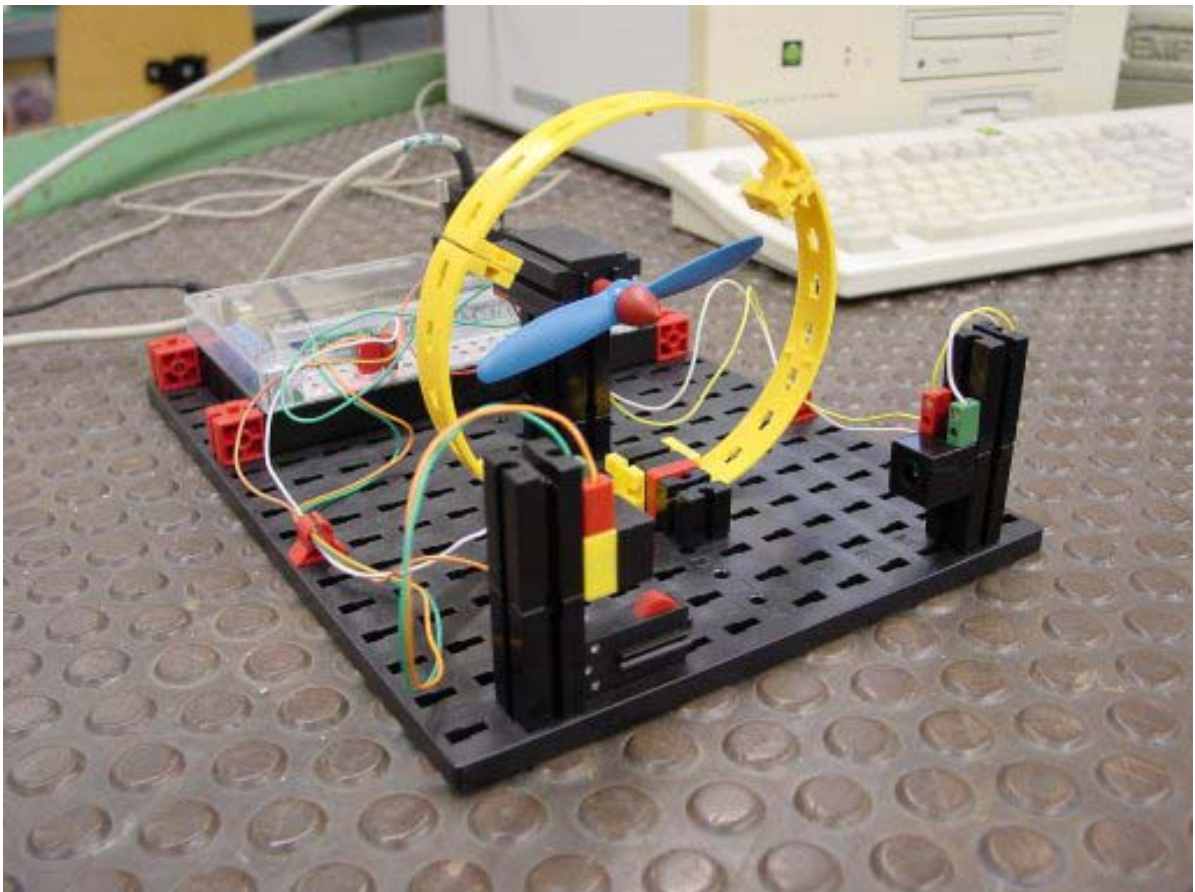


Fischertechnik im praktischen Schulunterricht

Schülerheft



Name: _____

Klasse: _____

INHALTSVERZEICHNIS

1. EINLEITUNG	7
2. BESCHREIBUNG DER KOMPONENTEN	9
2.1. Lernbaukasten " <i>Mechanik</i> "	9
2.2. Lernbaukasten " <i>Elektrische Schaltungen</i> "	9
2.3. " <i>Computing Starter Pack</i> "	9
2.4. Das " <i>Intelligent Interface</i> "	10
2.5. Die Software " <i>LLWin</i> "	11
2.6. Schülertafel	13
2.7. Symboltafeln	13
3. PRAKTISCHE ANWENDUNGEN MIT DEM BAUKASTEN "MECHANIK"	15
3.1. Projekt 1: Ausgabe und Eingabe erkennen	15
3.1.1. Lernziel	15
3.1.2. Erklärung	15
3.1.3. Aufgabe	15
3.1.4. Schlussfolgerung	17
3.2. Projekt 2: Eine Drehbewegung übertragen	18
3.2.1. Lernziel	18
3.2.2. Aufgaben	18
3.2.3. Schlussfolgerung	25
3.3. Projekt 3: Eine Drehbewegung in eine lineare Bewegung umwandeln	26
3.3.1. Lernziel	26
3.3.2. Erklärung	26
3.3.3. Aufgaben	26
3.3.4. Schlussfolgerung	30
3.4. Projekt 4: Hin- und Herbewegung und schwingende Bewegung	31
3.4.1. Lernziel	31
3.4.2. Aufgaben	31
3.4.3. Schlussfolgerung	35
3.5. Projekt 5: Eine Drehbewegung übertragen	36
3.5.1. Lernziel	36
3.5.2. Aufgaben	36
3.5.3. Schlussfolgerung	39
3.5.4. Aufgabe	39
3.5.5. Schlussfolgerung	40
3.5.6. Aufgabe	40
3.5.7. Schlussfolgerung	41
4. PRAKTISCHE ANWENDUNGEN MIT DEM BAUKASTEN "ELEKTRISCHE SCHALTUNGEN"	43
4.1. Projekt 1: Energieformen	43
4.1.1. Lernziel	43
4.1.2. Erklärungen	43
a) Stromkreis	43
b) Strom	44

c) Spannung	44
4.1.3. Aufgaben	46
4.2. Projekt 2: Stromkreise mit zwei Glühlampen	48
4.2.1. Lernziel	48
4.2.2. Aufgaben	48
4.3. Projekt 3: Stromkreis mit Schalter	50
4.3.1. Lernziel	50
4.3.2. Erklärung	50
4.3.3. Aufgabe	51
4.3.4. Schlussfolgerung	51
4.3.5. Aufgabe	52
4.3.6. Schlussfolgerung	52
4.4. Projekt 4: Elektrische und mechanische Systeme	53
4.4.1. Lernziel	53
4.4.2. Erklärungen	53
4.4.3. Aufgaben	54
4.4.4. Schlussfolgerung	57
4.5. Projekt 5: Schaltkreise	58
4.5.1. Lernziel	58
4.5.2. Erklärungen	58
4.5.3. Aufgaben	59
4.6. Projekt 6: Einen Motor schalten	62
4.6.1. Lernziel	62
4.6.2. Aufgaben	62
4.6.3. Schlussfolgerung	66
5. PRAKTISCHE ANWENDUNGEN MIT DEM BAUKASTEN “COMPUTING STARTER“	67
5.1. Die wichtigsten Bauteile	67
5.1.1. Motor	67
5.1.2. Getriebe	67
5.1.3. Kugellampe	67
5.1.4. Linsenlampe	67
5.1.5. Fototransistor	68
5.1.6. Taster	68
5.1.7. NTC-Widerstand	68
5.2. Projekt 1: Einfache Motorsteuerung	69
5.2.1. Lernziel	69
5.2.2. Aufgabe	69
5.2.3. Erklärungen	70
a) Test der Hardware	70
b) Erstes Steuerungsprogramm	71
c) Verbindungslinien	73
d) Baustein START	74
e) Baustein EINGANG	74
f) Baustein AUSGANG	76
g) Baustein ENDE	77
h) Testen	78
i) Symbolleiste	78
j) Öffnen (Projekt Menü)	79

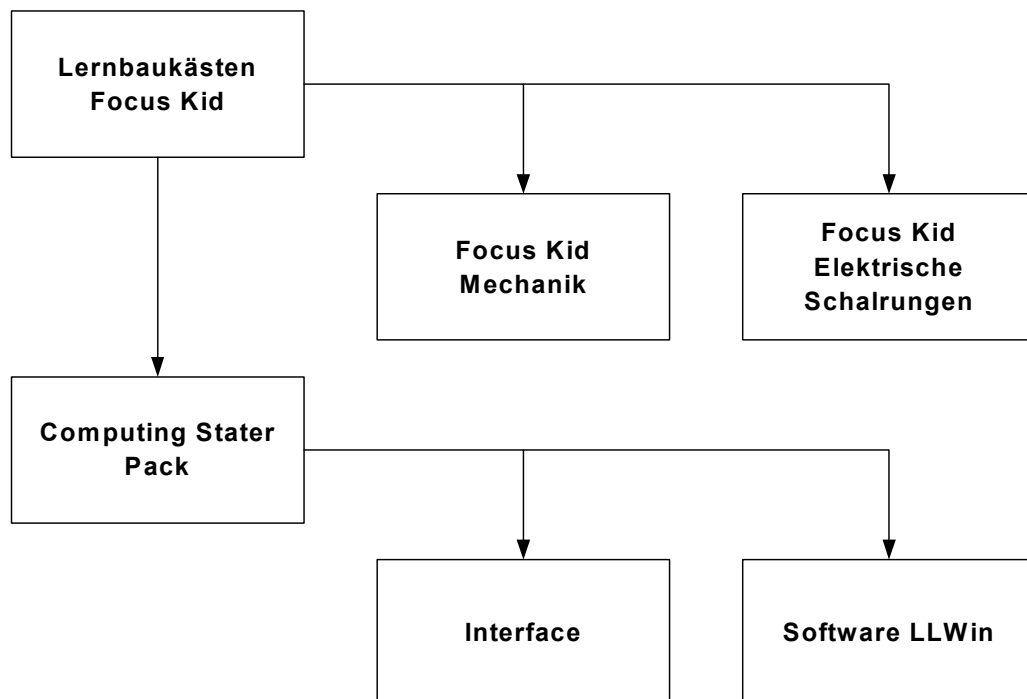
k)	Speichern (Menü Projekt)	79
l)	Speichern unter (Menü Projekt)	79
m)	Schließen (Menü Projekt)	80
5.2.4.	Aufgabe	81
5.3.	Projekt 2: Erweiterte Motorsteuerung	84
5.3.1.	Lernziel	84
5.3.2.	Aufgaben	84
5.3.3.	Erklärungen	87
a)	Grundlagen zur Verwendung von Variablen	87
b)	Steuern paralleler Abläufe mit Variablen	87
c)	Baustein VARIABLE+/-1	88
d)	Baustein VERGLEICH	88
e)	Baustein WERTE ANZEIGEN	89
5.4.	Projekt 3: Motorsteuerung mit Zeitverzögerung	91
5.4.1.	Lernziel	91
5.4.2.	Aufgaben	91
5.4.3.	Erklärungen	91
a)	Baustein WARTE	91
5.5.	Projekt 4: Verkehrsampel	93
5.5.1.	Lernziel	93
5.5.2.	Aufgabe	93
5.5.3.	Funktionsbeschreibung	93
5.5.4.	Aufgabe	93
5.5.5.	Erklärungen	93
a)	Baustein UNTERPROGRAMM	93
b)	Beispiel für die Verwendung eines Unterprogramms	94
5.6.	Projekt 5: Händetrockner	98
5.6.1.	Lernziel	98
5.6.2.	Aufgabe	98
5.6.3.	Funktionsbeschreibung	98
5.6.4.	Aufgabe	98
5.7.	Projekt 6: Schiebetür	99
5.7.1.	Lernziel	99
5.7.2.	Aufgabe	99
5.7.3.	Erklärungen	99
a)	Baustein TEXT	99
5.7.4.	Funktionsbeschreibung	99
5.7.5.	Aufgabe	100
5.8.	Projekt 7: Temperaturregelung	101
5.8.1.	Lernziel	101
5.8.2.	Aufgabe	101
5.8.3.	Erklärungen	101
a)	Verarbeiten von Analogwerten mit Variablen	101
b)	Baustein TERMINAL	102
5.8.4.	Funktionsbeschreibung	104
5.9.	Projekt 8: Stanzmaschine	106
5.9.1.	Lernziel	106
5.9.2.	Aufgabe	106
5.9.3.	Funktionsbeschreibung	106
5.9.4.	Erklärungen	107

a) Baustein BEEP	107
b) Baustein FLANKE	107
5.10. Projekt 9: Parkhausschranke	109
5.10.1. Lernziel	109
5.10.2. Aufgabe	109
5.10.3. Funktionsbeschreibung	109
5.10.4. Funktionsbeschreibung	109
5.11. Projekt 10: Schweißroboter	111
5.11.1. Lernziel	111
5.11.2. Aufgabe	111
5.11.3. Funktionsbeschreibung	111
5.11.4. Erklärungen	112
a) Baustein POSITION	112
b) Baustein RESET	114

1. EINLEITUNG

Maschinen und Roboter konstruieren, am PC die Steuersoftware dafür schreiben und das Ganze in Bewegung setzen, all dies ist in unserer digitalisierten Welt zu einem wichtigen Teil des Fachbereichs Elektrotechnik geworden. Dies wird in dieser Arbeit, dem Schüler, indem er Projekte plant und entwirft, anhand der Modelle von Fischertechnik näher gebracht.

Hierzu werden in dieser Arbeit folgende Fischertechnik Baukästen benutzt:



An Hand der Bausätze werden verschieden mechanische Antriebe durchgespielt um danach einige dieser Antriebe elektrisch zu betreiben. Zuerst geschieht dies mit der konventionellen Technik (Schalter usw.) später dann werden einfache Bewegungsmodelle über Interface und PC mit Hilfe der Software LLWin gesteuert.

Durch eine Auflage bei der Genehmigung dieser Arbeit liegt der Schwerpunkt auf dem praktischen Teil, dem Lehrgerät, mit dem der Lehrer die Software auch dann erklären kann wenn er in seiner Werkstatt nicht über PC, Laptop, Projektor usw. verfügt. Auch die Schüler können auf ihren zwölf Tafeln ein Steuerungsprogramm erstellen ohne zur Hilfenahme eines Computers.

2. BESCHREIBUNG DER KOMPONENTEN

2.1. Lernbaukasten "*Mechanik*"

Der Baukasten Mechanik (2 – 3 Schüler) bietet eine breite Palette an mechanischen Systemen, darunter:

- Drehbewegung, Kraftübertragung, Getriebesysteme,
- Umwandlung von Drehbewegungen in lineare Bewegungen,
- Umwandlung von Drehbewegungen in Hin- und Herbewegungen bzw. schwingende Bewegungen,
- Ketten, Riemenscheiben und Riemenantriebe.



