglichst robust funktioniert? (*Kreativität*)

DIE UNTERRICHTSIDEE AUF EINEN BLICK

Klassenstufe:	11–13
Zeitaufwand:	3 Doppelstunden (erweiterbar bis zu 6 DS)
Schwierigkeitsgrad:	Modell  Programmierung 
Modellart:	Tischmodell für Spritzenpumpen

MODELLBESCHREIBUNG / AUFGABE

Die Schülerinnen und Schüler (SuS) planen und realisieren eine medizinische Spritzenpumpe zur Feindosierung von Medikamenten.

Die Bedienelemente wie Auswahlbuttons und Anzeigeelemente werden auf dem Touch-Display programmiert. Die Dosierung wird über die schrittweise Ansteuerung eines Encodermotors präzise erreicht.

Zwei LEDs geben Lichtsignale bei Dosierungsbetrieb (grün) bzw. in der Endstellung und bei zu geringem Flüssigkeitsvolumen für eine Dosis (rot). Über Touch-Buttons werden schrittweise verschiedene Funktionen wie „Start“, „Rücklauf“, „Dosisvorwahl“ und



Spritzenpumpe

„Injektionszeit“ implementiert. Aus der Anzahl der Motorumdrehungen werden die jeweils aktuell injizierte Dosis sowie die Restmenge des Medikaments in der Spritze ermittelt und auf dem Display ausgegeben. Reicht die Restmenge für eine vorgewählte Dosis nicht aus, wird der Start unterbunden. Ebenso stoppt der Motor, wenn die Spritze vollständig eingeschoben ist. In beiden Fällen blinkt die rote LED.

Bei gezielter Anforderung zum Wechsel der Spritze sowie automatisch beim Programmstart wird der Spritzenantrieb in die Startposition zurückgefahren und stoppt automatisch.

○ ALLTAGSBEZUG

Medizinische Spritzenpumpen sind präzise Dosiergeräte, die Medikamente oder Flüssigkeiten in genau kontrollierten Mengen über einen bestimmten Zeitraum verabreichen. Sie kommen in der Medizin häufig zum Einsatz, zum Beispiel bei Narkosen, in der Intensivmedizin oder bei der Schmerztherapie.

Darüber hinaus können Spritzenpumpen zum präzisen Dosieren in biologischen oder chemischen Experimenten und Herstellungsprozessen verwendet werden.

○ FÄCHERBEZUG

Informatik:	Fortgeschrittene Programmierung, Bedingungsschleifen, Funktionen
Physik:	Bewegungsänderung
Technik:	Stabiles Bauen, Konstruktionstechnik
Mathematik:	Bedingtes Zählen, Volumen- und Zeitberechnung
Biologie, Chemie:	Experimente mit Dosierungsaufgaben: In chemischen und biologischen Experimenten lässt sich die Spritzenpumpe fächerverbindend einsetzen, um präzise Dosierungen vorzunehmen.

○ UNTERRICHTSVERLAUF

Einführungsphase



Unterrichtsgespräch

- Bekanntgeben des Themas, ggf. Video „Der Perfusor und Dreiwegehahn“ zeigen:
→ www.youtube.com/watch?v=9eelQAbdFsA
- Wesentliche Merkmale der Spritzenpumpe abfragen.



ggf. Hilfestellung

- Sensoren, Aktoren und Bauteile aus dem Baukasten zeigen, wenn nötig, Präsentationsmedien einsetzen.

Planungsphase



Unterrichtsgespräch

- Die Vorgehensweise zum Bau des Modells und die zu erzielende Funktion werden gemeinsam erarbeitet.
- Die Arbeitsschritte der App werden vorgegeben bzw. besprochen.



Partner- oder Einzelarbeit

- Die SuS machen sich mit der App vertraut und laden die entsprechende Aufgabe.
- Die SuS definieren sinnvolle Funktionen einer automatischen Spritzenpumpe.
- Die SuS erstellen mithilfe der App die Anforderungsliste für die zu bauende Spritzenpumpe.



Optional:
Partner- oder Gruppenarbeit

- Optional skizzieren die SuS die möglichen Aufbauten der Spritzenpumpe mit weiterer Sensorik (Endtaster) und Aktorik (LED).
- Die SuS diskutieren die Ergebnisse in der Gruppe und legen sich auf ein Design fest.

Konstruktionsphase



Partner- oder Einzelarbeit

- Die SuS nutzen die App zum Bau der Spritzenpumpe. Die App führt kleinschrittig durchs Programm.

Programmierphase



Partner- oder Gruppenarbeit

- Die SuS schreiben das Programm für die Spritzenpumpe. Die App führt hier in aufeinander aufbauenden Schritten mit offenen Fragestellungen durch die Programmieraufgabe.
- Hilfe wird in der App angeboten.
- Das Programm wird nach jedem Differenzierungsschritt auf den TXT 4.0 Controller übertragen.

Experimentier- und Testphase



Partner- oder Gruppenarbeit

- Die Spritzenpumpe wird in Betrieb genommen.
- Mögliche Störungen im Funktionsablauf müssen gefunden und eliminiert werden.
- Eventuelle Fehlersuche ist anhand von Vorschlägen in der App möglich.
- Eventuelle Optimierungen bei der Hardware und der Programmierung können vorgenommen werden.

Abschlussphase



Diskussion im Plenum

- Nachbesprechung des Projekts im Klassenverbund
- Klärung von zukünftigen Einsatzmöglichkeiten im Alltag

○ METHODISCH-DIDAKTISCHE HINWEISE

Fachliche Erläuterung

In der medizinischen Praxis arbeiten professionelle Spritzenpumpen mit einer hochpräzisen Schrittdosierung. Anstatt eine Flüssigkeit kontinuierlich fließen zu lassen, wird der gewünschte Volumenfluss durch das wiederholte Verabreichen kleinster Einzeldosen (z. B. 0,01 ml oder sogar 0,001 ml) erreicht. Dieses Prinzip sorgt für maximale Genauigkeit und Sicherheit, insbesondere bei stark wirksamen Medikamenten in der Intensiv- oder Notfallmedizin.

Didaktische Vorteile

Modellhaft wird hier eine Mikrodosis von 0,1 ml vorgegeben. Die Dosierung in solchen diskreten Schritten lässt sich programmier-technisch besonders klar umsetzen, da sie gut in Schleifen oder zeitgesteuerte Routinen übersetzbar ist. Die Dosiergeschwindigkeit lässt sich einfach anpassen, indem man entweder die Anzahl oder die Frequenz der Mikrodosen verändert – Überlegungen dazu fördern ein aktives Verständnis für Regelung und Steuerung. Zudem ist das Verfahren für die SuS intuitiv nachvollziehbar: Es ist greifbarer zu überlegen, wie viele Einzelschritte notwendig sind, um beispielsweise 1 ml Flüssigkeit

abzugeben, als sich einen gleichmäßigen Fluss abstrakt vorzustellen.

Differenzierungsmöglichkeiten

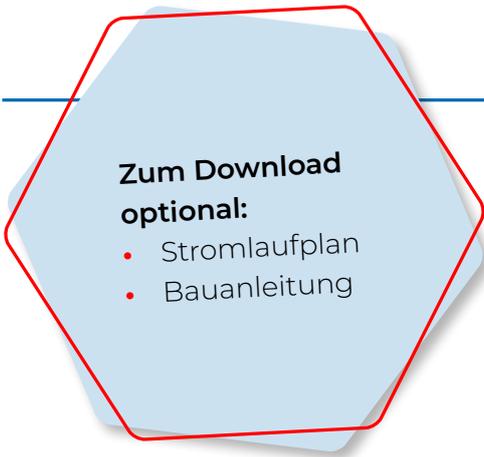
Je nach Arbeitsgeschwindigkeit der einzelnen SuS können Variationen in der Minimaldosis und der Injektionsgeschwindigkeit programmiert werden (Differenzierungsmöglichkeit 1). Weiterhin kann das Touch-Display um einen Slider erweitert werden, mit dem die Injektionszeit stufenlos eingestellt werden kann (Differenzierungsmöglichkeit 2).

Motivationale Aspekte

Die Arbeit mit Spritzenpumpen verbindet Technik mit sinnvoller Anwendung – hier geht es nicht nur um Motoren und Sensoren, sondern darum, wie Technik Leben retten kann. Für Jungen, gerade aber auch für Mädchen kann dieser medizintechnische Kontext besonders motivierend sein, weil er zeigt, dass Programmieren und Bauen nicht abstrakt, sondern sinnstiftend und gesellschaftlich relevant sind. Wer hier tüftelt, arbeitet an Lösungen, die Menschen helfen – das macht Technik greifbar, spannend und wertvoll.

○ PROGRAMMIERKENNTNISSE

- Programmstart
- Dauerschleife **wiederhole dauerhaft (repeat forever)**
- Einbindung von Sensoren und Aktoren
- Programmierung der Anzeige auf dem Display des TXT 4.0 Controllers
- Einbindung von Buttons auf dem Touch-Display
- Schleife **wiederhole n-mal (repeat n times)**
- Befehl **warte (wait)**
- Nutzen von Variablen und deren Veränderung
- Arbeit mit Unterprogrammen
- Arbeit mit Threads
- Arbeit mit Events



○ ZUSATZMATERIALIEN

- Wenn vorhanden, könnte für die Einführungsphase in das Thema ein Video genutzt werden.
- Zeichenmedien (Papier, Whiteboard oder Projektionsfläche).

—○ FUNKTIONEN DES MODELLS UND DEREN TECHNISCHE LÖSUNGEN

Funktion der Sensoren/Aktoren	Technische Lösung
Ausführen einer Drehung eines Encodermotors	Ansteuern des Antriebsmotors für eine Drehung mit festgelegter Schrittweite
Lichtsignale rote/grüne LED	Ansteuern der LED für Betriebs- und Fehlermeldungen
Berechnen und Anzeigen von Informationen auf dem Touch-Display	Programmierung und Auswertung von Mengen und Zeiten

○ MATERIALLISTE

Sensoren	Funktion
1 On/Off-Taster am TXT 4.0 Controller	1. Einschalten der Spritzenpumpe 2. Not-Stopp der Spritzenpumpe
2 Mini-Taster	Erkennen der Start- und Endposition
Aktoren	Funktion
1 Encodermotor	Bewegung
2 LEDs (1 × rot, 1 × grün)	Zustandsanzeige

Tormodell

Zugang gestattet?



LEITFRAGEN:

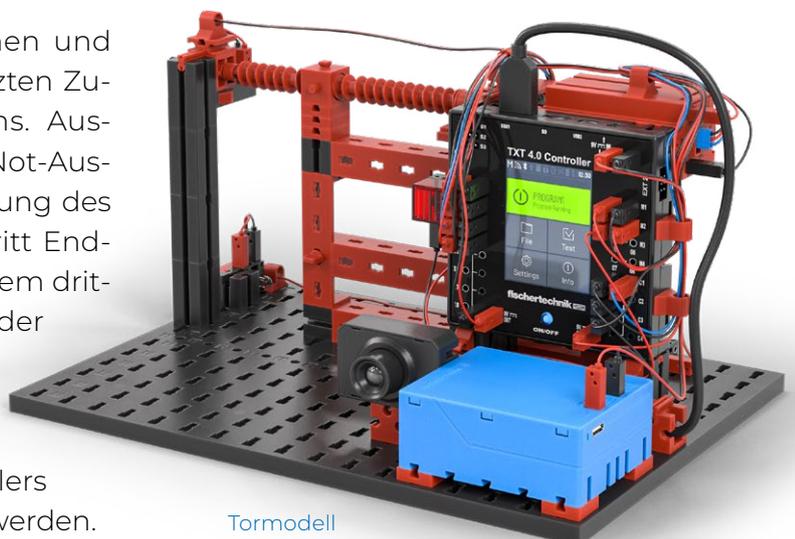
- Wie kann man mithilfe der Sensoren Endanschläge, Unterbrechungen und Abstände messen? *(Kommunikation und Information)*
- Mit welchen Sensoren lässt sich die Torsteuerung sinnvoll überwachen und steuern? *(Kollaboration)*
- Unter welchen Bedingungen soll das System den Anwenderschutz vor die Funktion stellen? *(kritisches Denken)*
- Was ist zu berücksichtigen, damit die Sensorik für verschiedene Anwendungsszenarien genutzt werden kann und das System möglichst robust funktioniert? *(Kreativität)*

DIE UNTERRICHTSIDE E AUF EINEN BLICK

Klassenstufe:	11–13
Zeitaufwand:	2 Doppelstunden (erweiterbar bis zu 8 DS)
Schwierigkeitsgrad:	Modell Programmierung
Modellart:	Tormodell zur Zugangssteuerung

MODELLBESCHREIBUNG / AUFGABE

Die Schülerinnen und Schüler (SuS) planen und realisieren ein Tormodell zur sensorgestützten Zugangskontrolle eines fiktiven Lagerraums. Ausgehend von einer einfachen, über einen Not-Aus-Schalter gesteuerten Öffnung und Schließung des Tormodells werden in einem zweiten Schritt Endtaster zum Schutz der Mechanik und in einem dritten Schritt Warn-LEDs zur Visualisierung der Aktivität des Torantriebs verwendet. Dabei soll eine einfache grafische Anzeige der Buttons „Open“, „Close“ und „Emergency Stop“ auf dem Display des TXT 4.0 Controllers programmiert und in Betrieb genommen werden.



Tormodell

Zur Vermeidung menschlicher Fehler bei der Überwachung des Schließvorgangs und der damit einhergehenden Betätigung des Buttons „Emergency Stop“ soll die Kontrolle des Schließens durch eine Lichtschranke automatisiert werden, die mögliche Blockierungen beim Zufahren des Tormodells detektiert und automatisch einen Not-Stopp herbeiführt. Parallel dazu informiert ein optisches Signal in Form von permanent rot leuchtenden LEDs über den Stopp des Schließvorgangs. Die erneute Freigabe der Schließung des Tores soll nur möglich sein, wenn sich kein Hindernis mehr im Tor befindet und der Button „Open“ auf dem Display des TXT 4.0 Controllers betätigt wurde. Gegenüber der ersten Lösung mit Endtastern stellt diese automatische Sicherheitsschaltung eine wesentliche Erweiterung in der Entwicklung der Automatisierung des Tormodells dar.

In der nächsten Stufe soll die automatisierte Öffnung des Tores bei Annäherung einer Person von innen durch die Verwendung eines Ultraschallsensors ermöglicht werden. Dabei wird vorausgesetzt, dass die Rechtmäßigkeit des Öffnungswunsches bereits im Vorfeld beim Betreten des Lagerraums sichergestellt worden ist. Nähert sich eine Person oder ein Fahrzeug dem Tor von innen bis auf 20 cm, so soll das Tor den Öffnungsvorgang automatisch einleiten. Erneut sollen die LEDs während der Bewegung des Tores blinken. Der Zugang zum Lager von außen soll durch Einbindung einer Farberkennung mithilfe der USB-Kamera begrenzt werden. Nähert sich ein grünes Objekt, soll sich das Tor automatisch öffnen, bei einer anderen Farbe soll es entweder verschlossen bleiben oder sich schließen.

Eine erweiterte Zugangskontrolle soll über die Programmierung eines einfachen Codeschlosses realisiert werden. Dazu werden 7 Buttons auf dem Display des TXT 4.0 Controllers platziert. Sechs davon dienen der Eingabe des Codes, ein weiterer der Übermittlung des Codes sowie der Einleitung des Öffnungsvorgangs für den Fall, dass die Codeeingabe korrekt war. Das Codeschloss ist ein einfaches Codeschloss, wobei die Reihenfolge der Eingaben oder Mehrfacheingaben unberücksichtigt bleiben. Es soll lediglich überprüft werden, ob Tasten gedrückt wurden. Jedoch bleibt es den SuS freigestellt, eigene komplexere Codeschlösser zum Schutz des Lagerraums zu programmieren.



○ ALLTAGSBEZUG

Insbesondere die Einbettung in einen realitätsnahen Anwendungskontext weist eine hohe Motivationalität auf. Eine Integration der Thematik in die vorberufliche Orientierung könnte im Hinblick auf mechatronische oder informationstechnische Berufsfelder er-

folgen. Hier wird das automatisierte Öffnen und Schließen in vielen Bereichen genutzt. In besonderer Weise wird die automatisierte Öffnung von Türen oder auch Rollläden nicht nur in der Industrie, sondern auch im häuslichen Umfeld verstärkt genutzt.

Informatik:	Fortgeschrittene Programmierung, Bedingungsschleifen, Funktionen, Codierung, grafische Anzeigen
Physik:	Lichtschanke, Messung von Strecken mit dem Ultraschallsensor
Technik:	Stabiles Bauen, Steuerung von Aktoren, Auslesen von Sensoren, Konstruktionstechnik
Mathematik:	Berechnung von Termen

Einführungsphase



Unterrichtsgespräch

- Bekanntgeben des Themas, ggf. Video der automatisierten Öffnung eines/einer Schiebetors/Schiebetür (evtl. mit Codeschloss):
→ www.youtube.com/watch?v=CIFLFhAPMqI
- Abfragen, was die wesentlichen Anforderungen an die automatische Öffnung oder Schließung eines Tor- oder Türmodells sind.
- Szenarien abfragen, in denen automatisch öffnende und schließende Tor-/Türsysteme eingesetzt werden.
- Einsatzmöglichkeiten der gesammelten Szenarien diskutieren (z. B. Garagentor, Hoftor, Zugangstor, Schiebetür, Rollladen).
- Anforderungen an den Versuchsaufbau ermitteln.



ggf. Hilfestellung

- Sensoren, Aktoren und Bauteile aus dem Baukasten zeigen, wenn nötig, Präsentationsmedien einsetzen.

Planungsphase



Unterrichtsgespräch

- Die Vorgehensweise zum Bau des Modells und die zu erzielende Funktion werden gemeinsam erarbeitet.
- Die Arbeitsschritte der App werden vorgegeben bzw. besprochen.



Partner- oder Einzelarbeit

- Die SuS machen sich mit der App vertraut und laden die entsprechende Aufgabe.
- Die SuS definieren sinnvolle Funktionen der Steuerung mit dem TXT 4.0 Controller.
- Die SuS erstellen mithilfe der App die Anforderungsliste für das zu bauende Tormodell.



Optional:

Partner- oder Gruppenarbeit

- Die SuS skizzieren die möglichen Aufbauten des Tormodells selbst und konstruieren es frei.
- Die SuS diskutieren die Ergebnisse in der Gruppe und legen sich auf ein Design fest.

Konstruktionsphase



Partner- oder Einzelarbeit

- Die SuS nutzen die App zum Bau des Tormodells. Die App führt kleinschrittig durchs Programm.

Programmierphase



Partner- oder Einzelarbeit

- Die SuS schreiben die Programme für die mit dem Versuchsaufbau durchzuführenden Steuerungsaufgaben. Die App führt hier in aufeinander aufbauenden Stufen durch die Programmieraufgabe
- Hilfe wird in der App angeboten.
- Das Programm wird nach jedem Differenzierungsschritt auf den TXT 4.0 Controller übertragen.

Experimentier- und Testphase



Partner- oder Gruppenarbeit

- Der Versuchsaufbau wird in Betrieb genommen.
- Mögliche Störungen im Funktionsablauf müssen gefunden und eliminiert werden.
- Eventuelle Fehlersuche ist mithilfe von Vorschlägen in der App möglich.
- Eventuelle Optimierungen bei der Hardware und der Programmierung können vorgenommen werden.

Abschlussphase



Optional:

Vorstellung und Zuteilung der Differenzierung

- Für die Differenzierung infrage kommende SuS werden ggf. durch die Lehrperson angesprochen. Dabei wird die Möglichkeit der Programmierung eines erweiterten Codeschlusses vorgestellt.



Diskussion im Plenum

- Nachbesprechung des Projekts im Klassenverbund
- Klärung von zukünftigen Einsatzmöglichkeiten im Alltag

METHODISCH-DIDAKTISCHE HINWEISE

Differenzierungsmöglichkeiten

Je nach Dauer der Unterrichtsreihe und der Stärke der SuS können/kann

- das einleitende Szenario der nicht Endtaster gestützten und manuell überprüften Öffnung und Schließung des Tormodells entfallen.
- der Versuchsaufbau verändert und Sensoren an anderen Stellen eingebaut/verwendet werden.
- das Codeschloss verbessert werden.
- sich ein Wettbewerb „Wer knackt den Code?“ zur besten Programmierung eines Codeschlusses anschließen (Umbau des Tormodells zu einem Safe, den es entweder zu knacken oder zu schützen gilt).
- weitere Anwendungsszenarien seitens der SuS formuliert und samt Aufbau und Programmierung realisiert werden.

Motivationale Aspekte

Das Thema automatische Steuerung von Bewegung ist allen SuS aus dem Alltag bekannt. In vielen Haushalten gehören automatisiert öffnende Rolläden, Markisen oder Garagentore neben vielen weiteren smarten Anwendungen längst zum Alltag. Elektronische, Code basierte Türschlösser finden immer häufiger Anwendung bei der Zugangssteuerung von Gebäuden.

PROGRAMMIERKENNTNISSE

- Programmstart
- Einbindung von Sensoren
- Einbindung von Aktoren
- Einbindung des Displays des TXT 4.0 Controllers
- Einbinden von Eingaben über ein Touch-Display
- Kamerakonfiguration
- Dauerschleife **wiederhole dauerhaft (repeat forever)**
- Bedingung **falls – mache (if – do)**
- Befehl **warte (wait)**
- Befehl **warte bis (wait until)**
- Arbeit mit Unterprogrammen
- Verwendung von Events und Threats

ZUSATZMATERIALIEN

Zum Download optional:

- Stromlaufplan
- Bauanleitung

- Falls vorhanden, könnte für die Einführungsphase in das Thema ein Video genutzt werden.
- Falls vorhanden, können Materialien zum Programmieren von Codeschlössern verwendet werden.
- Zeichenmedien (Papier, Whiteboard oder Projektionsfläche) einsetzen.

—○ FUNKTIONEN DES MODELLS UND DEREN TECHNISCHE LÖSUNGEN

Funktion der Sensoren/Aktoren	Technische Lösung
Lichtsignale rote LEDs	Ansteuern der LEDs bei einer Torbewegung
Unterbrechung der Lichtschranke	Unterbinden der Torbewegung
Ultraschallsensor	Annäherungsdetektion
Differenzierung	Entwurf von verschiedenen komplexen Torsicherungen oder Torsteuerungen (Codeschloss, Kombination von Sensoreingaben)

—○ MATERIALLISTE

Sensoren	Funktion
1 On/Off-Taster am TXT 4.0 Controller	1. Einschalten der Schrankensteuerung 2. Not-Stopp der Schranke
1 Ultraschallsensor	Annäherungserkennung
1 USB-Kamera	Farberkennung
1 Fototransistor	Unterbrechungserkennung Lichtschranke, Empfänger
2 Mini-Taster	Endabschalter zur Torsteuerung

Aktoren	Funktion
1 Display des TXT 4.0 Controllers	Anzeige
2 LEDs (2 × rot)	Warnung
1 Encodermotor	Bewegung des Tores
1 LED (weiß)	Beleuchtung Lichtschrankensensor

Parkhaus 1-3

Ist das Parkhaus voll?