2.2. Lernbaukasten "*Elektrische Schaltungen*"

Dieser Baukasten (2 – 3 Schüler) gilt als Ergänzungskasten zum Lernbaukasten Mechanik und erklärt:

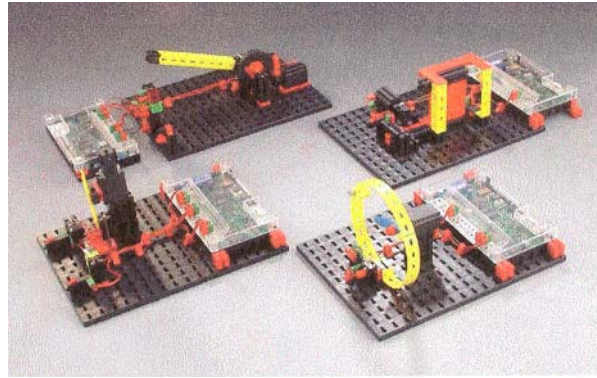
- Einfache Stromkreise mit Leuchten,
- Verwendung von Schaltern zur Steuerung von Vorrichtungen EIN/AUS,
- Schließer oder Öffner,
- Motor.



2.3. "*Computing Starter Pack*"

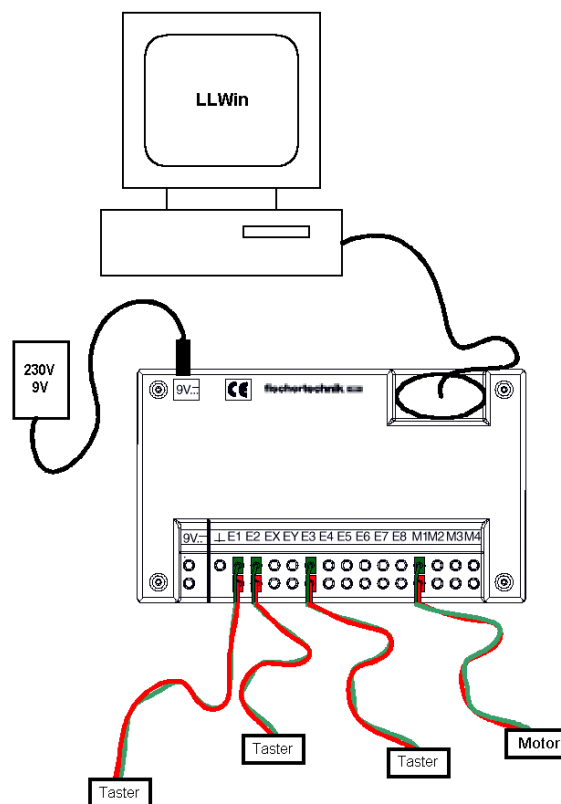
Unter dem Begriff "Computing" versteht man bei Fischertechnik das Programmieren und Steuern von Modellen über den PC. Der Baukasten "Computing Starter" bildet den optimalen Einstieg in dieses Thema. Man kann 8 verschiedene Modelle, vom Händetrockner über eine Parkhausschranke bis hin zum Schweißroboter, mit Hilfe der Bauanleitung in kürzester Zeit aufbauen. Über das Intelligent Interface verbindet man die Modelle mit dem PC. Schließlich programmiert man die Modelle schnell und einfach mit der grafischen Programmiersoftware LLWin 3.0.

- Motorsteuerung
- Händetrockner
- Verkehrsampel
- Schiebetür
- Temperaturreglung
- Stanzmaschine
- Parkhausschranke
- Schweißroboter



2.4. Das "Intelligent Interface"

Das "Intelligent Interface" ist das Herzstück der Fischertechnik-Computing-Reihe. Es ermöglicht die Kommunikation zwischen dem PC und den Modellen. Es dient dazu, die Befehle der Software so umzuwandeln, dass beispielsweise Motoren angesteuert und Signale von Sensoren wie Tastern, Fototransistoren, Reedkontakten u. a. verarbeitet werden können.



Das Interface:

4 digitale Ausgänge:

Belastbarkeit der Ausgänge: Dauerstrom 250 mA

Strombegrenzung auf 1 A, kurzschlussfest.

8 digitale Eingänge: Spannungsbereich 9 V DC.

2 analoge Eingänge: Anschluss von Potentiometern, Wärme- oder Helligkeitssensoren

Stromversorgung: 9 V-Batterien bzw. Akkus

Netzteil 9 V DC/1000 mA.

Das "Intelligent Interface" wird über das mitgelieferte Schnittstellenkabel an einer freien seriellen Schnittstelle (z.B. COM1 bis COM4) am PC angeschlossen. Der Mikroprozessor bildet die Steuerzentrale des Interface. Der Prozessor arbeitet in zwei unterschiedlichen Betriebsformen. Sie werden Online-Modus und Download-Modus bezeichnet:

Im Online-Modus erfolgt die Abarbeitung des Programms auf dem PC. Das Verbindungskabel zum Computer kann nicht abgezogen werden. Im Download-Modus erfolgt die Abarbeitung des Programms entkoppelt vom PC auf dem Mikroprozessor des Interface. Mit der Software "LLWin" ab Version 2.1 erstellte Programme können im Download-Modus in den RAM-Speicher des Interface geladen werden. Dies tut man im Menü RUN mit dem Befehl DOWNLOAD. Danach bricht das Interface die Verbindung zum PC ab, und das Schnittstellenkabel kann abgezogen werden. Das Programm bleibt solange im RAM erhalten, bis die Stromversorgung unterbrochen wird.

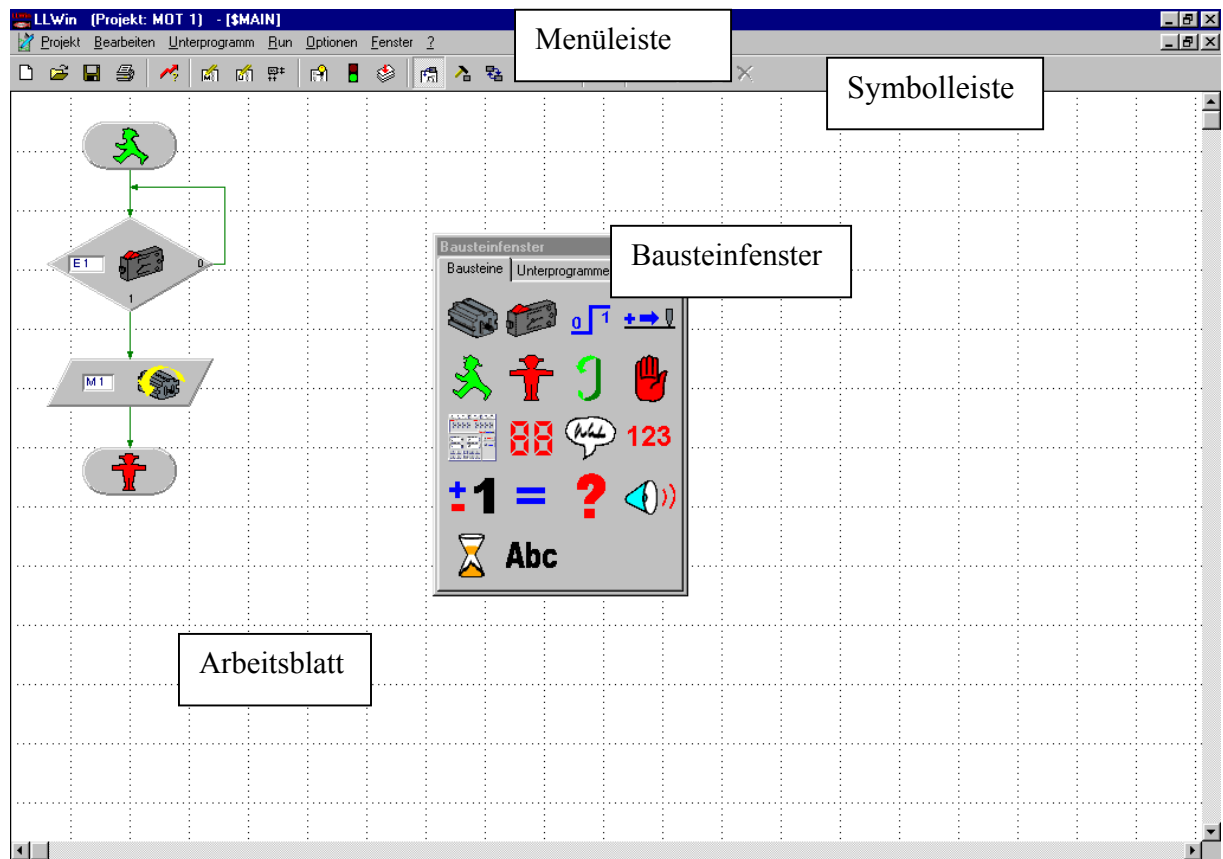
2.5. Die Software "LLWin"

So wie man mit Fischertechnik unterschiedliche Modelle aus gelben, roten und schwarzen Kunststoffbausteinen zusammen bauen kann, werden in LLWin Bausteine zu einem Steuerungsprogramm in Form von Programmablaufplänen (kurz Abläufe) zusammen gesetzt.

Die Befehle dieser Steuersprache werden mit graphischen Symbolen, die als Bausteine bezeichnet werden, dargestellt. Durch Verbindungslinien zwischen den Bausteinen wird die Reihenfolge, in der die Befehle abgearbeitet werden, festgelegt.

Sind PC, Interface und Modell fehlerfrei verdrahtet, können in den Abläufen digitale Eingänge (z.B. Taster) abgefragt und digitale Ausgänge (z.B. Motoren) eingeschaltet werden. Für jedes Modell wird ein eigenes Steuerungsprogramm benötigt. Aber auch für das gleiche Modell

können unterschiedliche Steuerungsprogramme erstellt werden. Eine Datei, die ein aus Bausteinen zusammengesetztes Steuerungsprogramm enthält, wird in LLWin Projekt genannt. Bevor die Funktionen, die LLWin beinhaltet, näher beschrieben werden, ist hier erst einmal ein kleiner Überblick über die Benutzeroberfläche.



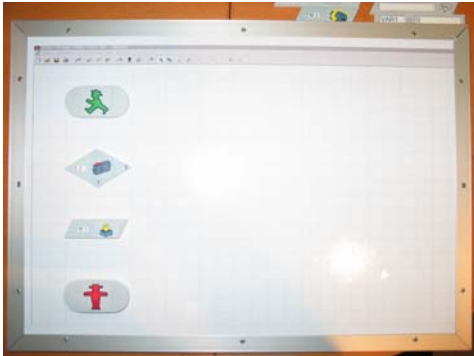
In der Menüleiste findet man alle Befehle, die zum Bedienen der Software benötigt werden. Neben den Befehlen zum Verwalten, Erstellen und Testen der Projekte sind hier auch die verschiedenen Befehle zu den Programmoptionen und die Online-Hilfe zu finden. Die wichtigsten Menübefehle können auch über Icons in der Symbolleiste (Seite 78) aufgerufen werden.

Aus dem Bausteinfenster können die Bausteine herausgezogen und auf dem Arbeitsblatt eingefügt werden. Durch Verbinden der Bausteine entstehen die Abläufe des Steuerungsprogramms. Die einzelnen Arbeitsschritte werden im Abschnitt "Das erste Steuerungsprogramm" genau beschrieben.

Das fertige Programm kann anschließend überprüft und mit einem angeschlossenen Interface getestet werden.

2.6. Schülertafel

Für die Schüler sind 12 Tafeln, im DIN A3 Format mit den dazugehörigen Magnetbausteinen hergestellt worden.



Die Verbindungslinien werden von den Schülern mit Hilfe eines Whiteboardstiftes, welcher später einfach mit einem trockenen Tuch abgewischt werden kann, realisiert.



2.7 Symboltafeln

Auf diese Oberfläche kann man mit Hilfe von Magnettafeln auf denen die Symbole der Bausteine mit Hilfe einer Selbstklebefolie aufgebracht sind, erklären wie ein Programmablauf erstellt wird.



3. PRAKTISCHE ANWENDUNGEN MIT DEM BAUKASTEN “*MECHANIK*“

3.1. Projekt 1: Ausgabe und Eingabe erkennen

3.1.1. Lernziel

Die Schüler sollen die Eingabe und die Ausgabe bei den verschiedensten Mechanismen erkennen können.

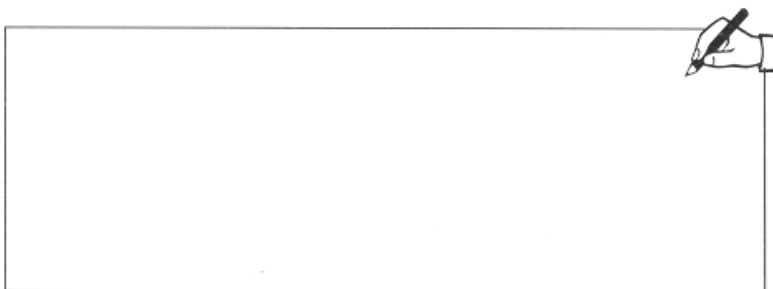
3.1.2. Erklärung

Ein Mechanismus ist eine Vorrichtung, die eine Bewegung überträgt, so dass die Ausgangsbewegung (Ausgabe) an einer anderen Stelle stattfindet als die Eingangsbewegung (Eingabe). Mechanismen können auch die Richtung und die Geschwindigkeit einer Bewegung ändern.

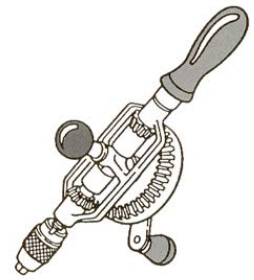
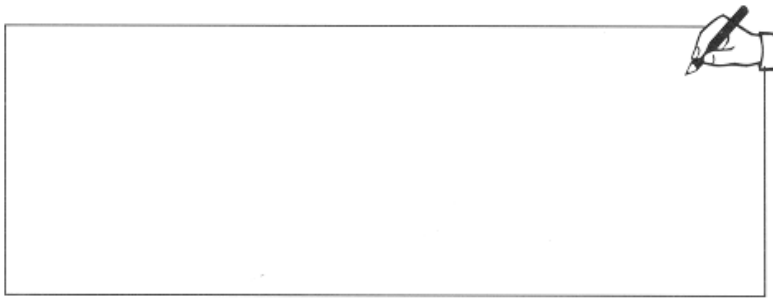
3.1.3. Aufgabe

Wie sieht die Eingabe und die Ausgabe bei den auf diesen Bildern gezeigten Mechanismen aus? Wie wird die Bewegung geändert?