LEITFRAGEN:

- Wo ist eine automatische Schrankenanlage im Alltag einsetzbar? (*Kommunikation*)
- Welche Funktionen muss die Steuerung der Parkhausschranke sinnvollerweise erfüllen? (*Kollaboration*)
- Unter welchen Bedingungen soll das System öffnen bzw. schließen? (*kritisches Denken*)
- Was ist zu berücksichtigen, damit die Steuerung an verschiedenen Standorten genutzt werden kann und das System möglichst robust funktioniert? (*Kreativität*)

DIE UNTERRICHTSIDEE AUF EINEN BLICK

Klassenstufe: 11–13

Zeitaufwand: 2 Doppelstunden pro Lerneinheit (erweiterbar bis zu 10 DS)

Schwierigkeitsgrad: Modell  bis 
Programmierung  bis 

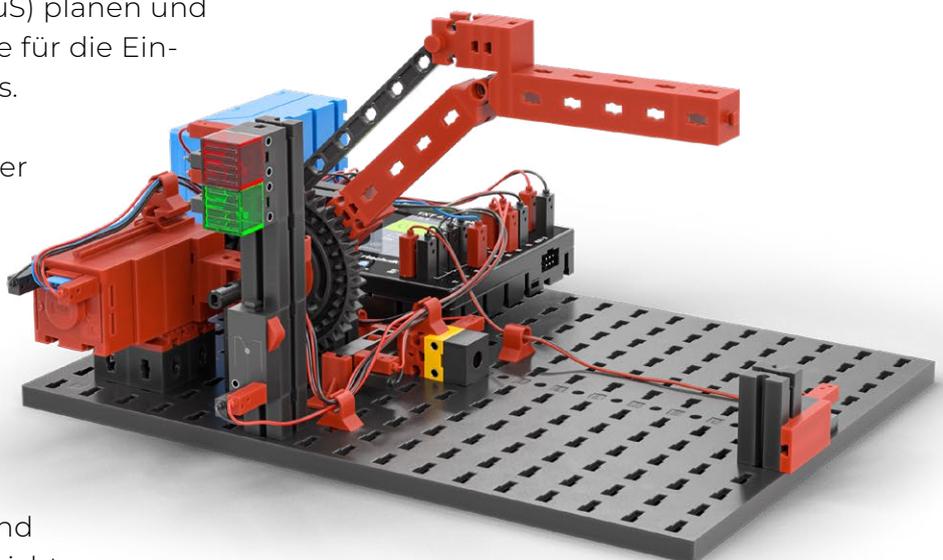
Modellart: Tischmodell für Parkhausschranken

MODELLBESCHREIBUNG / AUFGABE

Die Schülerinnen und Schüler (SuS) planen und realisieren eine Parkhausschranke für die Ein- und Ausfahrt aus einem Parkhaus.

Die Anforderung zum Öffnen der Schranke erfolgt zunächst über einen Taster (Einfahrt) bzw. einen Button auf dem Display des TXT 4.0 Controllers (Ausfahrt).

Zwei LEDs geben Lichtsignale bei vollständiger Öffnung (grün) bzw. während des Öffnungs- und Schließvorgangs (rot). Über eine Lichtschranke wird überwacht, ob sich beim



Parkhausschranke

Schließvorgang ein Fahrzeug im Schrankenbereich befindet. Aus der Anzahl der Ein- und Ausfahrten wird die Anzahl der freien Stellplätze ermittelt und auf dem Display des TXT 4.0 Controllers ausgegeben, bei voller Belegung blinkt die rote LED (Parkhaus 1).

In einem zweiten Schritt schließen sich zwei Gruppen zusammen und verbinden ihre Modelle und Steuerungen so, dass eine Schranke zur Einfahrt, die zweite zur Ausfahrt dient (Parkhaus 2).

Im voll ausgebauten Modell wird die Schrankensteuerung um eine optische Kennzeichenerkennung mithilfe der USB-Kamera und eine automatische Parkpreisermittlung ergänzt (Parkhaus 3).

○ ALLTAGSBEZUG

Die schrankengesteuerte Zufahrt zu Parkplätzen und -häusern ist den SuS aus dem Alltag bekannt.

Insbesondere die automatische Zufahrt durch optische Kennzeichenerkennung weist einen hohen motivationalen Wert auf.

Eine Integration der Thematik in die vorberufliche Orientierung könnte im Hinblick auf informationstechnische Berufsfelder erfolgen. Hier wird das automatisierte Schalten durch das Erfassen physikalischer Größen in vielen Bereichen genutzt. In besonderer Weise wird die automatisierte Erkennung von Objekten in der Zugangstechnik zunehmend wichtiger.

○ FÄCHERBEZUG

Informatik:	Fortgeschrittene Programmierung, Bedingungsschleifen, Funktionen
Physik:	Bewegungsänderung
Technik:	Stabiles Bauen, Konstruktionstechnik
Mathematik:	Winkel, bedingtes Zählen, Zeit- und Preisberechnung

○ UNTERRICHTSVERLAUF

Einführungsphase



Unterrichtsgespräch

- Bekanntgeben des Themas, ggf. Video „Wie die Fahrt ins Parkhaus vereinfacht werden soll“ zeigen:
→ www.regio-tv.de/mediathek/video/wie-die-fahrt-ins-parkhaus-vereinfacht-werden-soll
- Abfragen, was die wesentlichen Merkmale der Schrankensteuerung sind.



ggf. Hilfestellung

- Sensoren, Aktoren und Bauteile aus dem Baukasten zeigen, wenn nötig, Präsentationsmedien einsetzen.

Planungsphase



Unterrichtsgespräch

- Die Vorgehensweise zum Bau des Modells und die zu erzielenden Funktionen werden gemeinsam erarbeitet.
- Die Arbeitsschritte der App werden vorgegeben bzw. besprochen.



Partner- oder Einzelarbeit

- Die SuS machen sich mit der App vertraut und laden die entsprechende Aufgabe.
- Die SuS definieren sinnvolle Funktionen einer automatisch öffnenden und schließenden Schranke.
- Die SuS erstellen mithilfe der App die Anforderungsliste für die zu bauende Schrankenanlage.



Optional:
Partner- oder Gruppenarbeit

- Optional skizzieren die SuS die möglichen Aufbauten der Schranke mit weiterer Sensorik (Lichtschranke) und Aktorik (LED).
- Die SuS diskutieren die Ergebnisse in der Gruppe und legen sich auf ein Design fest.

Konstruktionsphase



Partner- oder Einzelarbeit

- Die SuS nutzen die App zum Bau der Schranke. Die App führt kleinschrittig durchs Programm.

Programmierphase der Parkhausschranke (Parkhaus 1)



Partner- oder Einzelarbeit

- Die SuS entwickeln das Programm für die Schrankenanlage. Die App führt hier in aufeinander aufbauenden Stufen mit offenen Fragestellungen durch die Programmieraufgabe.
- Hilfe wird in der App angeboten.
- Das Programm wird nach jedem Differenzierungsschritt auf den TXT 4.0 Controller übertragen.

Experimentier- und Testphase der Parkhausschranke (Parkhaus 1)



Partner- oder Einzelarbeit

- Die Parkhausschranke wird in Betrieb genommen.
- Mögliche Störungen im Funktionsablauf müssen gefunden und eliminiert werden.
- Eventuelle Fehlersuche ist anhand von Vorschlägen in der App möglich.
- Eventuelle Optimierungen bei der Hardware und der Programmierung können vorgenommen werden.

Abschluss-/Anschlussphase der Parkhausschranke (Parkhaus 1)



Optional:
Vorstellung
und Zuteilung
der Differen-
zierungen

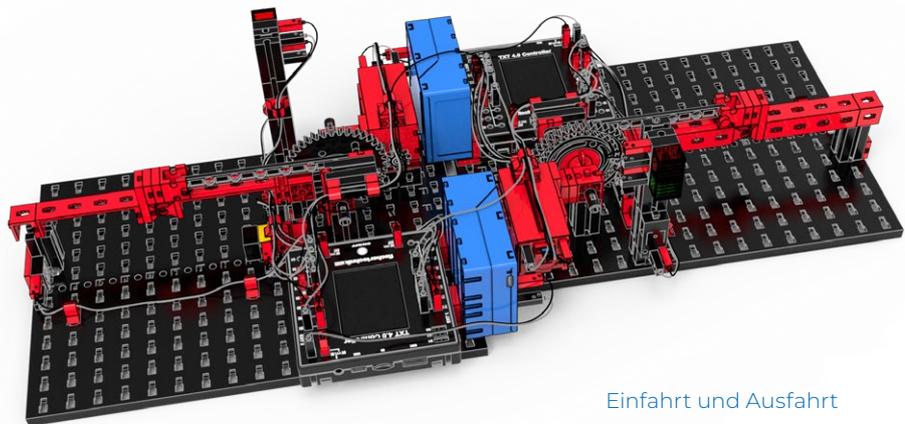
- Für die Differenzierung infrage kommende SuS (oder die gesamte Gruppe) werden ggf. durch die Lehrperson angesprochen. Dabei wird die Kombination zweier Schranken zur getrennten Steuerung von Einfahrt und Ausfahrt in Gruppenarbeit vorgestellt.
- Als weitere Differenzierung (oder direkt nach dem ersten Programm ohne anschließende Gruppenarbeit) werden die USB-Kamera und das Erkennen und Auswerten von Kennzeichen in der App angeboten. Die App bietet konkrete Ideen für z. B. eine Zeiterfassung und Berechnung von Parkgebühren.

Programmierphase der Einfahrt und Ausfahrt (Parkhaus 2)



Partner- oder
Einzelarbeit

- Die Gruppen entscheiden, welches Schrankenmodell für die Einfahrt und welches für die Ausfahrt zuständig ist.
- Kopplung der beiden TXT 4.0 Controller lt. Lernkarten
- Arbeitsteilige Anpassung der Programme für Ein- und Ausfahrt getrennt (ggf. aber auch gemeinsame Arbeit beider Gruppen)



Einfahrt und Ausfahrt

Experimentier- und Testphase der Einfahrt und Ausfahrt (Parkhaus 2)



Partner- oder
Gruppenarbeit

- Inbetriebnahme der Anlage
- Überprüfung der korrekten Kommunikation der beiden TXT 4.0 Controller
- Ggf. Fehlersuche und Behebung

Abschluss-/Anschlussphase der Einfahrt und Ausfahrt (Parkhaus 2)



Optional:
Vorstellung
und Zuteilung
weiterer Diffe-
renzierungen

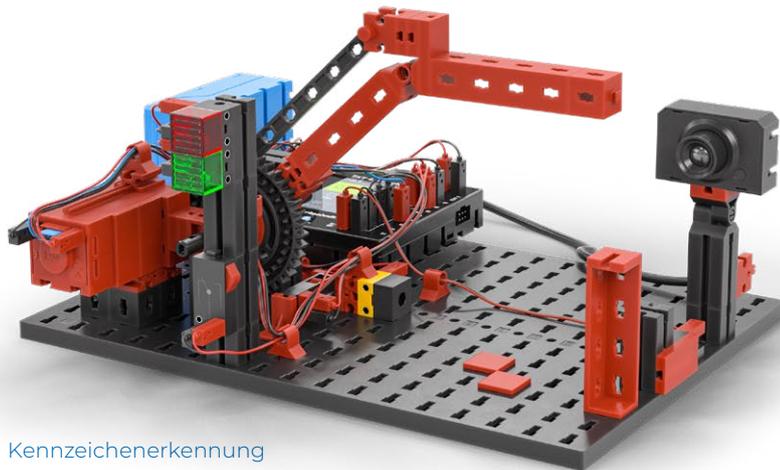
- Nun werden die USB-Kamera und das Erkennen und Auswerten von Kennzeichen in der App angeboten. Die App bietet konkrete Ideen für z. B. eine Zeiterfassung und Berechnung von Parkgebühren.

Programmierphase der Kennzeichenerkennung (Parkhaus 3)



Partner- oder Einzelarbeit

- Falls notwendig, Rückbau der Verknüpfung zweier Schranken und Weiterarbeit in den einzelnen Gruppen
- Anschließen der USB-Kamera
- Umgang mit dem Strichcode und dessen Decodierung sowie Speicherung in einem neuen Programm
- Laden des Programms aus Parkhaus 1
- Gestaltung des Displays des TXT 4.0 Controllers lt. Lernkarten
- Schrittweise Erweiterung des Programms: Erfassung und Speicherung einfahrender Kennzeichen, Berechnung der Parkdauer und der Gebühren ausfahrender Fahrzeuge
- Simulation des Bezahlvorgangs und Öffnen der Schranke, jeweils Berechnung und Anzeige freier Stellplätze



Kennzeichenerkennung

Experimentier- und Testphase der Kennzeichenerkennung (Parkhaus 3)



Partner- oder Gruppenarbeit

- Inbetriebnahme der Anlage
- Testen und ggf. Fehlerbehebung der einzelnen Programmierschritte
- Testen des Gesamtprogramms
- Ggf. Fehlersuche und Behebung

Abschlussphase der Kennzeichenerkennung (Parkhaus 3)



Optional:
Vorstellung und Zuteilung weiterer Differenzierungen

- Besonders effizient arbeitenden Gruppen kann die Erweiterungsaufgabe gestellt werden, die einfahrenden Elektro- und Normalautos in getrennten Tabellen zu speichern und z. B. 4 Parkplätze für E-Autos und 6 Parkplätze für N-Autos vorzuhalten. Auch die Parkgebühren können differenziert berechnet werden. Hier ist selbstständiges Arbeiten ohne Lernkarten vorgesehen.