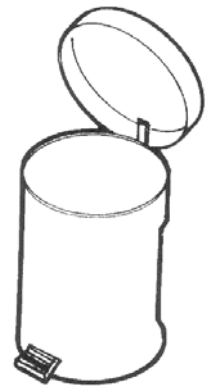
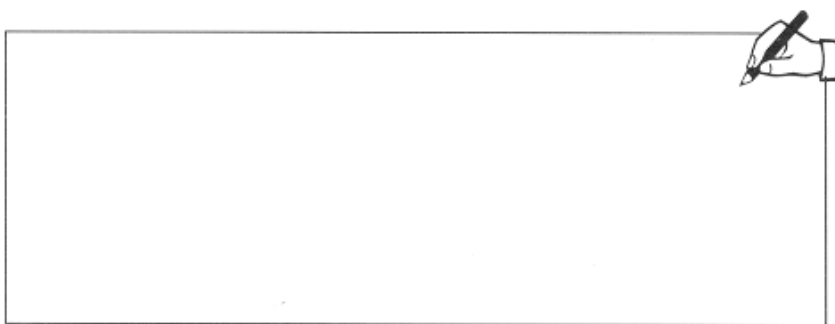
Fahrrad:



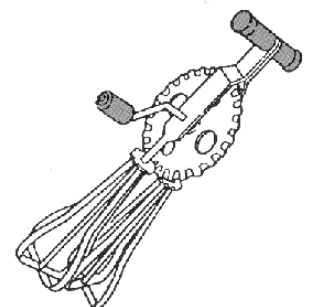
Handbohrer:



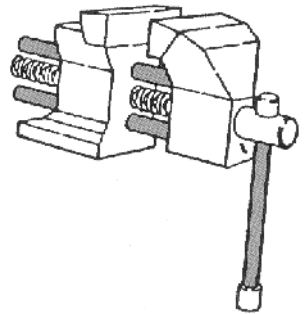
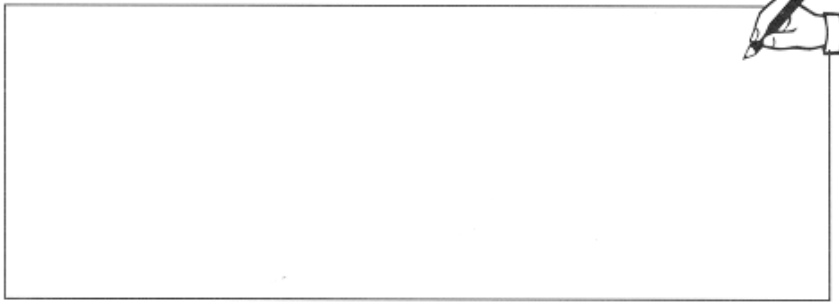
Treteimer:



Schneebesen:



Schraubstock:



3.1.4. Schlussfolgerung

3.2. Projekt 2: Eine Drehbewegung übertragen

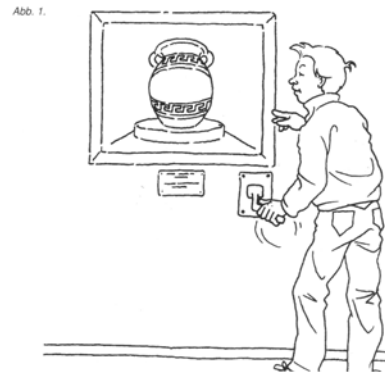
3.2.1. Lernziel

Der Schüler soll die verschiedenen Mechanismen zur Übertragung einer Drehbewegung kennen und verstehen lernen.

3.2.2. Aufgaben

Situation

Eine seltene griechische Vase wird in einem Schaukasten aufbewahrt, der auf der Vorderseite mit einer Glasscheibe versehen und aus Sicherheitsgründen in die Wand eingelassen ist. Künstler und Schüler kommen aus der ganzen Welt um sich die Vase anzusehen, aber das Problem besteht darin, dass man nur die Vorderseite der Vase sehen kann.



Eine Lösung wäre, die Vase auf eine Drehscheibe zu stellen, die die Besucher mit einem Griff außerhalb des Schaukastens drehen können.

Problemstellung

Die Techniker des Museums müssen ein mechanisches System konstruieren und herstellen, um die **Drehbewegung** vom Griff (Eingabe) zur Drehscheibe (Ausgabe) zu übertragen.

Drehbewegung

Das Symbol für eine Drehbewegung sieht folgendermaßen aus:



Der Mechanismus, den die Techniker verwenden, muss die Richtung der Bewegung ändern, wie in Abb. 1 zu sehen ist.

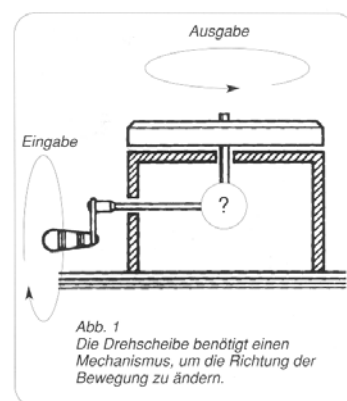
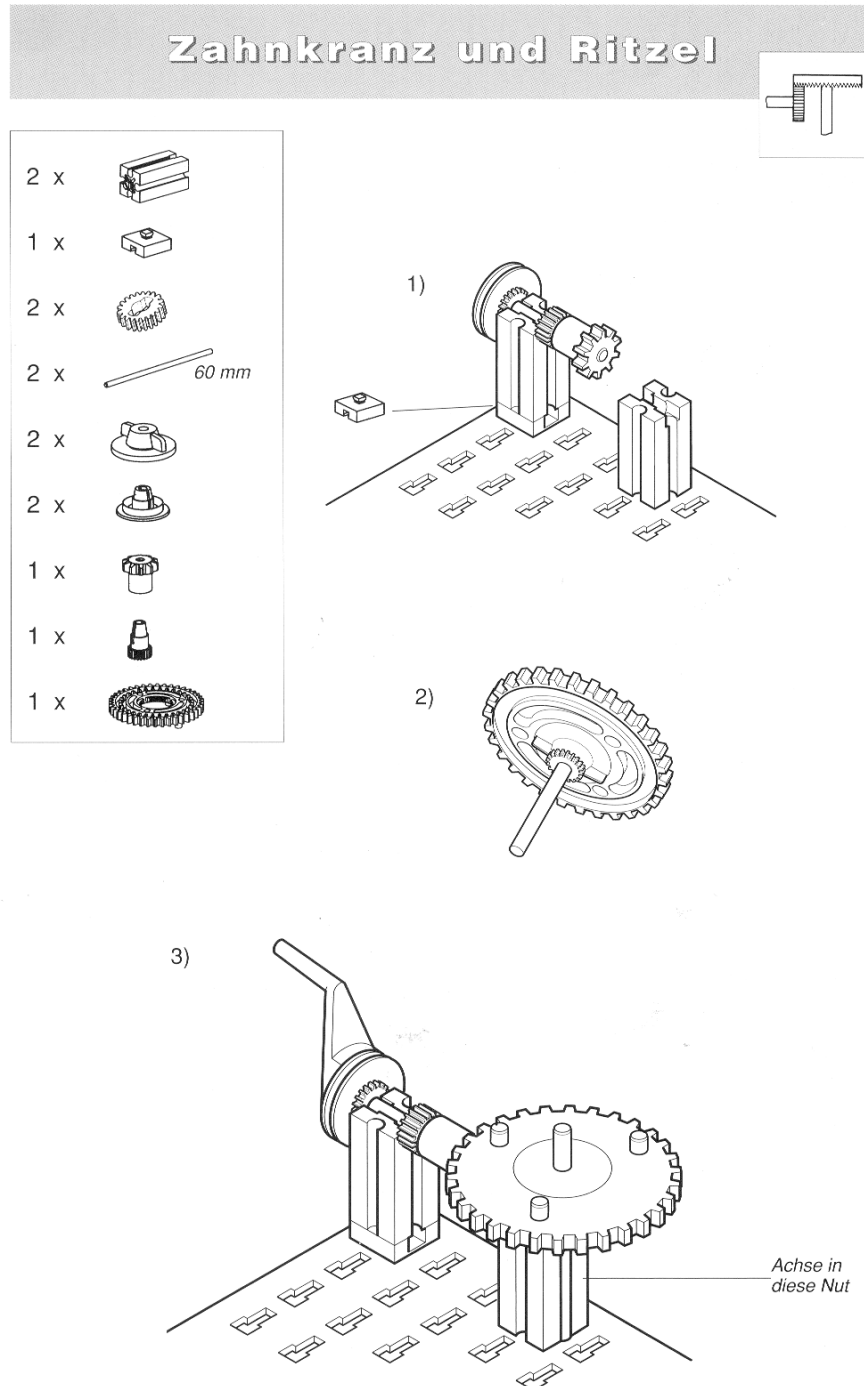


Abb. 1
Die Drehscheibe benötigt einen Mechanismus, um die Richtung der Bewegung zu ändern.

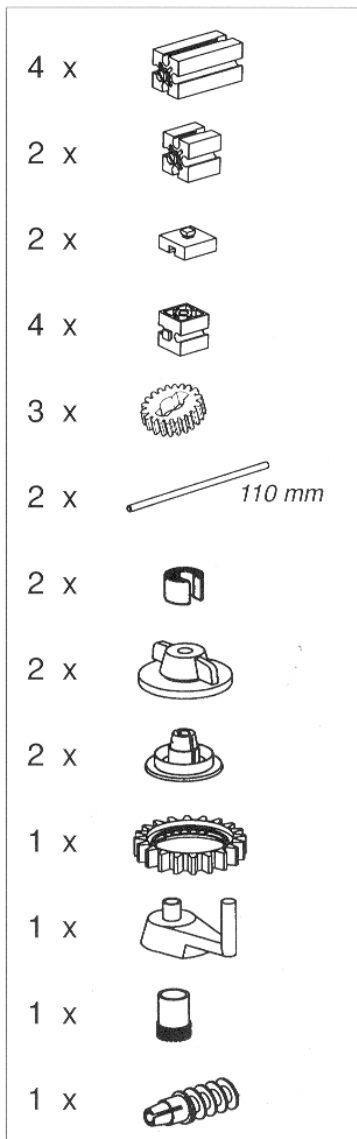
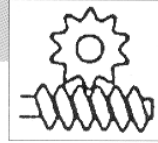
Wenn sie mögliche Mechanismen untersuchen, müssen sie die folgenden Punkte berücksichtigen:

- Die Vase muss sich auf eine sehr langsame, kontrollierte Weise drehen, so dass sie nicht wackelt oder umfällt.

- b) Die Vase ist schwer. Es ist eine beträchtliche Kraftanstrengung erforderlich um sie zu drehen.
- 1) Baue jetzt die vier folgenden Modelle zusammen und antworte auf die danach folgenden Fragen um die verschiedenen Mechanismen zu analysieren!

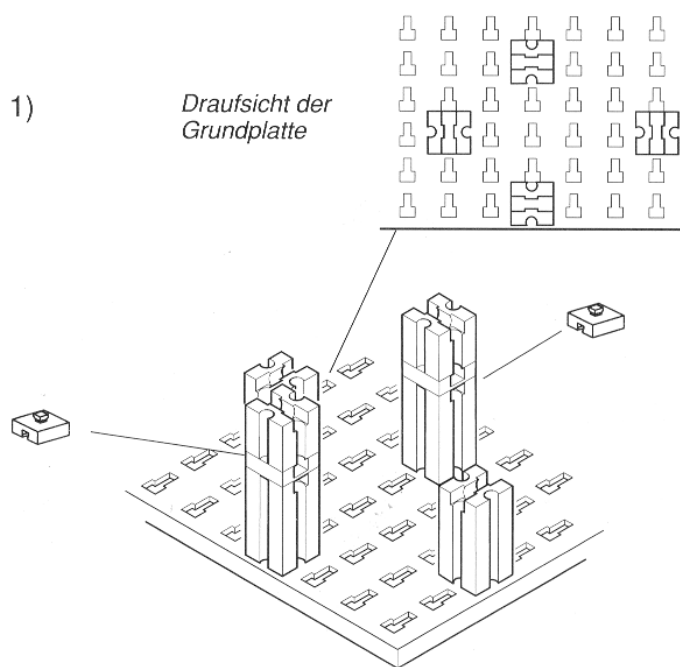


Schnecke und Schneckenrad

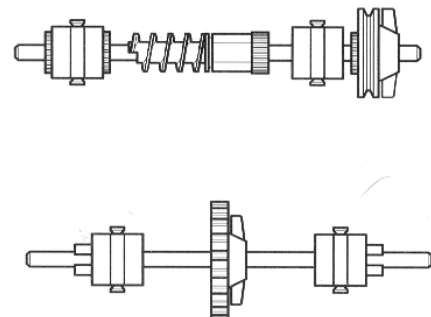


1)

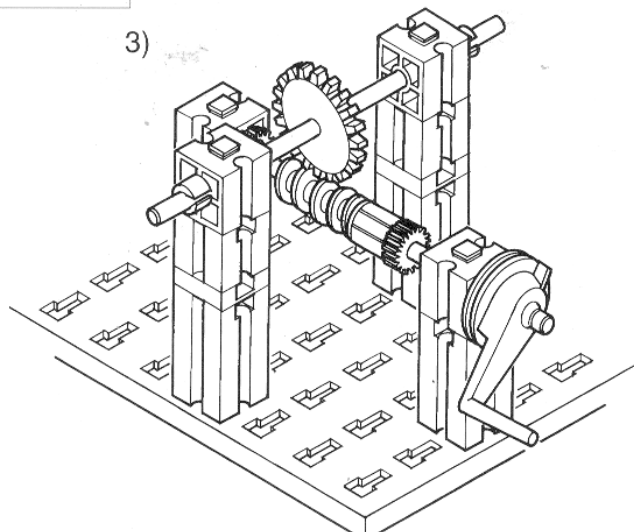
Draufsicht der Grundplatte



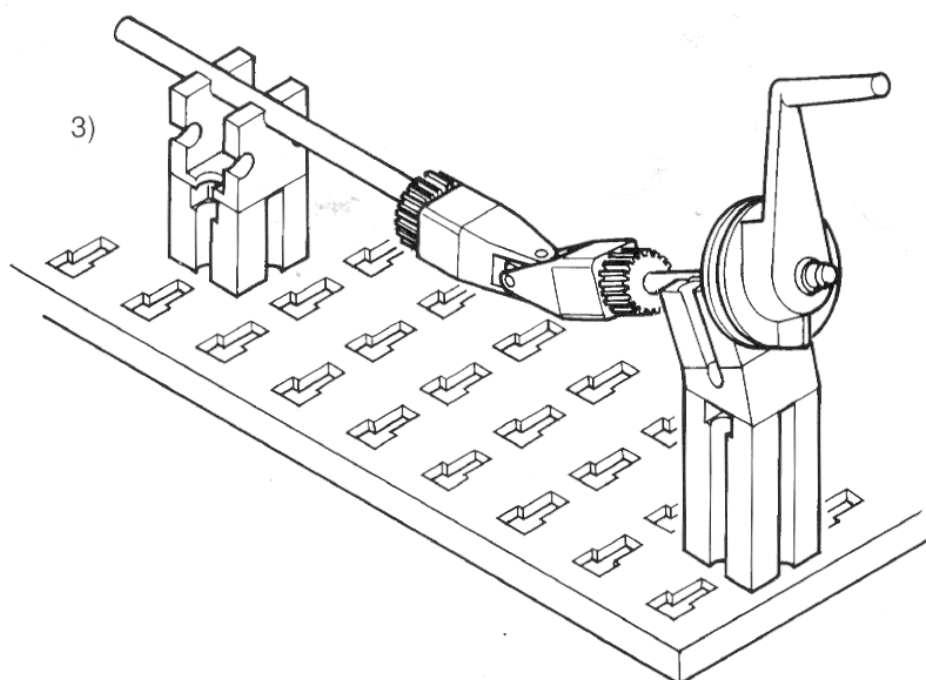
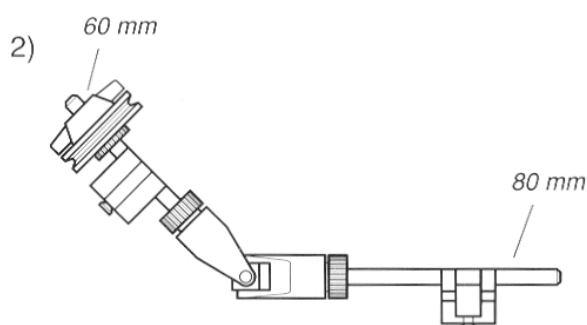
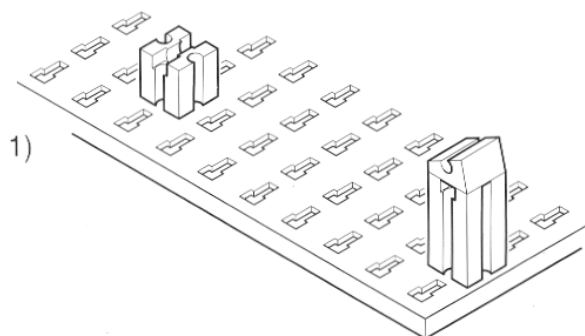
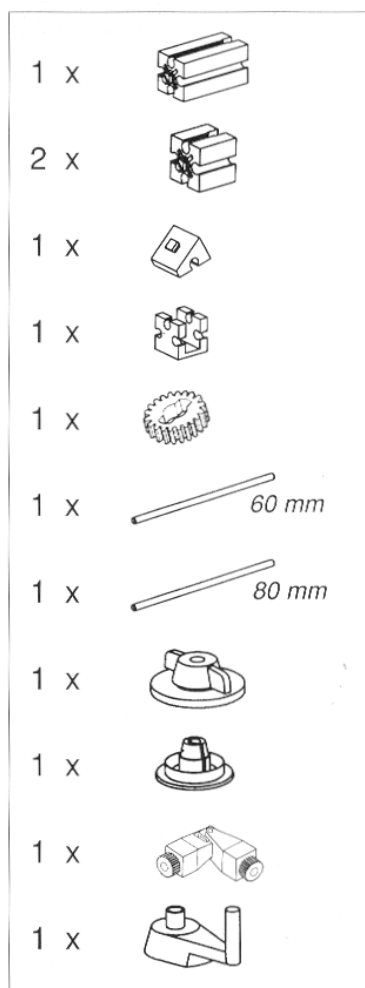
2)



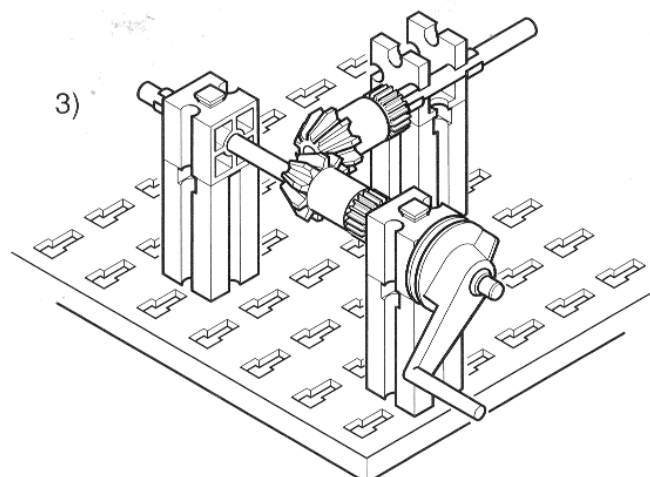
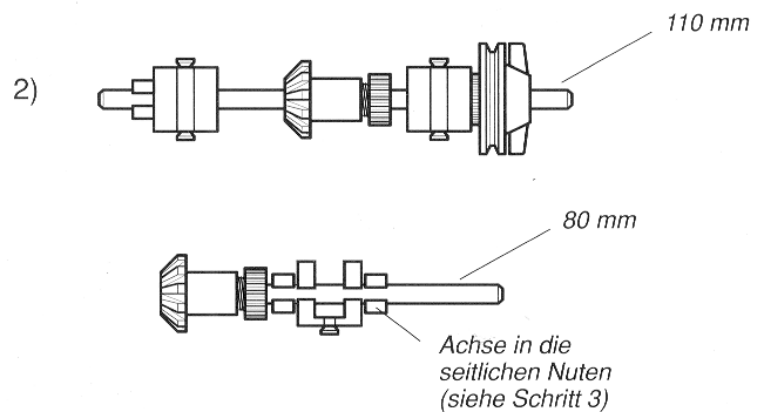
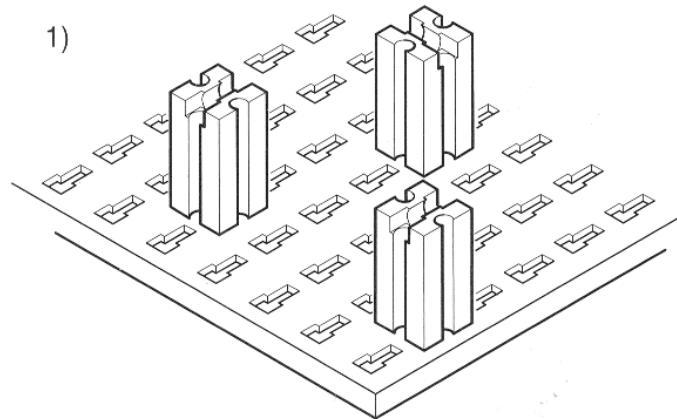
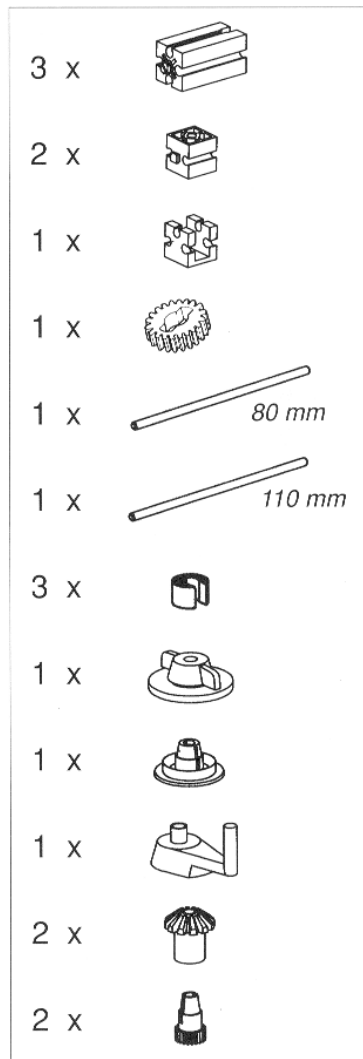
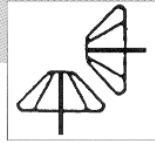
3)



Kardangelenk



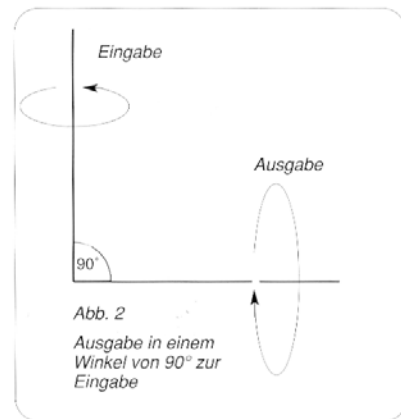
2 Kegelräder



Drehe jeweils am Griff jedes der Modelle und schaue, wo die Ausgabe geschieht!

Jeder Mechanismus überträgt eine Drehbewegung um eine Ecke. Die Ecke kann in jeder Richtung liegen. Probiere die Mechanismen aus, wenn die Grundplatte auf die Kante und auf die Kopfseite gelegt ist!

Sowohl der Handbohrer als auch der Schneebesen enthalten Mechanismen, um eine Drehbewegung um eine Ecke zu übertragen.



- 2) Bei drei der Mechanismen welche du gerade gebaut hast, liegt die Ausgabe in einem Winkel von 90° zur Eingabe.

Beim vierten Modell ist dies nicht der Fall.

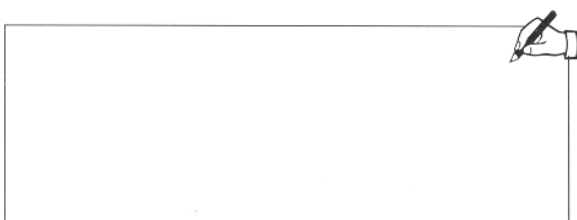
Welcher Mechanismus gehört nicht dazu?



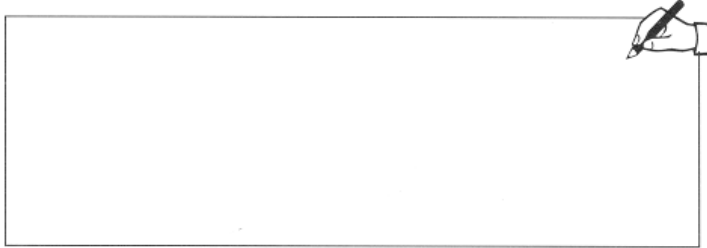
Funktioniert dieser Mechanismus bei 90° ? Versuche herauszufinden, warum bzw. warum nicht und unterstreiche die richtige Antwort!

1. Funktioniert.
2. Funktioniert nicht, ab einem bestimmten Winkel klemmt das Kardangelenke.
3. Funktioniert sehr schwergängig.

- 3) Bei zwei der Mechanismen wird die Geschwindigkeit der Bewegung geändert, daher ist die Ausgabe langsamer als die Eingabe. Welches sind die beiden?



Welcher Mechanismus könnte auch verwendet werden, um die Bewegung zu beschleunigen? Wie würdest du dies tun?

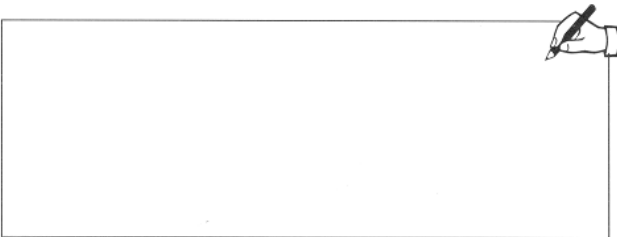


- 4) Bei zwei der Mechanismen wird die Kraftübertragung (Drehmoment) erhöht.

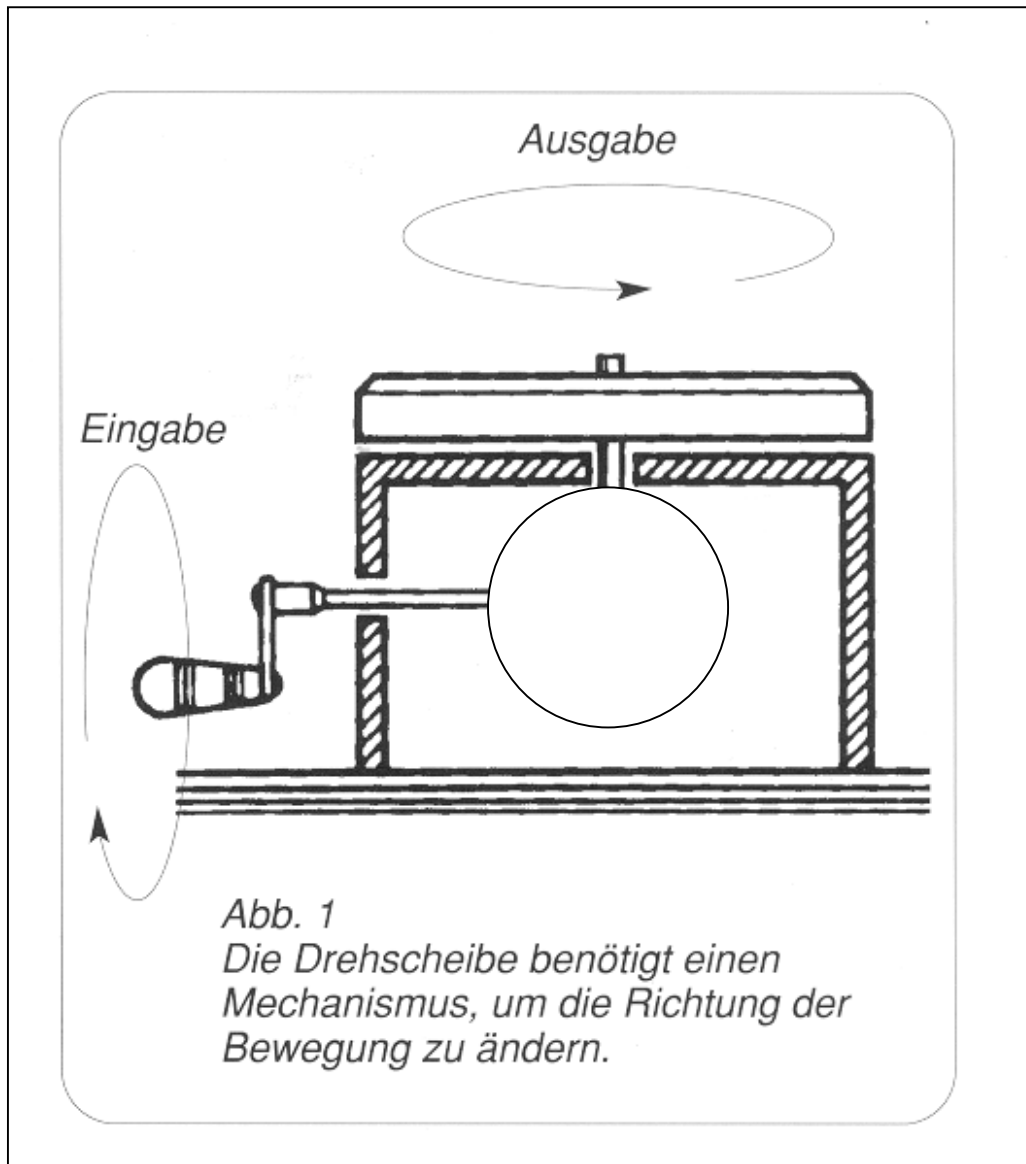
Führe das folgende Experiment mit jedem der Mechanismen durch.

Halte die Ausgangsachse fest und versuche sie anzuhalten, während du an der Eingabe drehst. Wenn es schwierig ist sie anzuhalten, dann bedeutet dies, dass das Drehmoment durch den Mechanismus erhöht wird.

Bei welchen beiden Mechanismen wird das Drehmoment erhöht?



- 5) Schau dir nochmals die Problemstellung für die Drehscheibe an und wähle den Mechanismus oder die Kombination von Mechanismen aus, die die Techniker des Museums deiner Meinung nach verwenden sollten. Skizziere deine Konstruktion in das folgende Kästchen.



3.2.3. Schlussfolgerung

3.3. Projekt 3: Eine Drehbewegung in eine lineare Bewegung umwandeln

3.3.1. Lernziel

Der Schüler soll die Mechanismen zur Umwandlung einer Drehbewegung in eine lineare Bewegung kennen und verstehen lernen.