Diskussion im Plenum

- Nachbesprechung des Projekts im Klassenverbund
- Klärung von zukünftigen Einsatzmöglichkeiten im Alltag

METHODISCH-DIDAKTISCHE HINWEISE

Differenzierungsmöglichkeiten

Je nach Dauer der Unterrichtsreihe und der Stärke der SuS können

- zwei Gruppen ihre Anlagen als Einfahrt bzw. Ausfahrt eines Parkhauses zusammenstellen,
- sie zwei TXT 4.0 Controller (Einfahrt/ Ausfahrt) koppeln, um Zählergebnisse zu synchronisieren,
- sie in Parkhaus 3 mithilfe der USB-Kamera die Kennzeichen zum Zugang sowie zur Ausfahrt die Parkzeit ermitteln und auf Basis der ermittelten Daten Parkgebühren auf dem Display anzeigen lassen.

Motivationale Aspekte

Das Thema automatische Schrankenanlage ist allen SuS aus dem Alltag bekannt. Eine automatische Kennzeichenerkennung, die ggf. auf weitere Automatisierungsprozesse übertragbar ist, motiviert zu intensiver Auseinandersetzung und Lösung einer entsprechenden Aufgabe.



PROGRAMMIERKENNTNISSE

- Programmstart
- Dauerschleife **wiederhole dauerhaft (repeat forever)**
- Einbindung von Sensoren und Aktoren
- Programmierung von Displayanzeigen
- Programmierung von Touch-Buttons
- Kamerakonfiguration
- Schleife **wiederhole bis (repeat until)**
- Befehle **warte (wait)** und **warte bis (wait until)**
- Zählschleifen **zähle von ... bis ... (count with – from ... to ... – by)**
- Nutzen von Variablen und deren Veränderung
- Arbeit mit Unterprogrammen
- Arbeit mit Events und Threads

Zum Download optional:

- Stromlaufplan
- Bauanleitung

ZUSATZMATERIALIEN

- Falls vorhanden, könnte für die Einführungsphase in das Thema ein Video genutzt werden.
- Zeichenmedien (Papier, Whiteboard oder Projektionsfläche).

—○ FUNKTIONEN DES MODELLS UND DEREN TECHNISCHE LÖSUNGEN

Funktion der Sensoren/Aktoren	Technische Lösung
Ausführen einer Drehung eines Encodermotors	Ansteuern des Antriebsmotors für genau eine Drehung
Lichtsignale rote/grüne LED	Ansteuern der LED für freie Fahrt und Stopp
Unterbrechung der Lichtschranke	Unterbinden der Schrankenbewegung
Unterbrechung der Lichtschranke	Zählen der Autos
Kennzeichenerkennung durch USB-Kamera	Öffnen der Schranke
Parkhaus 2: Verbindung zweier TXT 4.0 Controller	<ul style="list-style-type: none"> • Getrennte Ansteuerung von Ein- und Ausfahrtschranke • Datenaustausch zwischen zwei TXT 4.0 Controllern
Parkhaus 3: Kennzeichenerkennung durch USB-Kamera	<ul style="list-style-type: none"> • Strichcode-Decodierung • Programmierung und Auswertung von Start- und Endzeiten • Berechnung von Parkgebühren

—○ MATERIALLISTE

Sensoren	Funktion
1 On/Off-Taster am TXT 4.0 Controller	<ol style="list-style-type: none"> 1. Einschalten der Schrankensteuerung 2. Not-Stopp der Schranke
1 Fototransistor	Unterbrechungserkennung Lichtschranke, Empfänger
1 USB-Kamera	Strichcodeerkennung
1 Mini-Taster	Bedarfsanforderung zur Einfahrt
1 Mini-Taster	Statusabfrage der Schranke (offen oder geschlossen)

Aktoren	Funktion
1 Encodermotor	<ul style="list-style-type: none"> • Bewegung • Zählen der Umdrehungen
2 LEDs (1 × rot, 1 × grün)	Zustandsanzeige
1 LED (weiß)	Beleuchtung Lichtschrankensensor

Ampel- steuerung 1-3

Fließt der Verkehr?



LEITFRAGEN:

- Welche Arten der Steuerung von Verkehrsströmen gibt es? *(Kommunikation)*
- Wie kann man die Verkehrsströme von motorisiertem Verkehr und Fußgängern sicher lenken? *(Kollaboration)*
- Welche Wartezeiten sind den einzelnen Verkehrsteilnehmern zumutbar? *(kritisches Denken)*
- Wie kann man Lichtsignale zeit- und anforderungsgesteuert sicher betreiben? *(Kreativität)*

○ DIE UNTERRICHTSIDE E AUF EINEN BLICK

Klassenstufe:	11–13
Zeitaufwand:	2–3 Doppelstunden pro Lerneinheit (erweiterbar bis zu 10 DS)
Schwierigkeitsgrad:	Modell  Programmierung  bis 
Modellart:	Tischmodell für Lichtsignalanlage

○ MODELLBESCHREIBUNG / AUFGABE



Lichtsignalanlage

Die Schülerinnen und Schüler (SuS) planen und realisieren eine Lichtsignalanlage zur Steuerung von Personen- und Kraftfahrzeugverkehr. Ausgehend von einer einfachen zeitgesteuerten Fußgängerampel, über eine Querungsampel für eine Kraftfahrtstraße bis hin zu einer Kreuzung mit Kraftfahrzeug- und Fußgängerströmen sowie der KI-Phasensteuerung einer Querungsampel entwickelt sich das Modell fort und damit auch die Ansprüche an Aufbau und Programmierung.

Die SuS lernen Signallagepläne und Signalzeitenpläne kennen und lesen. Aus den Informationen der Signalzeitenpläne ermitteln sie mögliche Schaltgruppen, die gleichzeitig geschaltet werden können, sowie die zugehörigen Schaltzeiten.

Aus den so festgelegten Gruppen und Zeiten entwickeln die SuS ein Steuerprogramm, das anhand des Ampelmodells auf seine Tauglichkeit getestet werden kann.

○ ALLTAGSBEZUG

Die zeitgesteuerte Aktivierung von Aktoren ist den SuS aus dem technikwissenschaftlichen Unterricht oder Informatikunterricht bzw. ihrem Alltag bekannt.

Insbesondere die Einbettung in einen realitätsnahen Anwendungskontext, den die SuS aus ihrem Alltag sehr gut kennen, weist eine hohe Motivationalität auf.

Eine Integration der Thematik in die vorberufliche Orientierung könnte im Hinblick auf verkehrstechnische oder elektrotechnische Berufsfelder erfolgen, wo das automatisierte Ansteuern von Aktoren in vielen Bereichen genutzt wird. In besonderer Weise kommt die automatisierte Zeitsteuerung nicht nur in der Industrie, sondern auch im häuslichen Umfeld, z. B. bei Smart Home Anwendungen, zum Einsatz.

○ FÄCHERBEZUG

Informatik:	Fortgeschrittene Programmierung, Bedingungsschleifen, Funktionen, grafische Anzeigen, neuronale Netze, KI
Physik:	Stromkreis, Bewegung, Messungen von Zeiten
Technik:	Stabiles Bauen, Signalzeitenpläne, Konstruktionstechnik
Mathematik:	Berechnung von Termen

○ UNTERRICHTSVERLAUF

Einführungsphase



Unterrichtsgespräch

- Bekanntgeben des Themas, ggf. Video „Simulation einer Fußgängerampel“ zeigen:
→ www.youtube.be/a3Be5vOOUzs
- Abfragen, was die wesentlichen Merkmale für die Steuerung von Verkehrsströmen sind.
- Szenarien abfragen, in denen es zu Querungsproblemen kommen kann.
- Einsatzmöglichkeiten der gesammelten Szenarien diskutieren (z. B. Fußgängerampel, Fahrzeugampel, andere Lichtsignale).
- Anforderungen an den Aufbau der Modelle ermitteln.



ggf. Hilfe-
stellung

- Sensoren, Aktoren und Bauteile aus dem Baukasten zeigen, wenn nötig, Präsentationsmedien einsetzen.

Planungsphase



Unterrichts-
gespräch

- Die Vorgehensweise zum Bau des Modells und die zu erzielenden Funktionen werden gemeinsam erarbeitet.
- Die Arbeitsschritte der App werden vorgegeben bzw. besprochen.



Partner- oder
Einzelarbeit

- Die SuS machen sich mit der App vertraut und laden die entsprechende Aufgabe.
- Die SuS definieren sinnvolle Funktionen zur Steuerung einer Lichtsignalanlage (LSA) mithilfe des TXT 4.0 Controllers.
- Die SuS erstellen mithilfe der App die Anforderungsliste/Schaltzeitentabelle für den aktuellen Anwendungsfall.



Optional:
Partner- oder
Gruppenarbeit

- Die SuS messen die Signalzeiten einer schulnah gelegenen Fußgängerampel und erstellen einen angepassten Signalzeitenplan.
- Die SuS diskutieren die Ergebnisse in der Gruppe und arbeiten die Unterschiede zur vorgegebenen LSA heraus.

Konstruktionsphase der Lichtsignalanlage (Ampelsteuerung 1)



Partner- oder
Einzelarbeit

- Die SuS nutzen die App zum Bau der LSA. Die App führt kleinschrittig durchs Programm.

Programmierphase der Lichtsignalanlage (Ampelsteuerung 1)



Partner- oder
Einzelarbeit

- Die SuS schreiben das Programm für die Ansteuerung der LSA. Die App führt hier in aufeinander aufbauenden Stufen durch die Programmieraufgabe.
- Hilfe wird in der App angeboten.
- Entnehmen von wichtigen Informationen zur Ansteuerung der Lichtsignalanlage aus einem Signallage- und einem Signalzeitenplan sowie Festlegen der Schaltzeiten und der aufeinanderfolgenden Schaltzustände
- Clustern von Signalgruppen zu Schaltphasen
- Programmierung einer einfachen zeitgesteuerten Lichtsignalanlage für den Fahrzeug- und Fußgängerverkehr
- Das Programm wird nach jedem Differenzierungsschritt auf den TXT 4.0 Controller übertragen.

Experimentier- und Testphase der Lichtsignalanlage (Ampelsteuerung 1)



Partner- oder Gruppenarbeit

- Das Ampelmodell wird in Betrieb genommen.
- Mögliche Störungen im Funktionsablauf müssen gefunden und eliminiert werden.
- Eventuelle Fehlersuche ist mithilfe von Vorschlägen in der App möglich.
- Einbau einer Anforderungssteuerung für die Fußgängerampel
- Weiterentwicklung des Programms zu einer Tag- und Nachtschaltung mit Anforderungssteuerung der Fußgängerampel
- Eventuelle Optimierungen bei Hardware und Programmierung können vorgenommen werden.

Abschluss-/Anschlussphase der Lichtsignalanlage (Ampelsteuerung 1)



Optional:
Vorstellung und Zuteilung der Differenzierungen

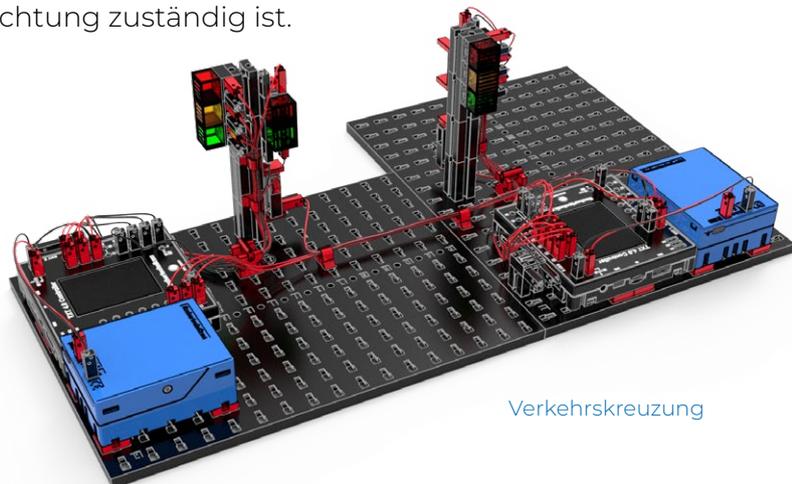
- Für die Differenzierung infrage kommende SuS (oder die gesamte Gruppe) werden ggf. durch die Lehrperson angesprochen. Dabei kann das Wartezeitmodell vorgestellt werden.

Konstruktionsphase der Verkehrskreuzung (Ampelsteuerung 2)



Partner- oder Gruppenarbeit

- Konstruktion der Teilkreuzung aus zwei Ampelmodellen
- Die Gruppen entscheiden, welches Ampelmodell für welche Richtung zuständig ist.



Verkehrskreuzung

Programmierphase der Verkehrskreuzung (Ampelsteuerung 2)



Partner- oder Gruppenarbeit

- Kopplung der beiden TXT 4.0 Controller lt. Lernkarten
- Arbeitsteilige Anpassung der Programme für die Erstellung der komplexen Ampelsteuerung einer Verkehrskreuzung (ggf. auch gemeinsame Arbeit beider Gruppen)
- Programmierung der beiden TXT 4.0 Controller als Master- und Slave-Controller

Experimentier- und Testphase der Verkehrskreuzung (Ampelsteuerung 2)



Partner- oder Gruppenarbeit

- Die Teilkreuzung wird in Betrieb genommen.
- Mögliche Störungen bei der Kopplung und im Funktionsablauf müssen gefunden und eliminiert werden.
- Eventuelle Fehlersuche ist mithilfe von Vorschlägen in der App möglich.
- Die Weiterentwicklung des Programms zu einer komplexeren Ampelsteuerung ist möglich.
- Eventuelle Optimierungen bei Hardware und Programmierung können vorgenommen werden.

Abschluss-/Anschlussphase der Verkehrskreuzung (Ampelsteuerung 2)



Optional:
Vorstellung und Zuteilung weiterer Differenzierungen

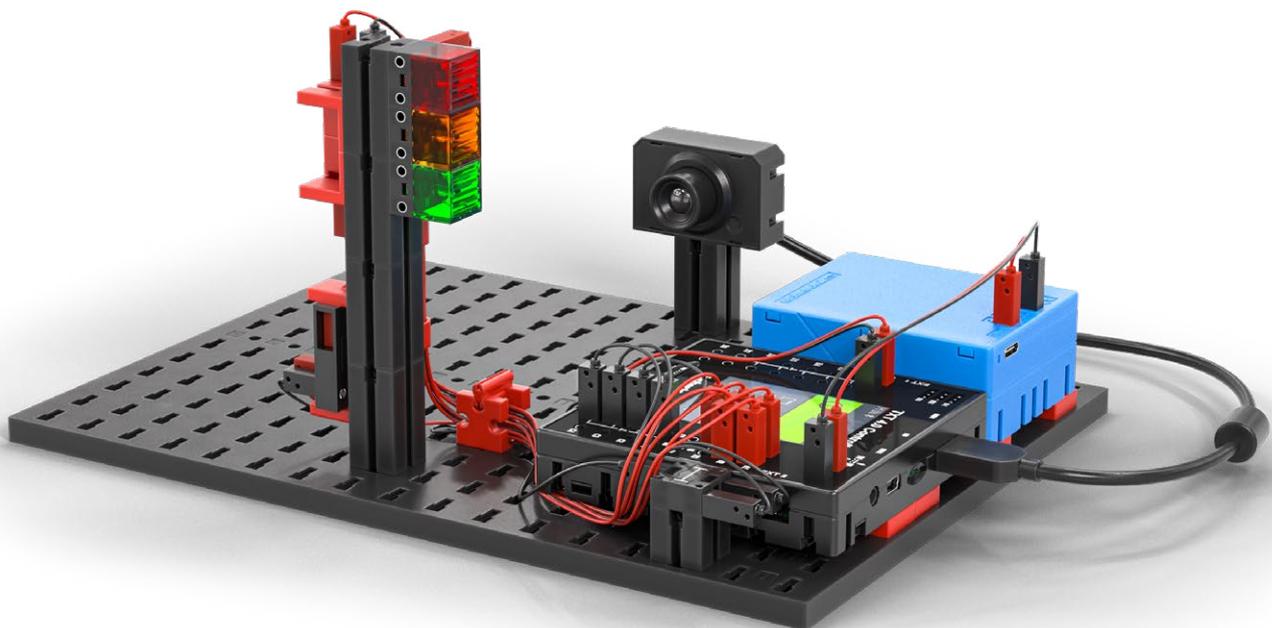
- Für die Differenzierung infrage kommende SuS (oder die gesamte Gruppe) werden ggf. durch die Lehrperson angesprochen. Dabei sollen die Konzepte der neuronalen Netze und der KI vorgestellt werden.

Konstruktionsphase der KI-Phasensteuerung (Ampelsteuerung 3)



Partner- oder Einzelarbeit

- Einbau der USB-Kamera zur Objektkategorisierung in die einzelne Lichtsignalanlage (Ampelsteuerung 1)



Lichtsignalanlage
mit USB-Kamera

Programmierphase der KI-Phasensteuerung (Ampelsteuerung 3)



Partner- oder Einzelarbeit

- Konfiguration der USB-Kamera
- Erfassen von Trainingsdaten für das KI-Modell
- Programmierung der KI-gestützten Farberkennung
- Anpassung der Schaltzeiten der Fußgängerampel hinsichtlich des erkannten Querungsteilnehmers (Rollstuhlfahrer oder Fußgänger) und der zugehörigen Querungszeiten

Experimentier- und Testphase der KI-Phasensteuerung (Ampelsteuerung 3)



Partner- oder Gruppenarbeit

- Testen der Programmierung mithilfe farbiger Blöcke
- Testen der Programme mithilfe simulierter Verkehrsteilnehmer (Rollstuhlfahrer, Fußgänger)

Abschlussphase der KI-Phasensteuerung (Ampelsteuerung 3)



Optional:
Vorstellung und Zuteilung weiterer Differenzierungen

- Evtl. Konstruktion einer Vollkreuzung durch den Einsatz von mehreren Konstruktionskästen (nur ohne KI möglich)



Diskussion im Plenum

- Nachbesprechung des Projekts im Klassenverbund
- Klärung von zukünftigen Einsatzmöglichkeiten im Alltag

METHODISCH-DIDAKTISCHE HINWEISE

Differenzierungsmöglichkeiten

Je nach Dauer der Unterrichtsreihe und der Stärke der SuS kann

- die Komplexität der Anforderungen an die Steuerung der LSA nicht nur über die Anzahl der Lichtsignale, sondern auch durch die Anzahl der sicher zu querenden Verkehrsströme gesteigert werden.
- eine reale Messung der Signalzeiten einer LSA im schulnahen Umfeld erfolgen, die die Grundlage für eine geänderte Steuerung der LSA bilden könnte.
- das Ampelmodell durch eine Gruppenarbeit erweitert werden.
- mithilfe der USB-Kamera eine Erkennung von Verkehrsteilnehmern und daraus folgend eine Anpassung der Phasenlängen aufgrund von geänderten Räumzeiten realisiert werden.

Motivationale Aspekte

Das Thema automatische Steuerung von Aktoren ist allen SuS aus dem Alltag bekannt. In vielen Haushalten gehören etliche smarte Anwendungen längst zum Alltag. Sich zeitgesteuert autonom einschaltende Wasch- oder Spülmaschinen, zentral angesteuerte Rollläden oder teilautonom kochende Küchenmaschinen finden immer häufiger Anwendung.



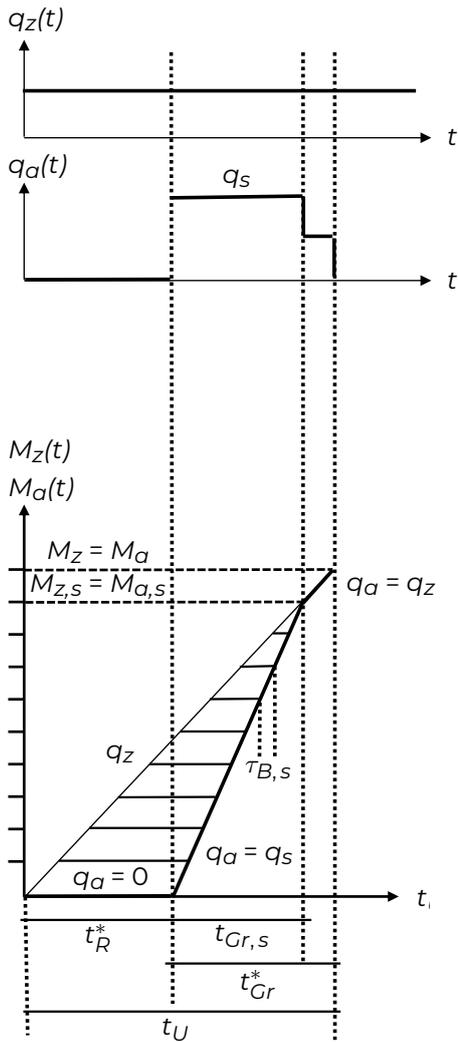
GRUNDLAGEN VERKEHRSTECHNIK

Die Nutzung von verkehrstechnischen Einrichtungen zur Steuerung von Verkehrsströmen ist sowohl den SuS als auch den LuL geläufig.

Die Dimensionierung der LSA stellt eine zentrale Aufgabe in der Verkehrstechnik dar, um den Verkehrsfluss effizient zu steuern und die Verkehrssicherheit zu erhöhen. Dabei sind mehrere Faktoren zu berücksichtigen, die sowohl verkehrstechnische als auch technische Aspekte umfassen.

Zunächst erfolgt eine umfassende Verkehrserhebung, bei der Verkehrsaufkommen, Fahrzeugtypen, Fußgängerströme sowie Querschnitt und Verkehrsarten erfasst werden. Eine solche Erhebung könnte im Rahmen des Unterrichtes an einer schulnah gelegenen Lichtsignal gesteuerten Kreuzungsanlage mit dem Ziel der Erstellung von lokalen Signallageplänen und Signalzeitenplänen durchgeführt werden. Auf Grundlage dieser Daten werden die Phasenzeiten der Lichtsignalanlage bestimmt, die das Verhältnis von Grün-, Gelb- und Rotphasen regeln.

Die Bemessung der Grünzeiten orientiert sich an den erforderlichen Durchflussmengen und den Anhaltezeiten, wobei auch die Reaktionszeiten der Verkehrsteilnehmer berücksichtigt werden. Vereinfacht kann dafür das folgende Wartezeitenmodell verwendet werden:



- q_z : Zufluss
- q_a : Abfluss
- q_s : Gesättigter Abfluss
- M_a : Anzahl abgeflossener Fahrzeuge (Fz)
- $M_{a,s}$: Anzahl abgeflossener Fahrzeuge bis Ende $t_{Gr,s}$
- M_z : Anzahl zugeflossener Fahrzeuge
- $M_{z,s}$: Anzahl zugeflossener Fahrzeuge bis Ende $t_{Gr,s}$
- $\tau_{B,s}$: Zeitbedarfswert, Brutto-Zeitlücke aufeinander folgender Fahrzeuge, wenn $q = q_s$
- t_R^* : Effektive Sperrzeit
- t_{Gr}^* : Effektive Freigabezeit
- $t_{Gr,s}$: Gründauer mit gesättigtem Abfluss
- t_U : Umlaufzeit

Dabei findet die Staubildung während der effektiven Sperrzeit t_R^* mit konstantem Zufluss q_z statt. Der Stauabbau findet während der effektiven Grünzeit t_{Gr}^* statt. Bis sich der Stau aufgelöst hat, erfolgt der Abbau mit dem gesättigten Abfluss q_s (in der Zeit $t_{Gr,s}$). Nach dem Abbau des Staus gilt Abfluss $q_a =$ Zufluss q_z .