3.3.2. Erklärung

Mechanismen können eine Bewegung übertragen und deren Geschwindigkeit, Kraft und Richtung ändern.

Sie können ebenfalls eine Art von Bewegung in eine andere Art umwandeln. Die Eingabe für das in Abb. 1 gezeigte Schiebetürensysteem ist die Drehbewegung von einem Elektromotor. Die Ausgabe ist die Bewegung der Tür in einer geraden Linie, wenn sie sich öffnet und schließt.

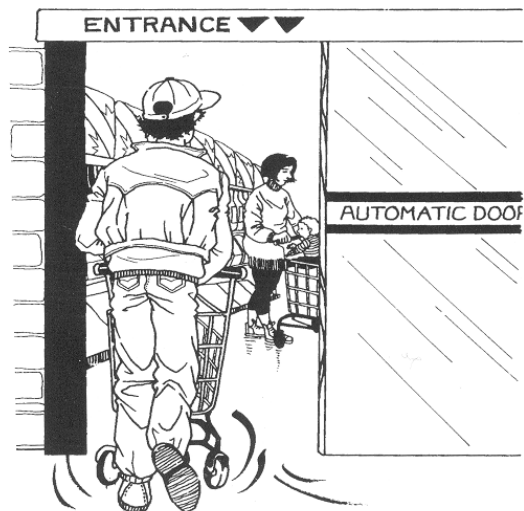


Abb. 1

3.3.3. Aufgaben

- 1) Baue die **Zahnstangen-** und **Schrauben-spindel-Modelle** auf separaten Grundplatten und antworte auf die danach folgenden Fragen um die verschiedenen Mechanismen zu analysieren! Beide dieser Mechanismen wandeln die Drehbewegung in eine **lineare** Bewegung.

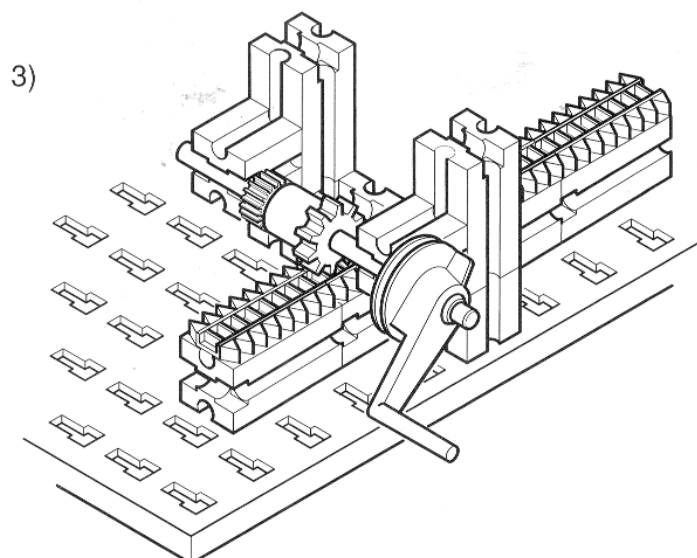
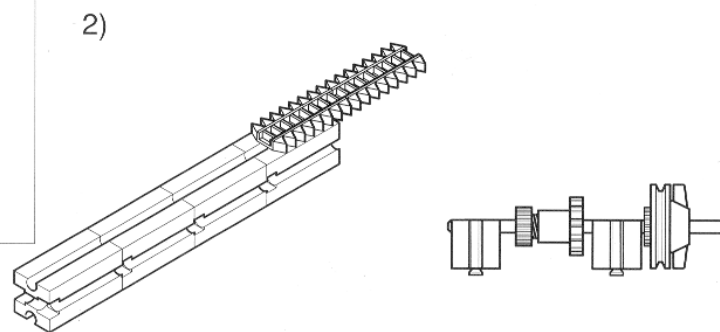
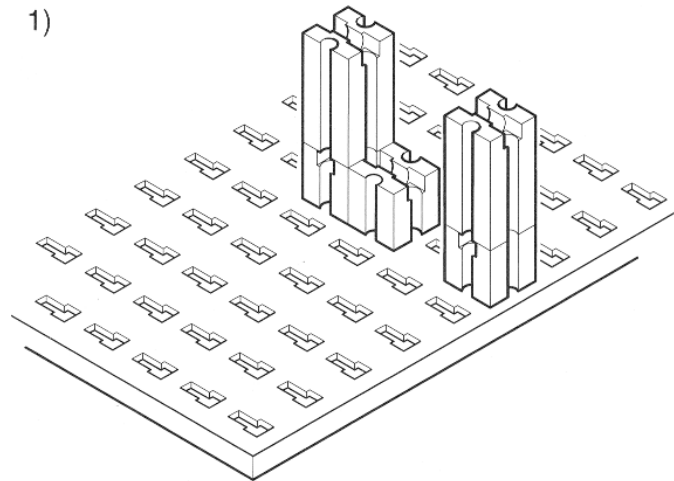
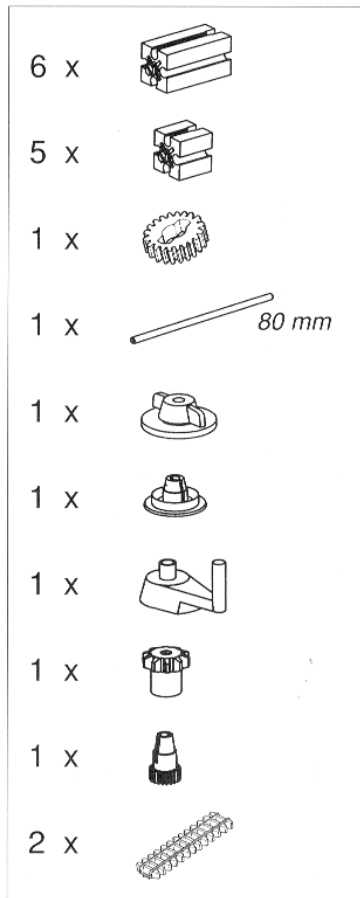
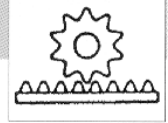
Lineare Bewegung

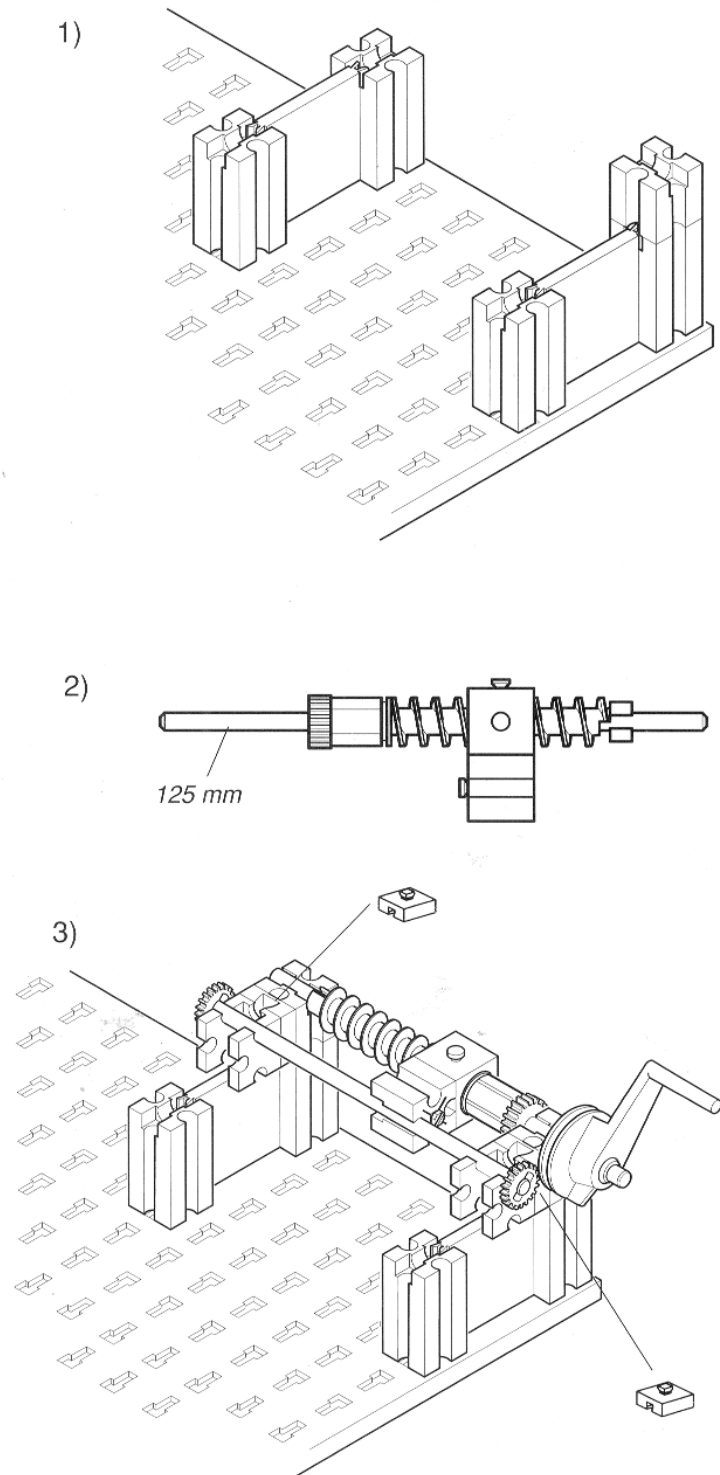
Linear bedeutet eine Hin- und Herbewegung in einer geraden Linie.

Das Symbol für eine lineare Bewegung sieht folgendermaßen aus:

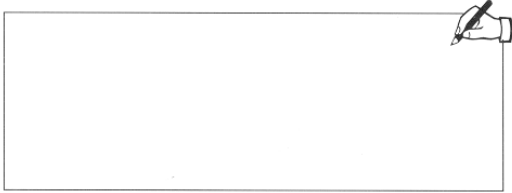


Zahnstange und Ritzel





- 2) Nur durch einen der Mechanismen wird die Kraft erhöht. Halte die Ausgabe von jedem Mechanismus fest und versuche ihn anzuhalten, wenn du am Griff drehst. Bei welchem nimmt die Kraft zu?



- 3) Versuche die Ausgabe jedes Mechanismus zu ziehen oder zu schieben. Dreht sich der Griff? Welcher dieser beiden Mechanismen ist umkehrbar?



- 4) Führe die folgenden Versuche durch um die Geschwindigkeit der Ausgabe jedes Mechanismus miteinander zu vergleichen.

a) Zahnstange

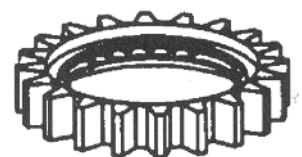
Drehe den Griff einmal in einer vollen Umdrehung. Messe, wie weit sich die Zahnstange bewegt.



Das Zahnrad des Modells besitzt 10 Zähne. Ändere es zu einem Zahnrad mit 20 Zähnen.

Was glaubst du, wird sich die Zahnstange schneller oder langsamer bewegen? Probiere es aus!

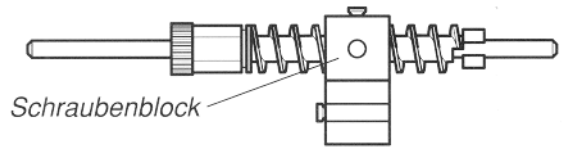
Messe, wie weit sich die Zahnstange jetzt bewegt, wenn du den Griff einmal drehst.



Zahnrad mit 20 Zähnen

b) Schraubenspindel

Drehe den Griff einmal in einem vollen Kreis. Messe wie weit sich der Schraubenblock bewegt.



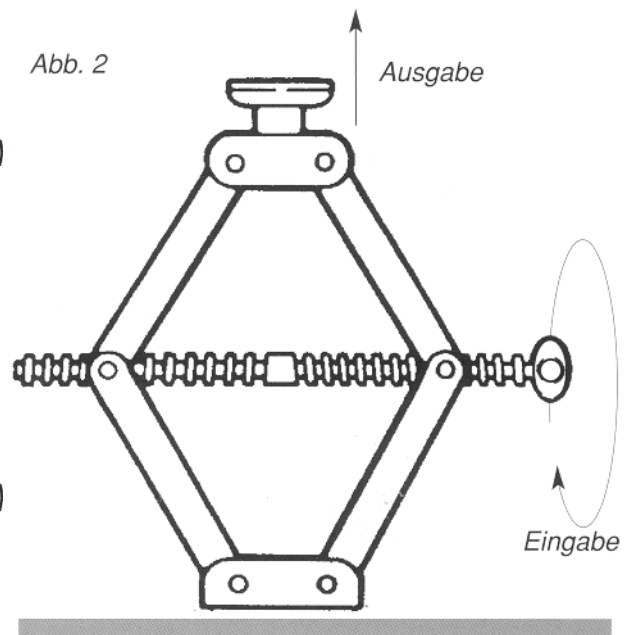
Welcher der beiden Mechanismen bewegt die Ausgabe schneller?

- 5) Bei der in Abb. 2 gezeigten Vorrichtung handelt es sich um einen Wagenheber, mit dem das Auto so weit angehoben wird, dass der Reifen gewechselt werden kann.

Schreibe zwei Gründe auf, warum deiner Meinung nach ein Schraubenspindel Mechanismus verwendet wird.

Welchen Nachteil hat die Verwendung dieses Mechanismus?

Abb. 2



3.3.4. Schlussfolgerung

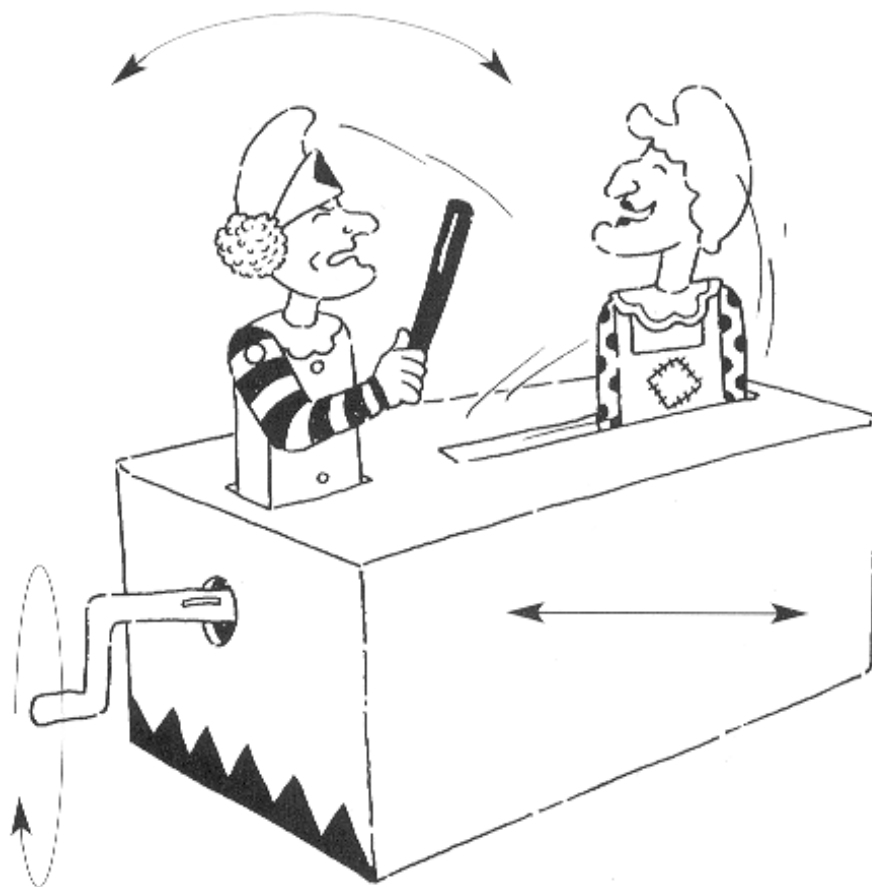
3.4. Projekt 4: Hin- und Herbewegung und schwingende Bewegung

3.4.1. Lernziel

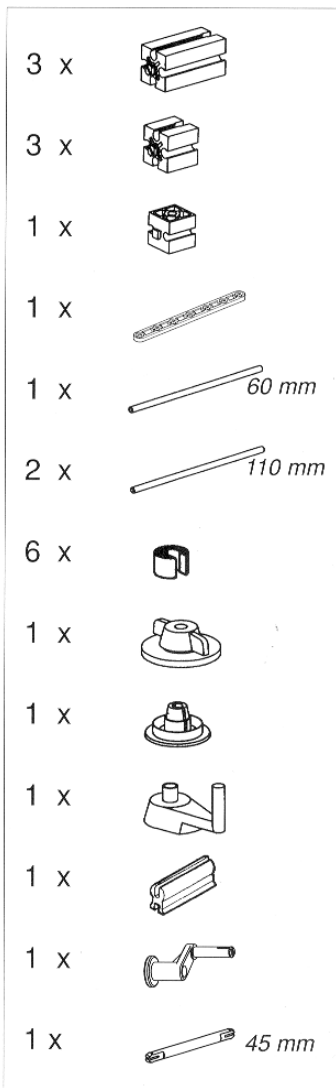
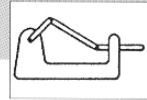
Die Schüler sollen die Möglichkeiten zur Umwandlung einer Drehbewegung in eine Hin- und Herbewegung oder eine schwingende Bewegung kennen und verstehen lernen.

3.4.2. Aufgaben

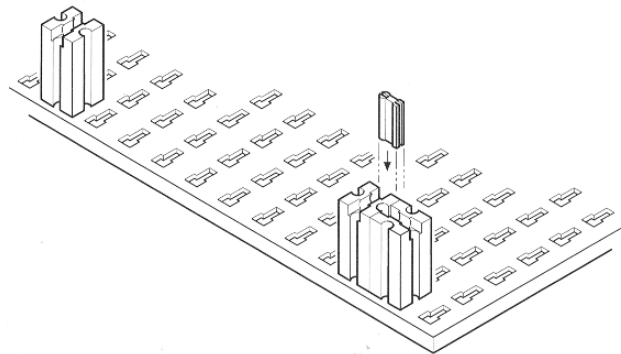
- 1) Baue die Kurbel- und Schieber-Modelle sowie die Nocken- und Stößel-Modelle auf separaten Grundplatten. Diese Mechanismen sorgen für die Bewegung, die für die Kasperle- und Gretel-Figuren erforderlich ist. Beide Mechanismen verwandeln eine Drehbewegung in eine Hin- und Herbewegung. Die folgenden Untersuchungen werden dir helfen herauszufinden, wie weit sich die Figuren bewegen können.



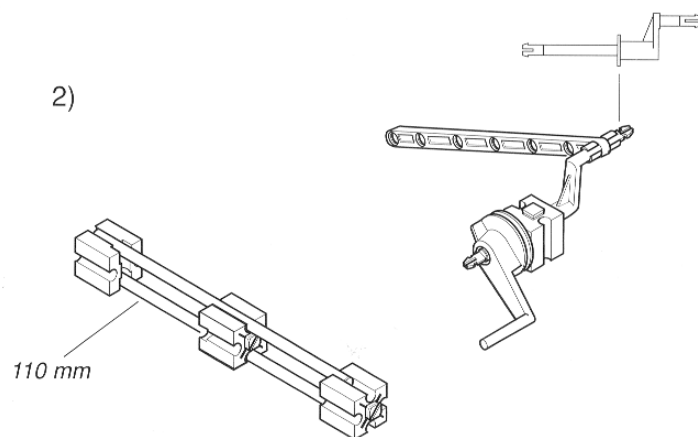
Kurbel und Schieber



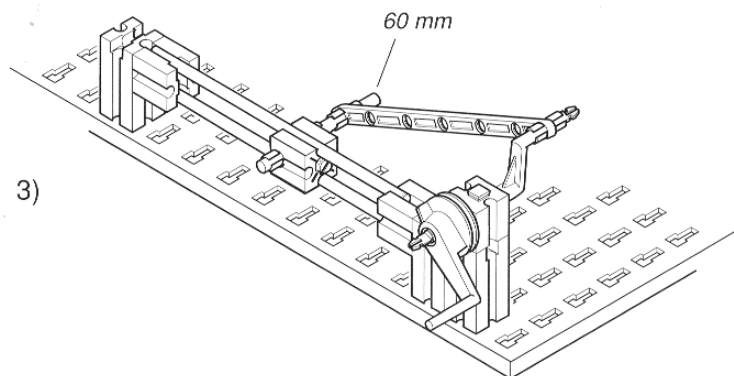
1)



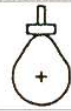
2)












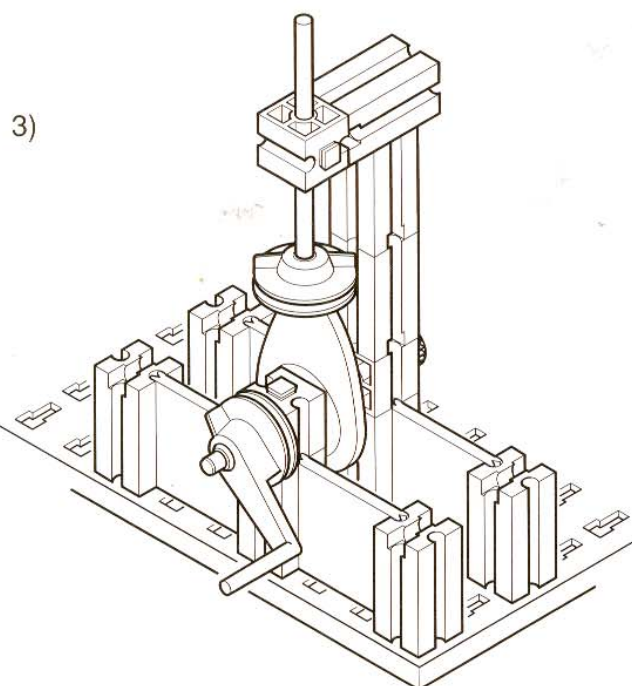
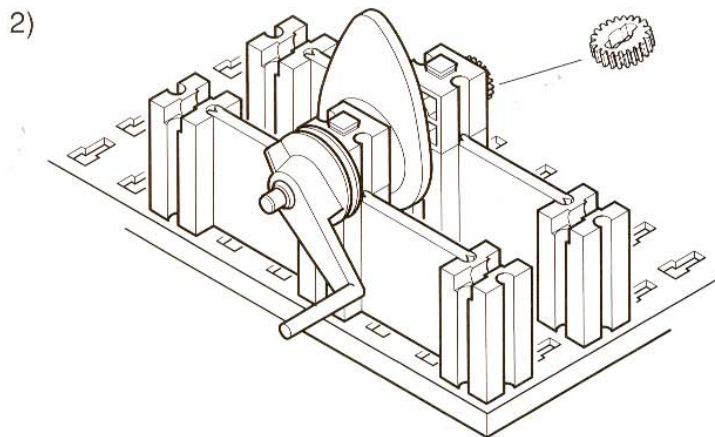
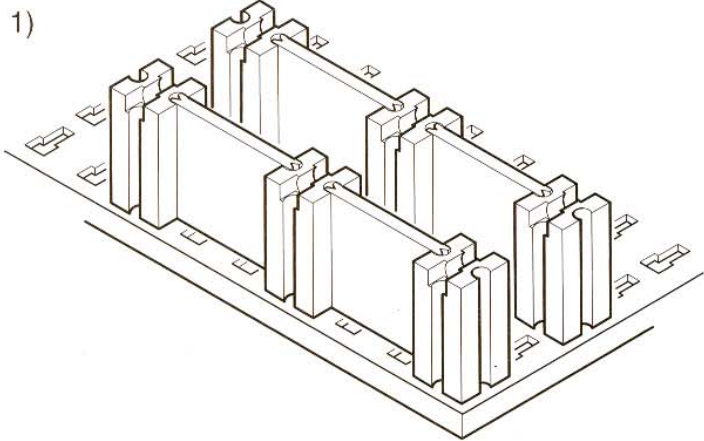
3)



Nocken und Stößel



- 9 x 
- 3 x 
- 4 x 
- 1 x 
- 2 x  80 mm
- 3 x 
- 3 x 
- 1 x 
- 1 x 

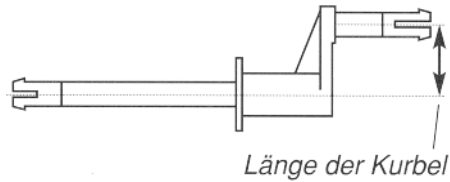


2) Antworte auf folgende Fragen um die verschiedenen Mechanismen zu analysieren.

Kurbel und Schieber

Vom Schieber zurückgelegte Entfernung = 2 x die Länge der Kurbel (siehe Abb. 1).

Abb. 1



Du kannst dies selbst überprüfen. Messe die Länge der Kurbel des Modells. Schreibe die Entfernung auf.

Messe die Entfernung, die der Schieber zurücklegt, wenn du am Griff drehst. Schreibe die Entfernung auf.

Du könntest die Gretel-Figur am Schieber befestigen, damit sie Kasperles Stock blitzschnell ausweichen kann.

a) Würdest du eine Kurbel mit einer Länge von 15 mm bauen, wie weit würde sie sich bewegen?

b) Möchtest du, dass sie die Entfernung von 24 mm zurücklegt, wie lang müsste die Kurbel sein?

Nocken und Stößel

Der Stößel ist die Stange, die sich auf und ab bewegt, wenn du den Nocken drehst. Der Stößel legt dieselbe Entfernung zurück wie der Hub des Nocken (siehe Abb. 2).

Du kannst dies selbst überprüfen. Messe den Hub des Nocken im Baukasten.



Messe, wie weit sich der Stößel bewegt, wenn du den Griff drehst.

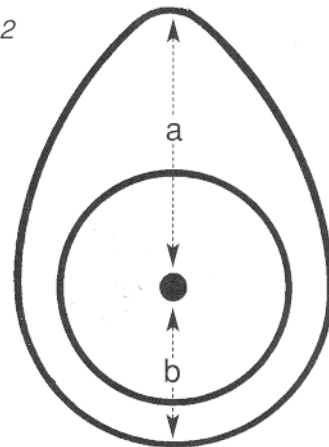


Ist Gretel am Stößel befestigt und möchtest du, dass sie sich 10 mm bewegt, wie groß müsste dann der Nockenhub sein?



Wenn du einen weiteren Nocken und Stößel am Modell anbringen würdest (siehe Abb. 3), so könnten sich sowohl Kasperle als auch Gretel zu unterschiedlichen Zeiten auf und ab bewegen. Probiere es aus!

Abb. 2



Abstand a – Abstand b = Hub des Nocken

aus!

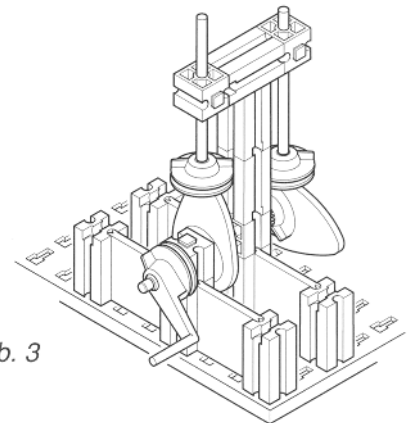


Abb. 3

3.4.3. Schlussfolgerung

3.5. Projekt 5: Eine Drehbewegung übertragen

3.5.1. Lernziel

Die Schüler sollen die Mechanismen zur Übertragung einer Drehbewegung von einer Achse auf eine andere kennen und verstehen lernen.

3.5.2. Aufgaben

- 1) Baue das **Riemenscheiben-** und **Riemen-**Modell und antworte auf die danach folgenden Fragen um die verschiedenen Mechanismen zu analysieren!