Es gilt:

Gründauer mit gesättigtem Abfluss:
$$t_{Gr,s} = \frac{q_z}{q_s - q_z} \cdot t_R^*$$

Maximale Wartezeit:
$$w_{max} = t_r^* = t_U - t_{Gr}^*$$

Summe der Wartezeiten pro Umlauf:
$$W = M_{z,s} \cdot \frac{t_R^*}{2}$$

Maximale Wartezeit:
$$w_{max} = t_R^* = t_U - t_{Gr}^*$$

Mittlere Wartezeit:
$$\bar{w} = \frac{t_R^* + t_{Gr,s}}{t_U} \cdot \frac{t_R^*}{2}$$

Beispiel: $q_z = \frac{450 \text{ Fz}}{h}$, $t_U = 90 \text{ s}$, $t_{Gr}^* = 30 \text{ s}$ \rightarrow $t_R^* = (90 - 30) \text{ s} = 60 \text{ s}$

Gründauer mit gesättigtem Abfluss:	$t_{Gr,s} = \frac{450}{1800 - 450} \cdot 60 \text{ s} = 20 \text{ s}$
Anzahl behinderter Fahrzeuge:	$M_{z,s} = 0,5 \frac{\text{Fz}}{\text{s}} \cdot 20 \text{ s} = 10 \text{ Fz (pro Umlauf)}$
Maximale Wartezeit:	$w_{\max} = t_r^* = 60 \text{ s}$
Summe der Wartezeiten pro Umlauf:	$W = 10 \text{ Fz} \cdot \frac{60 \text{ s}}{2} = 300 \text{ s (pro Umlauf)}$
Mittlere Wartezeit:	$\bar{w} = \frac{60 + 20}{90} \cdot \frac{60}{2} \text{ s} = 26,7 \text{ s}$

Daraus folgt, dass bei der vorgegebenen Verkehrssituation mit 300 s Wartezeit pro Umlauf zu rechnen ist. Volkswirtschaftlich betrachtet, sind die dadurch entstehenden Kosten aufgrund nicht produktiv genutzter Arbeitszeit nicht unerheblich. Daraus leitet sich die Pflicht ab, Ampelschaltungen mit möglichst geringen Wartezeiten zu realisieren.

Ein weiterer wichtiger Aspekt ist die Koordination der LSA innerhalb eines vernetzten Verkehrsraums, um eine optimale Verkehrsabwicklung entlang von Hauptachsen zu gewährleisten. Hierzu werden sogenannte Grünwellen berechnet, die auf die jeweiligen Verkehrsströme abgestimmt sind.

Die technische Auslegung umfasst zudem die Wahl der Signalgeber, die Positionierung der Sensoren zur Verkehrserfassung und die Berücksichtigung von Sicherheitsnormen und gesetzlichen Vorgaben. Insgesamt zielt die Dimensionierung darauf ab, Wartezeiten zu minimieren, die Verkehrssicherheit zu verbessern und eine nachhaltige Verkehrssteuerung zu ermöglichen.

◉ PROGRAMMIERKENNTNISSE

- Programmstart
- Dauerschleife **wiederhole dauerhaft (repeat forever)**
- Einbindung von Sensoren
- Einbindung des Displays des TXT 4.0 Controllers
- Programmierung von Displayanzeigen
- Programmierung des Touch-Buttons
- Kamerakonfiguration
- Schleife **wiederhole bis (repeat until)**
- Befehl **warte (wait)**
- Nutzen von Variablen und deren Veränderung
- Arbeit mit Unterprogrammen
- Programmierung zweier TXT 4.0 Controller als Master- und Slave-Controller
- Aufbau neuronaler Netze
- Erfassung von KI-Trainingsdaten

Zum Download optional:

- Stromlaufplan
- Bauanleitung

◉ ZUSATZMATERIALIEN

- Falls vorhanden, könnte für die Einführungsphase in das Thema ein Video genutzt werden.
- Zeichenmedien (Papier, Whiteboard oder Projektionsfläche).

—○ FUNKTIONEN DES MODELLS UND DEREN TECHNISCHE LÖSUNGEN

Funktion der Sensoren/Aktoren	Technische Lösung
Lichtsignalanlage (LSA)	Zeitsteuerung der Aktoren
Bedarfsanmeldung	Anpassung der Programmierung
Querungssimulation	Entflechtung von Verkehrsströmen
Farberkennung durch USB-Kamera	Steuerung von Grünphasenlängen
Ampelsteuerung 2: Verbindung zweier TXT 4.0 Controller	Datenaustausch zwischen zwei Controllern nach dem Konzept von Master und Slave
Ampelsteuerung 3: Farberkennung durch USB-Kamera	<ul style="list-style-type: none"> • Aufbau eines neuronalen Netzes • Erfassung von Trainingsdaten • Umsetzung eines Klassifikationsproblems mithilfe von KI
Weitere Differenzierung: Anpassung der LSA an real vorhandene LSA	Datenerfassung, Modellbildung, Phasierung, Programmierung, Erprobung

—○ MATERIALLISTE

Sensoren	Funktion
1 On/Off-Taster am TXT 4.0 Controller	Einschalten der Ampelsteuerung
1 Mini-Taster	Bedarfsanmeldung
1 USB-Kamera	Kategorisierung von Verkehrsteilnehmern
Aktoren	Funktion
1 Display des TXT 4.0 Controllers	Visualisierung des Programms
5 LEDs (2 × rot, 2 × grün, 1 × gelb)	Lichtsignale der Ampel

Fahrerloses Transport- fahrzeug 1-4

Sicher unterwegs im Parcours



LEITFRAGEN:

- Welche Sensoren und Steuerungsarten eignen sich für ein fahrerloses Transportfahrzeug (FTF) im Parcours? *(Kommunikation)*
- Wie koordiniert man verschiedene Sensorwerte (IR-Sensor, Ultraschallsensor, USB-Kamera) für eine sichere Navigation? *(Kollaboration)*
- Welche Kompromisse sind zwischen Geschwindigkeit, Präzision und Sicherheit sinnvoll? *(kritisches Denken)*
- Wie lässt sich das Verhalten des FTF von einer regelbasierten zu einer KI-gestützten Steuerung weiterentwickeln? *(Kreativität)*

DIE UNTERRICHTSIDEE AUF EINEN BLICK

Klassenstufe: 11–13

Zeitaufwand: 2–3 Doppelstunden pro Lerneinheit (erweiterbar bis zu 13 DS)

Schwierigkeitsgrad: Modell

Programmierung bis

Modellart: Tischmodelle für fahrerlose Transportfahrzeuge

MODELLBESCHREIBUNG / AUFGABE

Die Schülerinnen und Schüler (SuS) planen und realisieren fahrerlose Transportfahrzeuge (FTF), die sie Schritt für Schritt mit weiterer Sensorik und intelligenter Steuerung versehen. Ausgehend von einem Grundfahrzeug mit Encodermotoren, das definierte Strecken und Drehungen ausführt sowie erstes Linienfolgen mithilfe eines Spursensors bewältigt, über ein FTF mit Ultraschallsensor zur Kollisionsvermeidung sowie USB-Kamera zur Interaktion mit Farbflächen bis hin zum Regelkreis gesteuerten oder sogar KI-gestützten Linienfolgen entwickeln sich die Modelle fort – und damit auch die Ansprüche an Aufbau, Verdrahtung und Programmierung.



Hinderniserkennung

Die SuS lernen Zustandsübergangsdigramme für Fahrzustände zu erstellen und Sensordaten gezielt zu lesen und zu verwenden. Aus den Eigenschaften und Messwerten der Sensoren ermitteln sie passende Fahr- und Lenkgruppen, definieren Variablen für Weg- und Winkelsteuerung sowie Reaktionszeiten für sichere Kurskorrekturen.

Aus den so festgelegten Zuständen und Parametern entwickeln die SuS zunächst regelbasierte Steuerprogramme mit Variablen, Unterprogrammen und Zustandslogik; anschließend konfigurieren und trainieren sie ein neuronales Netz für ein Regressionsproblem, das aus Sensoreingängen geeignete Motorgeschwindigkeiten vorhersagt und als Differenzierung um den Abstandssensor erweitert wird. Die Tauglichkeit prüfen die SuS in Fahrversuchen auf dem Parcours – Geradeauslauf, Spurhalten, Ausweichen und Farbflächen-Reaktionen – und verbessern ihre Lösungen durch systematische Fehlersuche und Geschwindigkeitsoptimierung.

○ ALLTAGSBEZUG

Die sensorbasierte Steuerung eines Fahrzeugs ist den SuS aus dem Technik- oder Informatikunterricht und ihrem Alltag vertraut. Bekannte Anwendungen sind Tempomat und Spurhalteassistent im Auto, E-Scooter mit Sensorik sowie Saugroboter, die Linien folgen und Hindernisse meiden.

Die Einbettung in einen realitätsnahen Mobilitätskontext schafft hohe Motivationalität, weil die SuS Parallelen zu Stadtverkehr und Lagerlogistik unmittelbar erkennen.

Eine Integration der Thematik in die vorberufliche Orientierung bietet sich in Fahrzeugtechnik, Elektrotechnik und Robotik/Automatisierung an, wo sensorgetriebene Regelung und das automatisierte Ansteuern von Aktoren zentrale Kompetenzen sind.

Die Verbindung aus Sensorik und KI-gestützter Steuerung begegnet den SuS nicht nur in Industrie und Verkehr, sondern auch im häuslichen Umfeld, z. B. bei Smart-Home-Anwendungen, intelligenten Heizungen oder automatischer Beleuchtung – überall dort, wo Messwerte Entscheidungen auslösen und Bewegungen zuverlässig ausgeführt werden.

○ FÄCHERBEZUG

Informatik:	Fortgeschrittene Programmierung, Bedingungsschleifen, Funktionen, Zustandsautomaten, Ereignissteuerung, Kameraeinbindung, P- und PD-Regler, neuronale Netze, Training eines neuronalen Netzes
Physik:	Bewegung (Weg, Zeit, Geschwindigkeit), Signalverarbeitung, Ultraschall-Laufzeitmessung und Schallausbreitung, Trägheit und Bremsweg, Farberkennung, Regelungstechnik
Technik:	Stabiles Bauen, Konstruktionstechnik
Mathematik:	Berechnung von Termen, Skalierung, Proportionalität und lineare Funktionen, Einheitenrechnung, Normierung, Regression

Einführungsphase



Unterrichtsgespräch

- Bekanntgeben des Themas, ggf. Video „Fahrerloses Transportsystem“ zeigen.
→ www.youtube.com/watch?v=cJz3NxiKh4
- Abfragen, was die wesentlichen Merkmale für die Steuerung von Fahrerlosen Transportfahrzeugen sind.
- Szenarien abfragen, in denen Fahrerlose Transportfahrzeuge vorkommen.
- Einsatzmöglichkeiten diskutieren.
- Anforderungen an den Aufbau der Modelle ermitteln.



ggf. Hilfestellung

- Sensoren, Aktoren und Bauteile aus dem Baukasten zeigen, wenn nötig, Präsentationsmedien einsetzen.

Planungsphase



Unterrichtsgespräch

- Die Vorgehensweise zum Bau des Modells und die zu erzielende Funktion werden gemeinsam erarbeitet.
- Die Arbeitsschritte der App werden vorgegeben bzw. besprochen.



Partner- oder Einzelarbeit

- Die SuS machen sich mit der App vertraut und laden die entsprechende Aufgabe.
- Die SuS definieren sinnvolle Funktionen zur Steuerung eines FTFs mithilfe des TXT 4.0 Controllers.
- Die SuS erstellen mithilfe der App die Sensorliste für den aktuellen Anwendungsfall.
- Die SuS bereiten den Parcours vor. Die Fahrfläche sollte so glatt wie möglich sein.

Hinweis: Die Fahrexperimente sollten wahlweise auf dem Boden oder auf einer Fläche mit Randbegrenzung stattfinden, um ein Herunterfallen der Fahrzeugmodelle zu vermeiden.



Optional:
Partner- oder Gruppenarbeit

- Die SuS diskutieren die einsetzbare Sensorik und skizzieren mögliche Aufbauten.
- Die SuS diskutieren ihre Erfahrungen mit automatisiertem bzw. autonomen Fahren in der Gruppe.

Konstruktionsphase für das Fahrtraining (FTF 1)



Partner- oder Einzelarbeit

- Die SuS nutzen die App zum Bau des Hinderniserkenners. Die App führt kleinschrittig durchs Programm.
- Die SuS prüfen mithilfe des Schnittstellentests die Verkabelung der Sensoren.