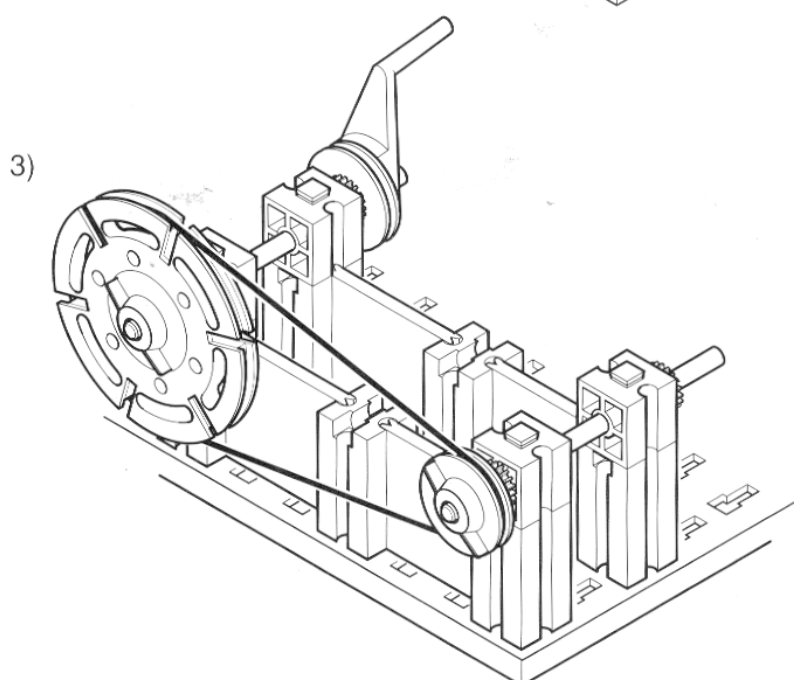
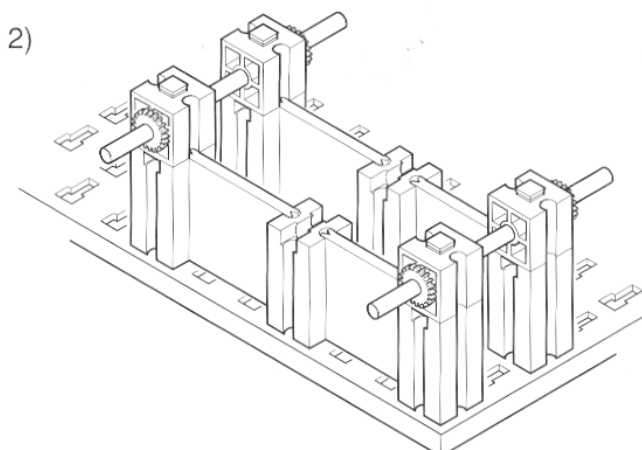
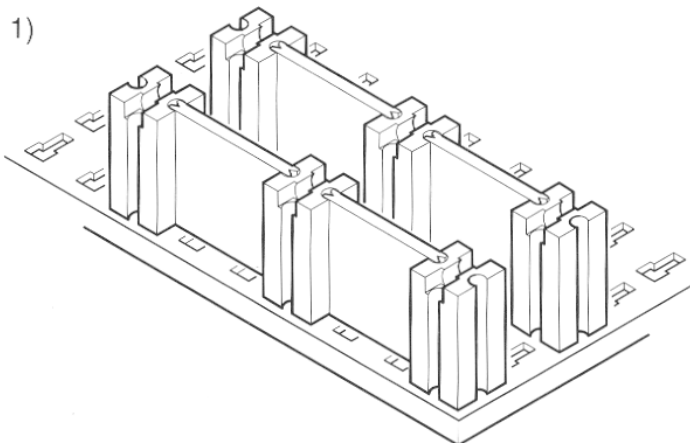
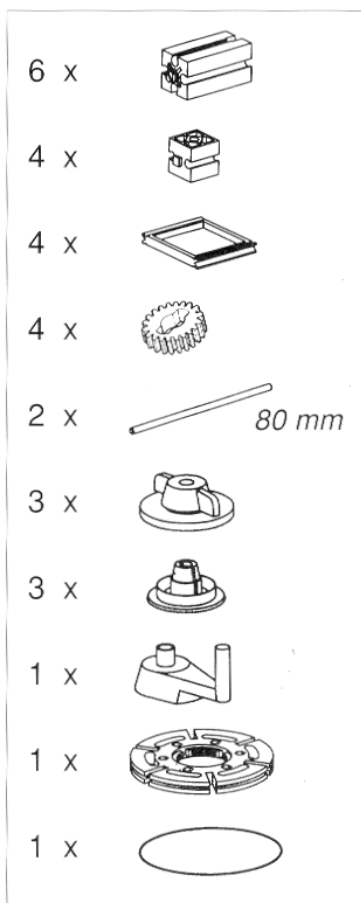
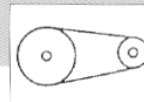
Was dreht sich schneller, die Eingabe oder die Ausgabe?



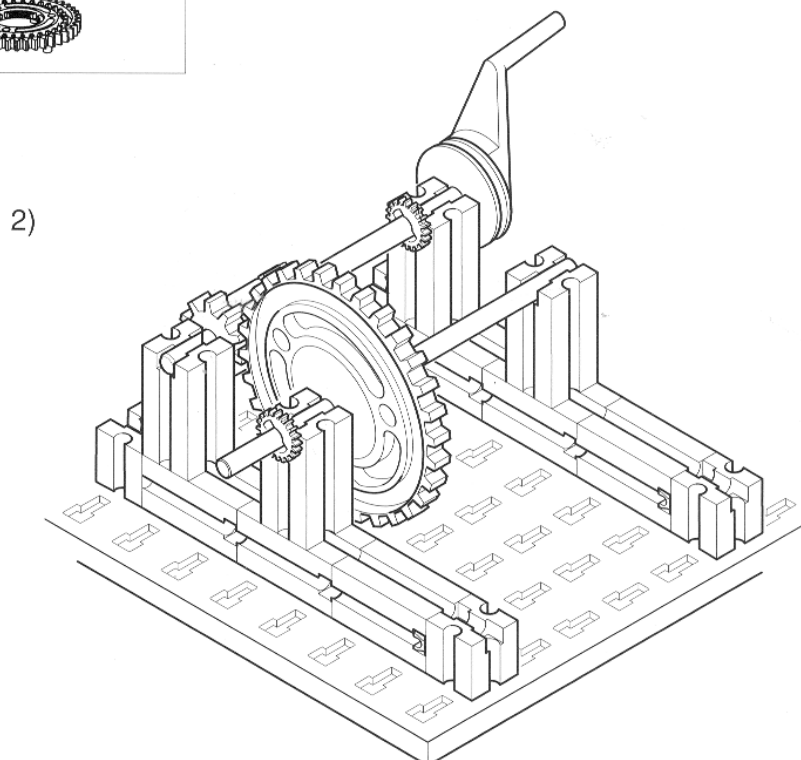
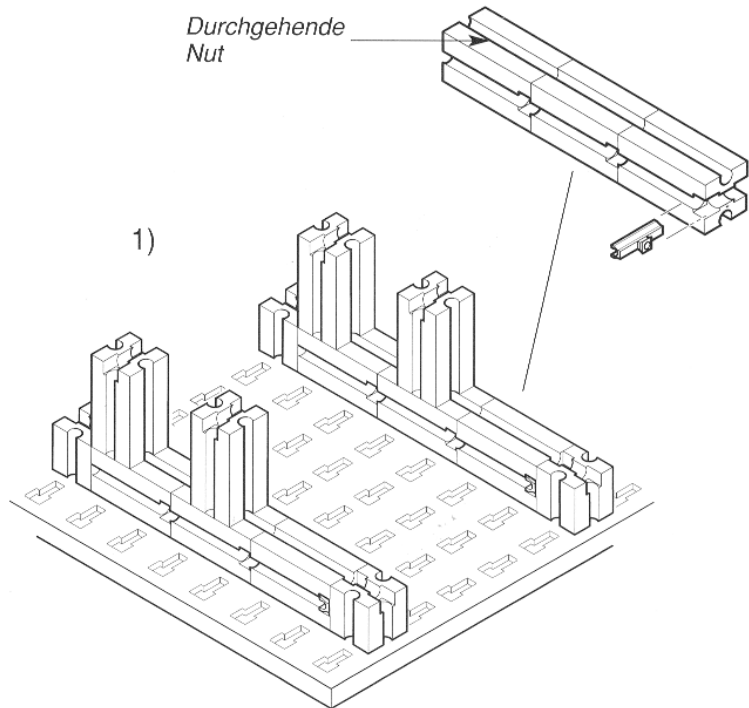
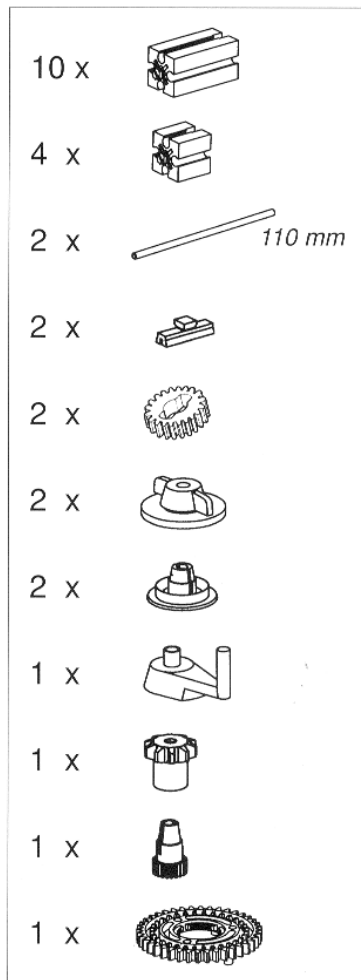
Mache den folgenden Versuch: Halte die Ausgabeachse fest und versuche sie anzuhalten, während du die Eingabe (den Griff) drehst. Ist es schwierig, die Achse anzuhalten, dann weißt du, dass die Kraftübertragung durch den Mechanismus erhöht wird. Wird die Kraftübertragung durch den Mechanismus erhöht?



Riemenscheiben und Riemen



Einfaches Getriebe



- 2) Verändere das Modell jetzt so, dass sich der Griff an derselben Achse befindet, wie die kleine Riemenscheibe.

Was dreht sich jetzt schneller, die Eingabe oder die Ausgabe?



Wird die Kraftübertragung durch den Mechanismus jetzt erhöht?



3.5.3. Schlussfolgerung

3.5.4. Aufgabe

- 1) Baue das Modell des einfachen Getriebes und antworte auf die danach folgenden Fragen um die verschiedenen Mechanismen zu analysieren!

Einfaches Getriebe

Untersuche den Mechanismus eines einfachen Getriebes, um die Antworten auf die folgenden Fragen zu finden:

Was dreht sich jetzt schneller, die Eingabe oder die Ausgabe?




Wird das Drehmoment durch den Mechanismus erhöht?



Verändere das Modell jetzt so, dass sich der Griff an derselben Achse befindet, wie das große Zahnrad.

Was dreht sich jetzt schneller, die Eingabe oder die Ausgabe?



Wird das Drehmoment durch den Mechanismus jetzt erhöht?



3.5.5. Schlussfolgerung

3.5.6. Aufgabe

1) Antworte auf folgende Fragen!

Drehen sich beide Riemenscheiben in die gleiche Richtung?

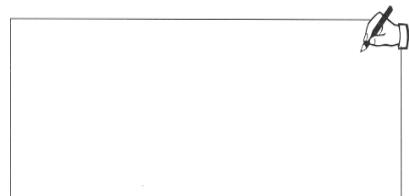


Drehen sich beide Zahnräder in die gleiche Richtung?

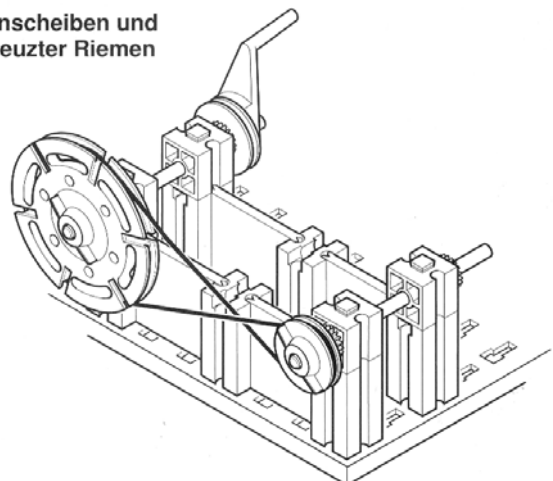


Lege den Riemen über Kreuz.

Welchen Unterschied macht dies?








Riemenscheiben und
überkreuzter Riemen



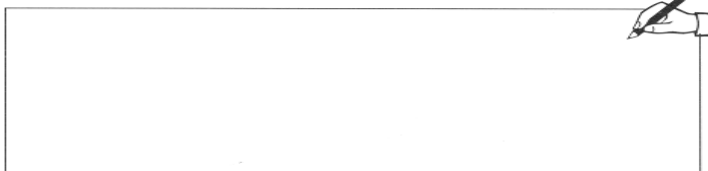
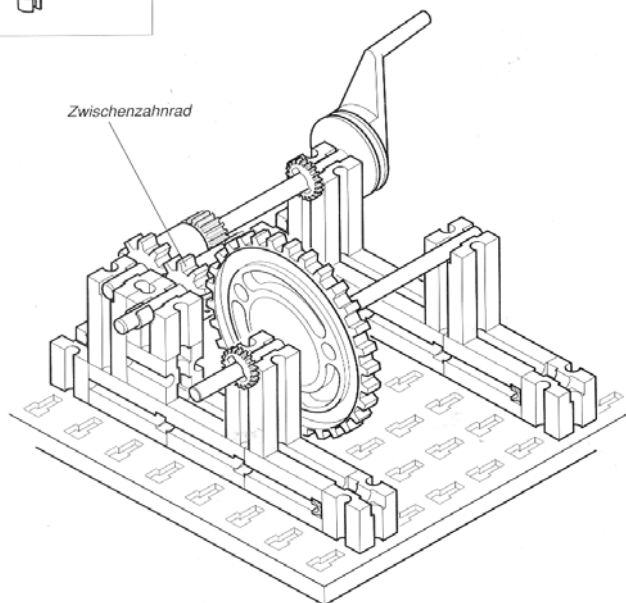
- 2) Füge ein zusätzliches kleines Zahnrad zum Getriebe hinzu. Welchen Unterschied macht dies?

Einfaches Getriebe mit Zwischenzahnrad

Zusätzliche Teile für
das Zwischenzahnrad

- 2 x 
- 1 x  60 mm
- 1 x 
- 1 x 
- 1 x 

Das **Zwischenzahnrad** beeinflusst die Richtung der Drehbewegung. Es beeinflusst jedoch nicht die Änderungen der Geschwindigkeit und des Drehmoments des Getriebes.



3.5.7. Schlussfolgerung

4. PRAKTISCHE ANWENDUNGEN MIT DEM BAUKASTEN “ELEKTRISCHE SCHALTUNGEN“

4.1. Projekt 1: Energieformen

4.1.1. Lernziel

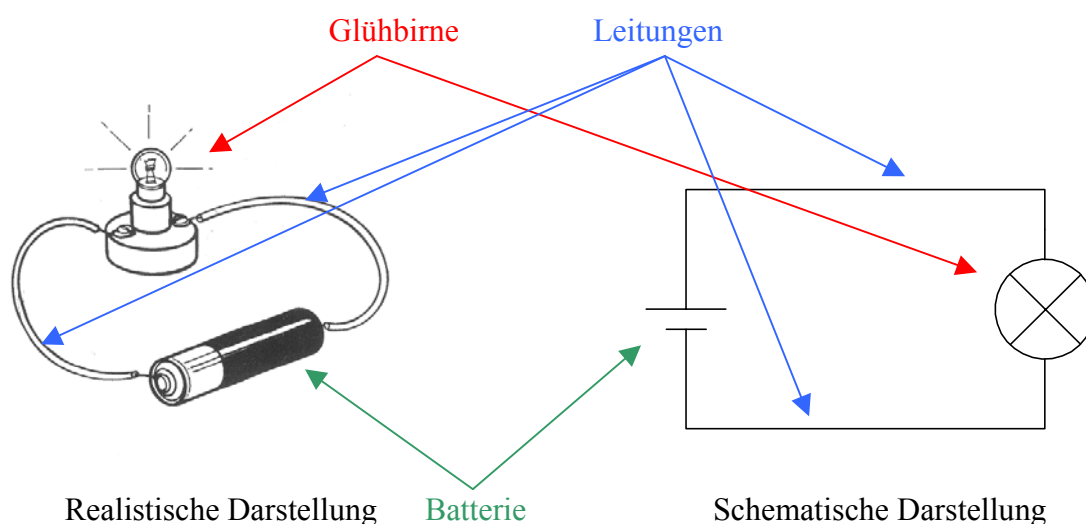
Die Schüler sollen die verschiedenen Energieformen kennen lernen.