Programmierphase für das Fahrtraining (FTF 1)



Partner- oder Einzelarbeit

- Die SuS entwickeln ein Programm für eine einfache Geradeausfahrt sowie für gezielte Drehungen mit dem FTF.
- Die SuS berechnen die dafür notwendigen Encoderimpulse und nehmen individuelle Anpassungen der Umrechnungsfaktoren „*Impulse pro cm*“ und „*Impulse pro Grad*“ an die Gegebenheiten des FTF vor.
- Die SuS erweitern das FTF um eine einfache Hinderniserkennung und erstellen ein einfaches Zustandsübergangsdiagramm zur Analyse der Programmieraufgabe.
- Die SuS entwickeln ein Programm zum einfachen Linienfolgen mithilfe einer digitalen Steuerung.
- Die App führt in aufeinander aufbauenden Stufen mit offenen Fragestellungen durch die Programmieraufgabe.
- Hilfe wird in der App angeboten.
- Das Programm wird nach jedem Differenzierungsschritt auf den TXT 4.0 Controller übertragen.

Experimentier- und Testphase für das Fahrtraining (FTF 1)



Partner- oder Gruppenarbeit

- Der Hinderniserkenner wird in Betrieb genommen.
- Mögliche Störungen im Funktionsablauf müssen gefunden und eliminiert werden.
- Eventuelle Fehlersuche ist anhand von Vorschlägen in der App möglich.
- Eventuelle Optimierungen bei der Hardware und der Programmierung, z. B. Anpassungen der Fahrgeschwindigkeit und Anpassungen des Geschwindigkeitsunterschieds der Motoren bei Drehungen, können vorgenommen werden.

Abschluss-/Anschlussphase für das Fahrtraining (FTF 1)



Optional:
Vorstellung und Zuteilung der Differenzierungen

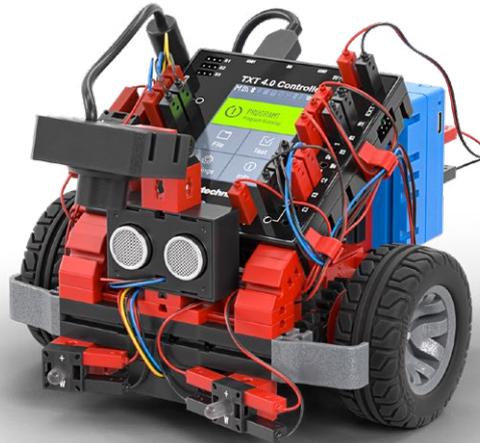
- Für die Differenzierung infrage kommende SuS (oder die gesamte Gruppe) werden ggf. durch die Lehrperson angesprochen. Dabei werden Möglichkeiten für das Umfahren von Hindernissen diskutiert.
- Die Integration der USB-Kamera unter Verwendung unterschiedlicher Farbräume wird vorgestellt.

Konstruktionsphase für den digitalen Spurfolger (FTF 2)



Partner- oder Einzelarbeit

- Die SuS nutzen die App zum Bau des digitalen Spurfolgers. Die App führt kleinschrittig durchs Programm.
- Die SuS prüfen mithilfe des Schnittstellentests die Verkabelung der Sensoren.



Digitaler Spurfolger

Programmierphase für den digitalen Spurfolger (FTF 2)



Partner- oder Einzelarbeit

- Die SuS skizzieren ein Zustandsübergangsdiagramm zur Planung der Programmieraufgabe für das Umfahren von Hindernissen mithilfe des digitalen Spurfolgers.
- Die SuS erweitern das Programm zum Linienfolgen aus FTF 1 um die Erkennung und das Umfahren eines Hindernisses.
- Die SuS planen die Integration der USB-Kamera, sodass das FTF auf Farbflächen rechts oder links der Spur reagieren kann.
- Die SuS konfigurieren die Farberkennung im HSV-Farbraum.
- Die SuS definieren die Aktionen für jede Farbfläche und integrieren sie in das Programm aus FTF 1, sodass das FTF beim Linienfolgen nun auch auf die erkannten Farbflächen reagiert.
- Die App führt in aufeinander aufbauenden Stufen mit offenen Fragestellungen durch die Programmieraufgabe.
- Hilfe wird in der App angeboten.
- Das Programm wird nach jedem Differenzierungsschritt auf den TXT 4.0 Controller übertragen.

Experimentier- und Testphase für den digitalen Spurfolger (FTF 2)



Partner- oder Gruppenarbeit

- Der digitale Spurfolger wird in Betrieb genommen.
- Die SuS testen die Reaktionen des FTF auf die Farbflächen entlang des Parcours.
- Mögliche Störungen im Funktionsablauf müssen gefunden und eliminiert werden.
- Eventuelle Fehlersuche ist anhand von Vorschlägen in der App möglich.

Abschluss-/Anschlussphase für den digitalen Spurfolger (FTF 2)



Optional:
Vorstellung
und Zuteilung
weiterer Diffe-
renzierungen

- Für die Differenzierung infrage kommende SuS (oder die gesamte Gruppe) werden ggf. durch die Lehrperson angesprochen. Dabei wird das Konzept des Regelkreises in der Steuerungstechnik vorgestellt.

Konstruktionsphase für den analogen Spurfolger (FTF 3)



Partner- oder
Gruppenarbeit

- Die SuS nutzen die App zum Bau des analogen Spurfolgers. Die App führt kleinschrittig durchs Programm.



Analoger Spurfolger

Programmierphase für den analogen Spurfolger (FTF 3)



Partner- oder
Gruppenarbeit

- Die SuS planen die Integration der USB-Kamera, sodass das FTF in diesem Fall über die Linienerkennung mithilfe der Kamera einer Spur folgen kann.
- Die SuS konfigurieren die Linienerkennung mithilfe der Kamera im RGB-Farbraum.
- Die SuS planen und implementieren einen P-Regler.
- Die SuS erweitern die Programmierung zu einem PD-Regler.
- Die App führt in aufeinander aufbauenden Stufen mit offenen Fragestellungen durch die Programmieraufgabe.
- Hilfe wird in der App angeboten.
- Das Programm wird nach jedem Differenzierungsschritt auf den TXT 4.0 Controller übertragen.

Experimentier- und Testphase für den analogen Spurfolger (FTF 3)



Partner- oder Gruppenarbeit

- Der analoge Spurfolger wird in Betrieb genommen.
- Mögliche Störungen im Funktionsablauf müssen gefunden und eliminiert werden.
- Eventuelle Fehlersuche ist anhand von Vorschlägen in der App möglich.
- Die SuS testen verschiedene Einstellungen für den Proportionalitätsfaktor des P-Reglers.
- Die SuS vergleichen das Fahrverhalten des FTFs bei Steuerung mit einem P- bzw. PD-Regler.

Abschlussphase für den analogen Spurfolger (FTF 3)



Optional:
Vorstellung und Zuteilung weiterer Differenzierungen

- Für die Differenzierung infrage kommende SuS (oder die gesamte Gruppe) werden ggf. durch die Lehrperson angesprochen. Dabei sollen die Konzepte der neuronalen Netze und der KI vorgestellt werden.

Konstruktionsphase des KI-Linienfolgers (FTF 4)



Partner- oder Einzelarbeit

- Die SuS nutzen die App zum Bau des digitalen Spurfolgers. Die App führt kleinschrittig durchs Programm.
- Die SuS prüfen mithilfe des Schnittstellentests die Verkabelung der Sensoren.
- Falls der digitale Spurfolger in FTF 2 bereits gebaut wurde, entfällt die Konstruktionsphase.

Programmierphase des KI-Linienfolgers (FTF 4)



Partner- oder Einzelarbeit

- Die SuS machen sich mit dem Aufbau eines einfachen neuronalen Netzes vertraut.
- Die SuS konfigurieren mithilfe der App ein neuronales Netz, das aus den Sensordaten des Spursensors Motordrehzahlen berechnet und geben Trainingsdaten vor.
- Die SuS erweitern ggf. die Programmierung aus FTF 1 so, dass die Drehzahlen der beiden Motoren nun durch das neuronale Netz bestimmt werden.
- Die SuS trainieren das neuronale Netz.
- Die SuS ergänzen das Programm um die Abfrage des Abstandssensors.
- Die SuS erweitern das neuronale Netz um ein Eingangsneuron, das die Sensordaten des Abstandssensors aufnimmt.

Experimentier- und Testphase des KI-Linienfolgers (FTF 4)



Partner- oder Gruppenarbeit

- Die SuS beobachten das Fahrverhalten des FTFs und testen, wie sich Anpassungen der Tabelle mit den Trainingsdaten, unterschiedliche Einstellungen für das Training und unterschiedliche Konfigurationen des neuronalen Netzes auf das Fahrverhalten auswirken.
- Die SuS testen das Bremsverhalten des FTFs.
- Die SuS optimieren das Fahr- und Bremsverhalten des FTFs.

Abschlussphase des KI-Linienfolgers (FTF 4)



Optional:
Vorstellung und Zuteilung weiterer Differenzierungen

- Besonders effizient arbeitenden SuS oder Gruppen kann die Erweiterungsaufgabe gestellt werden, eigene Anwendungsfälle für das Fahrverhalten des FTFs zu entwickeln und die dazu notwendige Sensorik auszuwählen.
- Die SuS können die im Verlauf der Lerneinheiten erworbenen Programmierkenntnisse und Regeltechniken kombinieren, um ein FTF mit möglichst gleichmäßigem Fahrverhalten und vielfältigen Reaktionsmöglichkeiten zu entwickeln.
- Verschiedene Gruppen können gegeneinander antreten: Welches FTF absolviert den Parcours am schnellsten? Welches FTF ist am stabilsten gegenüber Störungen? Hier ist selbstständiges Arbeiten ohne Lernkarten vorgesehen.



Diskussion im Plenum

- Nachbesprechung des Projekts im Klassenverbund
- Klärung von zukünftigen Einsatzmöglichkeiten im Alltag



METHODISCH-DIDAKTISCHE HINWEISE

Differenzierungsmöglichkeiten

Je nach Dauer der Unterrichtsreihe und der Stärke der SuS kann

- die Komplexität der Aufgabe durch Einbeziehung mehrerer Sensoren gesteigert werden.
- eine Erweiterung der Fähigkeiten des FTFs durch komplexere Strategien für das Umfahren von Hindernissen oder die Aufnahme und Reaktion auf farbliche Reize aus der Umgebung erfolgen.
- das Fahrverhalten des FTFs durch Anwendung von P- und PD-Reglern gleichmäßiger gestaltet werden.
- die Motorsteuerung beim Linienfolgen oder Abbremsen durch ein neuronales Netz übernommen werden.

Motivationale Aspekte

Das Arbeiten mit fahrerlosen Transportfahrzeugen (FTF) knüpft unmittelbar an Alltagserfahrungen der SuS an. Schon beim **Fahrtraining** erleben sie, wie bereits wenige Programmzeilen ein Fahrzeug zuverlässig in Bewegung setzen – eine direkte Rückmeldung, die eine hohe Motivationalität aufweist und Neugier weckt.

Beim **digitalen Spurfolger** verstärkt sich der Alltagsbezug durch Parallelen zu Saugrobotern, Rasenmärobotern oder Fahrerassistenzsystemen, die Linien erkennen und Hindernisse selbstständig vermeiden. Die Vorstellung, dass das FTF nun wie ein Roboter „sehen“ und reagieren kann, fördert die Identifikation mit der Aufgabe und steigert das Interesse am praktischen Testen.

Der **analoge Spurfolger** eröffnet mit Kamera- und Regelungstechnik spannende Einblicke in moderne Fahrzeugtechnologien wie Spurhalteassistenten. Hier wird deutlich, wie Parameterveränderungen oder die Hinzunahme eines Differentialanteils das Fahrverhalten beeinflussen – ein motivierendes Experimentierfeld.

Mit dem **KI-Linienfolger** schließlich rückt Künstliche Intelligenz in den Mittelpunkt: SuS erleben, wie das Training eines neuronalen Netzes das Fahrverhalten verbessert. Dies schlägt die Brücke zu aktuellen Diskussionen über KI im Alltag und gibt den SuS die Möglichkeit, Zukunftstechnologien selbst verstehen und anwenden zu können.

GRUNDLAGEN REGELUNGSTECHNIK

Ein P-Regler ist die einfachste Form einer **Rückkopplungsregelung**: Er vergleicht fortlaufend einen gewünschten Zielwert (Soll) mit dem aktuellen Messwert (Ist) und wandelt die dabei entstehende Abweichung – den Fehler – direkt in eine Korrektur um, deren Größe **proportional** zum Fehler ist. Je größer der Fehler, desto kräftiger die Gegensteuerung; die „Empfindlichkeit“ des Systems bestimmt der Proportionalfaktor k_P . Ist k_P zu klein, reagiert das System träge und kommt nur langsam auf Kurs; ist er zu groß, reagiert es über, beginnt zu schwingen und wird unruhig. Im Unterrichtskontext des analogen Spurfolgers bedeutet das: Die Kamera misst die seitliche Abweichung von der Spur, der P-Regler passt die linke und rechte Motor-geschwindigkeit gegensinnig an und bringt das Fahrzeug zurück in die Spurmitte. Der P-Regler ist damit die Basis jeder weiteren Verfeinerung (z. B. durch einen D-Anteil) und eignet sich hervorragend, um mit wenig Programmieraufwand das Grundprinzip automatischer, stabilisierender Steuerungen anschaulich zu machen.

Die zentrale Regelgröße beim Spurfolgen ist die seitliche Abweichung der Spurmitte von der Bildmitte der Kamera. Wir bezeichnen diese Abweichung als Fehler e in **Pixeln** (positiv, wenn die Spur rechts der Bildmitte liegt; negativ, wenn sie links liegt). Der Regler bildet aus diesem Fehler eine Stellgröße u , mit der die linke und die rechte Motorgeschwindigkeit gegensinnig angepasst wird. Mit einer Basisgeschwindigkeit v_0 und der Begrenzung auf den zulässigen Bereich $[0, v_{\max}]$ ergibt sich

$$v_{\text{links}} = \text{clip}(v_0 + u, 0, v_{\max}), \quad v_{\text{rechts}} = \text{clip}(v_0 - u, 0, v_{\max}).$$

Dabei begrenzt („sättigt“) die clip-Funktion den Wert der Geschwindigkeit auf den zulässigen Bereich. Liegt die Spur rechts der Mitte, soll das FTF nach rechts abbiegen, um die Spur wiederzufinden. Dazu muss das rechte Rad langsamer und das linke Rad schneller drehen. Liegt die Spur links der Mitte, muss das FTF eine Linkskurve fahren, d.h. das linke Rad muss nun langsamer und das rechte Rad schneller drehen.

Beim **Proportional-Regler (P-Regler)** ist die Stellgröße proportional zum momentanen Fehler

$$u = k_P \cdot e,$$

wobei k_P die Einheit „Geschwindigkeit pro Pixel“ hat.

Beispiel 1: $v_0 = 320$, $v_{\max} = 500$, $k_P = 1,2$, Spur bei $+40$ px, rechts der Bildmitte, also $e = +40$ px. Damit ergibt sich $u = 1,20 \cdot 40 = 48$, $v_{\text{links}} = 368$, $v_{\text{rechts}} = 272$ (keine Sättigung nötig).

Beispiel 2: $v_0 = 320$, $v_{\max} = 500$, $k_P = 1,2$, Spur bei -35 px, links der Bildmitte, also $e = -35$ px. Damit ergibt sich $u = 1,20 \cdot (-35) = -42$, $v_{\text{links}} = 278$, $v_{\text{rechts}} = 362$ (keine Sättigung nötig).

Falls eine Geschwindigkeit die Grenzen $[0, v_{\max}]$ überschreitet, wird sie durch die clip-Funktion begrenzt. Dauerhaftes Clipping sollte man jedoch durch etwas kleinere Werte für k_P oder v_0 vermeiden.

Der **Proportional-Differential-Regler (PD-Regler)** ergänzt den P-Regler um einen vorausschauenden D-Anteil, der die Änderungsgeschwindigkeit des Fehlers berücksichtigt. Mit der Diskretisierung i formuliert, ergibt sich bei einer Abtastzeit Δt von für die Stellgröße $u[i]$

$$u[i] = k_P \cdot e[i] + k_D \frac{e[i] - e[i-1]}{\Delta t}.$$

Hier hat k_D die Einheit „Geschwindigkeit je Pixel pro Sekunde“. Nähert sich das FTF bereits der Spurmitte, wirkt der D-Anteil bremsend und reduziert Überspringen. Wächst der Fehler, d.h. kommt das FTF immer stärker von der Spur ab, verstärkt der D-Anteil die Korrektur und hilft so, schneller zur Spur zurückzukehren.

Beispiel 3 (Spur rechts, Annäherung an die Mitte):

$v_0 = 320$, $v_{\max} = 500$, $k_P = 1,2$, $k_D = 0,03$ (Geschwindigkeit je Pixel pro Sekunde),

$\Delta t = 0,02$ s (Abtastrate), $e[i] = +45$ px, $e[i-1] = +60$ px.

Für den P-Anteil gilt dann $k_P \cdot e = 1,20 \cdot 45 = 54$.

Die Änderungsrate beträgt $(45-60)/0,02 = -750 \text{ px/s}$. Der D-Anteil liefert also $0,03 \cdot (-750) = -22,5$. Insgesamt ergibt sich $u = 54 - 22,5 = 31,5$. Damit werden die Motorgeschwindigkeiten zu $v_{\text{links}} = 320 + 31,5 = 351,5$, $v_{\text{rechts}} = 320 - 31,5 = 288,5$. Ohne D-Anteil ergäbe ein reiner P-Regler $u = 54$, also $v_{\text{links}} = 374$, $v_{\text{rechts}} = 266$; der PD-Anteil dämpft die Korrektur sichtbar, weil sich das Fahrzeug der Spurmitte bereits nähert.

Beispiel 4 (Spur links, Fehler wächst):

$v_0 = 320$, $v_{\text{max}} = 500$, $k_P = 1,2$, $k_D = 0,03$ (Geschwindigkeit je Pixel pro Sekunde),
 $\Delta t = 0,02 \text{ s}$ (Abtastrate), $e[i] = -45 \text{ px}$, $e[i-1] = -20 \text{ px}$.
 Für den P-Anteil gilt dann $k_P \cdot e = 1,20 \cdot (-45) = -54$.

Die Änderungsrate beträgt $(-45-(-20))/0,02 = -1250 \text{ px/s}$. Der D-Anteil liefert also $0,03 \cdot (-1250) = -37,5$. Insgesamt ergibt sich $u = -54 - 37,5 = -91,5$. Damit werden die Motorgeschwindigkeiten zu $v_{\text{links}} = 320 - 91,5 = 228,5$, $v_{\text{rechts}} = 320 - (-91,5) = 411,5$; der PD-Anteil verstärkt die Korrektur deutlich, weil sich das Fahrzeug weiter von der Spurmitte entfernt.

Wenn das Programm für unterschiedliche Bildbreiten W funktionieren soll, wäre es sinnvoll, mit einem normierten Fehler zu arbeiten, d. h. $e_n = x/W \in [-1, 1]$.

PROGRAMMIERKENNTNISSE

- Programmstart
- Dauerschleife **wiederhole dauerhaft (repeat forever)**
- Kamerakonfiguration
- Schleife **wiederhole solange (repeat while)**
- Befehl **warte bis (wait until)**
- Bedingung **falls – mache (if – do)**
- Nutzen von Variablen und deren Veränderung
- Rechnen mit Variablen
- Arbeit mit Unterprogrammen
- Programmierung Neuronaler Netze
- Eingabe von KI-Trainingsdaten

Zum Download optional:

- Stromlaufplan
- Bauanleitung

ZUSATZMATERIALIEN

- Falls vorhanden, könnte für die Einführungsphase in das Thema ein Video genutzt werden.
- Zeichenmedien (Papier, Whiteboard oder Projektionsfläche).

—○ FUNKTIONEN DES MODELLS UND DEREN TECHNISCHE LÖSUNGEN

Funktion der Sensoren/Aktoren	Technische Lösung
Drehung der Encodermotoren	Geschwindigkeitsanpassung zur Fahrzeugsteuerung
Mini-Taster	Erkennen von Hindernissen (FTF 1)
Messung von Helligkeitsunterschieden	Erkennen der Fahrspur (FTF 1–2, FTF 4)
Messung von Abständen	Kollisionsvermeidung (FTF 2, FTF 4)
FTF 2: Farberkennung durch USB-Kamera	Reaktion auf Farbflächen
FTF 3: Spurerkennung durch USB-Kamera	Spurhalten mithilfe eines P-Reglers und eines PD-Reglers
FTF 4: Spurerkennung durch Spursensor	<ul style="list-style-type: none"> • Aufbau eines neuronalen Netzes • Eingabe von Trainingsdaten • Umsetzung eines Regressionsproblems mithilfe von KI
Weitere Differenzierungsmöglichkeiten	Optimierung der Geschwindigkeitssteuerung, Optimierung der Strategien für das Umfahren von Hindernissen, Optimierung der Strategie für das Wiederfinden der Spur bei komplettem Spurverlust, Konzeption eines eigenen Fahrzeugs mit selbst gewählter Sensorausstattung



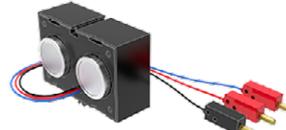
MATERIALLISTE

Sensoren	Funktion
1 On/Off-Taster am TXT 4.0 Controller	Einschalten des FTF
2 Mini-Taster	Hinderniserkennung (FTF 1)
1 Spursensor (mit 2 IR-Sensoren)	Spurerkennung (FTF 1–2, FTF 4)
1 Ultraschallsensor	Abstandsmessung (FTF 2, FTF 4)
1 USB-Kamera	Farberkennung (FTF 2) Spurerkennung (FTF 3)
Aktoren	Funktion
2 Encodermotoren	Antrieb der Fahrzeuge
2 LEDs (2 × weiß)	Scheinwerfer (FTF 2–4)

Wo ist was verbaut?



Encodermotor
Art. Nr. 153422



Ultraschallsensor m. ft-Steckern
Art. Nr. 219835



USB-Kamera
Art. Nr. 152522

1	Analoge Sensoren MODELL 1		1 ×	
2	Spritzenpumpe MODELL 2	1 ×		
3	Tormodell MODELL 3	1 ×	1 ×	1 ×
4	Parkhaus 1: Parkhausschranke MODELL 4	1 ×		
5	Parkhaus 2: Einfahrt und Ausfahrt 2 × MODELL 4	2 ×		
6	Parkhaus 3: Kennzeichenerkennung MODELL 5	1 ×		1 ×
7	Ampelsteuerung 1: Lichtsignalanlage MODELL 6			
8	Ampelsteuerung 2: Verkehrskreuzung 2 × MODELL 6			
9	Ampelsteuerung 3: KI-Phasensteuerung MODELL 7			1 ×
10	FTF 1: Fahrtraining MODELL 8	2 ×		
11	FTF 2: Digitaler Spurfolger MODELL 9	2 ×	1 ×	1 ×
12	FTF 3: Analoger Spurfolger MODELL 10	2 ×		1 ×
13	FTF 4: KI-Linienfolger MODELL 9	2 ×	1 ×	1 ×

AKTOREN UND SENSOREN



IR-Spursensor m. ft-Steckern
Art. Nr. 219831



Mini-Taster
Art. Nr. 037783



LED
Art. Nr. 162134



Lichtschranken LED
Art. Nr. 162135



Fototransistor
Art. Nr. 036134



NTC-Widerstand
Art. Nr. 219830

					1 ×	1
	2 ×	2 ×				2
	2 ×	2 ×	1 ×	1 ×		3
	2 ×	2 ×	1 ×	1 ×		4
	4 ×	4 ×	2 ×	2 ×		5
	2 ×	2 ×	1 ×	1 ×		6
	1 ×	4 ×	1 ×			7
	2 ×	8 ×	2 ×			8
	1 ×	4 ×	1 ×			9
1 ×	2 ×					10
1 ×		2 ×				11
		2 ×				12
1 ×		2 ×				13