4.1.2. Erklärungen

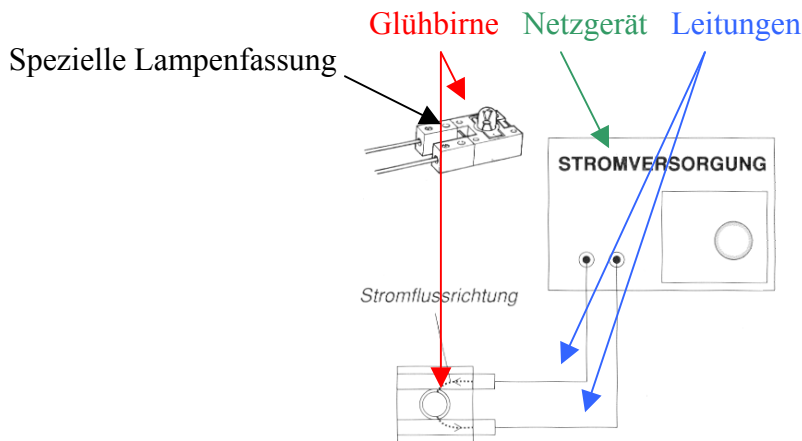
In den Abschnitten a – c wird der Stromkreis, der Strom und die Spannung, welche schon zu Beginn des Jahres im Detail durchgenommen worden sind, kurz wiederholt. Dies hauptsächlich um aufzuzeigen, mit welchen Materialien und Schemas in Anleitungen von Fischertechnik gearbeitet wird.

a) Stromkreis

Ein Stromkreis enthält eine Stromversorgung (Batterie), Verbindungsdrähte (Leitungen) und einen Verbraucher, wie z.B. eine Glühbirne.



Zur Vermeidung der leeren Batterien wird bei Fischertechnik wenn möglich mit einem Netzgerät an Stelle der Batterie gearbeitet. Der Baukasten beinhaltet auch spezielle Lampenfassungen.



Darstellung mit Fischertechnik

b) Strom

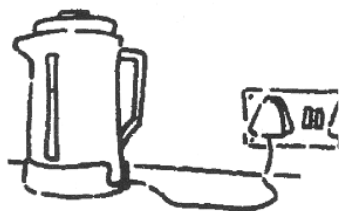
Ist der Kreis vollständig, fließt ein Strom von einem Anschlusspunkt der Stromversorgung durch die Glühlampe und zurück zum anderen Anschlusspunkt der Stromversorgung.

Ein Strom ist ein Elektronenfluss. Die Elektronen fließen durch die Leitungen. Der Strom liefert die Energie, die zum Aufleuchten der Glühlampe erforderlich ist.

Die Stromstärke wird in Ampere gemessen.

c) Spannung

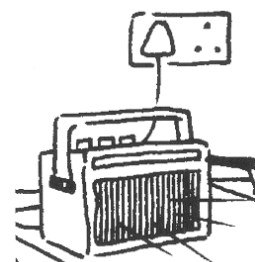
Die Spannung ist die Kraft die benötigt wird um den Strom durch die Leitungen zu treiben.



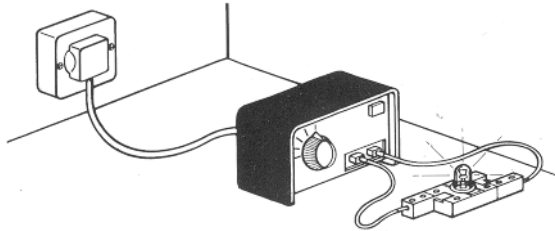
Dieser elektrische Wasserkocher benötigt eine Spannungsversorgung von 230 V. Dies ist die Standardversorgung für Haushalte und Schulen.

Dieses Radio benötigt eine 9 V-Spannungsversorgung.

Diese kann von einer Batterie geliefert werden, wenn man mobil sein will. Alternativ dazu kann das Gerät auch an das Spannungsnetz angeschlossen werden, da es einen Transformator (Umwandler) enthält, der die Spannung von 230 V auf 9 V herunter transformiert.

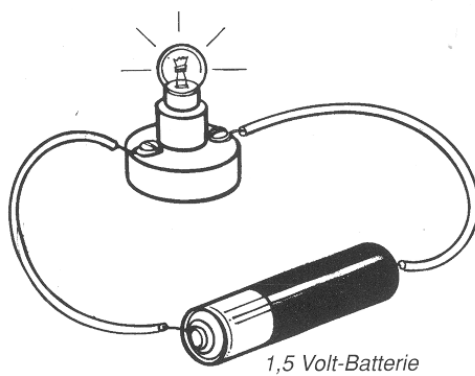


Der Motor und die Glühbirnen von Fischertechnik sind so konstruiert, dass sie eine Stromversorgung von 9 Volt benötigen. Das Netzgerät von Fischertechnik enthält eine Spannung von 9 Volt.



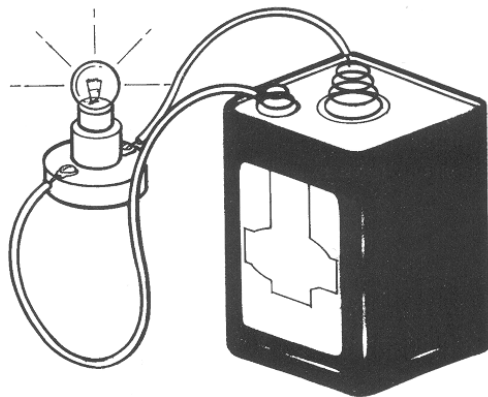
Es ist wichtig, dass die Spannung des Netzgerätes zur Nennspannung des Geräts passt. In der Abb. 4 findest du ein Beispiel für die Nennspannung von zwei Glühleuchten.

Abb. 4 9 Volt-Glühbirne



1,5 Volt-Glühbirne

9 Volt-Batterie



4.1.3. Aufgaben

- 1) Die Glühbirnenfassung von Fischertechnik enthält zwei Metallstreifen, die Elektrizität zur Glühbirne leiten. Dies bedeutet, dass man sie auf zwei verschiedene Arten in einem Stromkreis anschließen kann, wie in Abb. 2 gezeigt.

Befestige die rote Glühbirne auf der Grundplatte und baue einen Stromkreis, um sie zum Leuchten zu bringen.

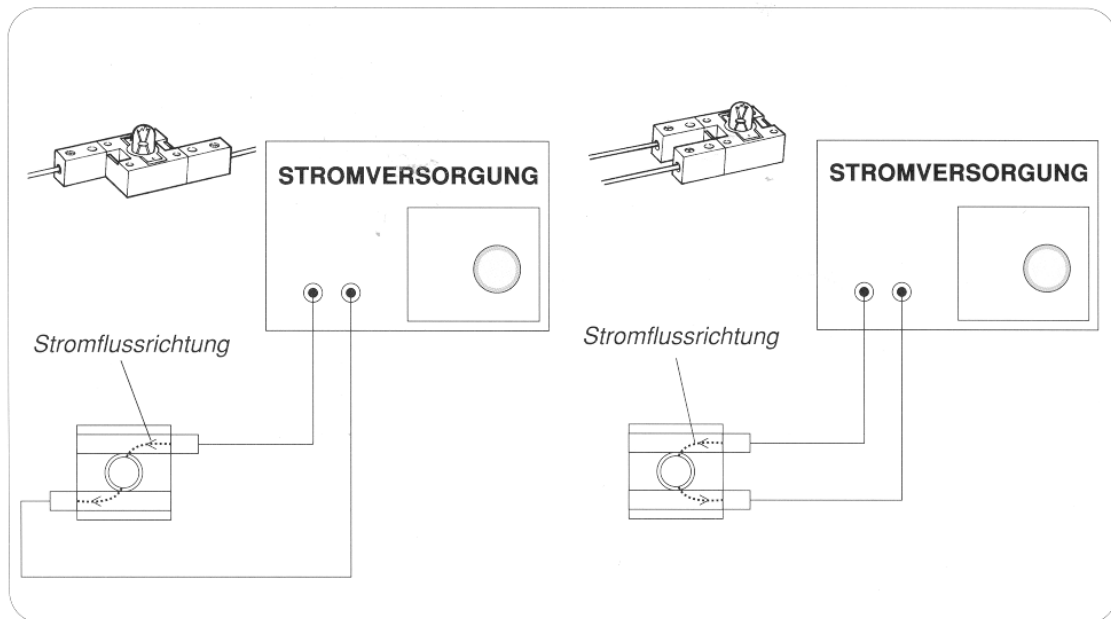
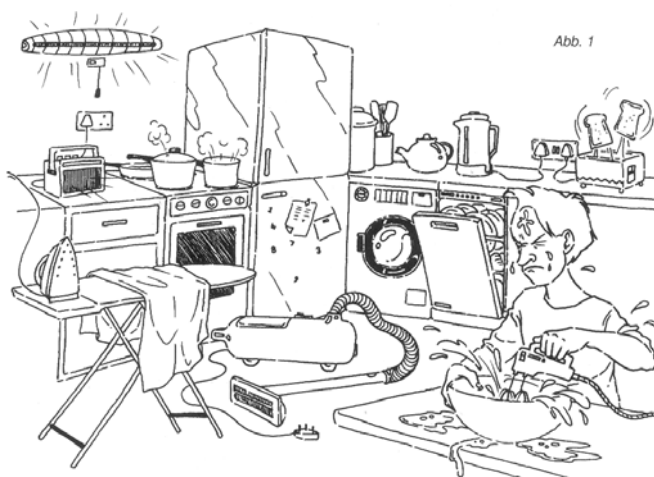


Abb. 2

Versuche, die Glühlampe auf die in Abb. 2 gezeigten zwei verschiedenen Arten anzuschließen.

- 2) Wie viele Geräte auf diesem Bild werden mit Elektrizität betrieben?

Erstelle eine Liste in der zweiten Spalte der Tabelle.



Eingabe	Elektrisches Gerät	Nutzen	Energieform
Elektrische Energie			

Bei all diesen Geräten wird eine Eingabe in Form von **elektrischer Energie** in eine Ausgabe in Form einer anderen Energie, wie z.B. **Wärme, Licht** oder **Bewegung** umgewandelt.

- 3) Trage die Ausgabe einmal als Nutzen, den wir aus diesem Gerät ziehen und einmal als Energieform mit der, der Nutzen erzeugt wird, in deine Liste ein.

4.2. Projekt 2: Stromkreise mit zwei Glühbirnen

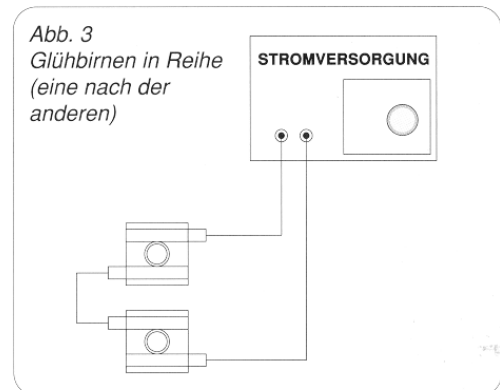
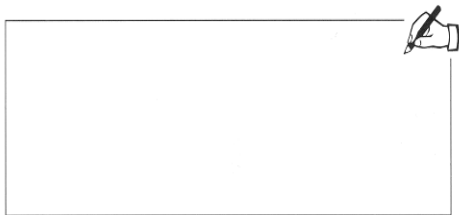
4.2.1. Lernziel

Die Schüler sollen die verschiedenen Schaltungen mit zwei Glühbirnen von Fischertechnik kennen und verstehen lernen.

4.2.2. Aufgaben

- 1) Baue den Stromkreis wie in Abb. 3 gezeigt und antworte auf die folgenden Fragen um die verschiedenen Stromkreise zu analysieren.

Was bemerkst du hinsichtlich der Helligkeit der Glühbirnen im Vergleich zum ersten Stromkreis, den du gebaut hast (Abb. 2)?

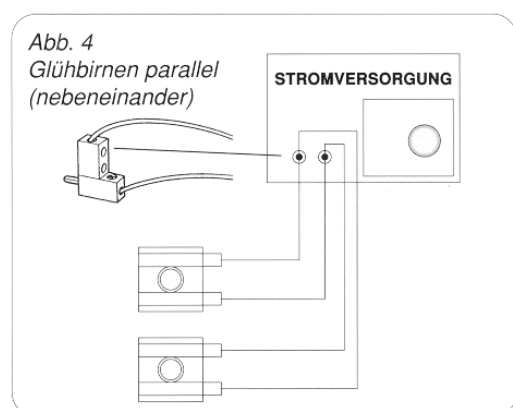


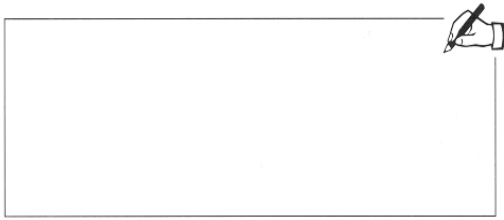
Jede Glühbirne benötigt 9 Volt der elektrischen Kraft, um genug Strom durch die Glühbirne zu schieben, damit sie vollständig aufleuchtet. In diesem Stromkreis werden die 9 Volt, die von der Stromversorgung zur Verfügung gestellt werden, zwischen den beiden Glühbirnen geteilt, so dass jede nur teilweise aufleuchtet.

Die Glühbirnen sind in Reihe geschaltet.

- 2) Baue den Stromkreis wie in Abb. 4 gezeigt und antworte auf die folgenden Fragen um die verschiedenen Stromkreise zu analysieren.

Jetzt sind die Glühbirnen parallel geschaltet. Was bemerkst du hinsichtlich der Helligkeit der Glühbirnen im Vergleich zum Stromkreis in Reihe?

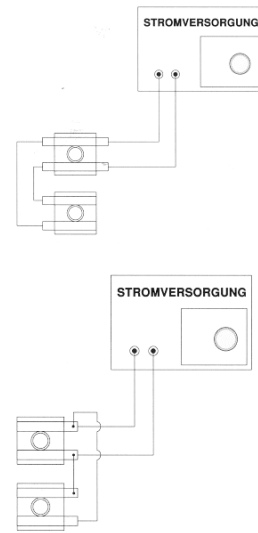




Jede Glühbirnen, wird jetzt von 9 Volt versorgt, so dass sie beide vollständig aufleuchten. Die Stromversorgung liefert jetzt doppelt so viel Strom als es der Fall war, als die Glühbirnen in Reihe geschaltet waren.

Die Glühbirnen sind parallel geschaltet

Zwei weitere Methoden, fischertechnik-Glühbirnen parallel zu schalten. Probiert sie aus!



4.3. Projekt 3: Stromkreis mit Schalter

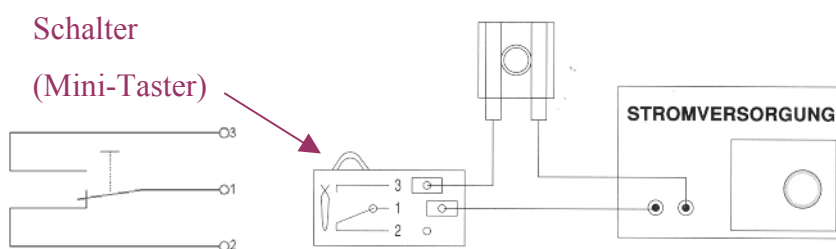
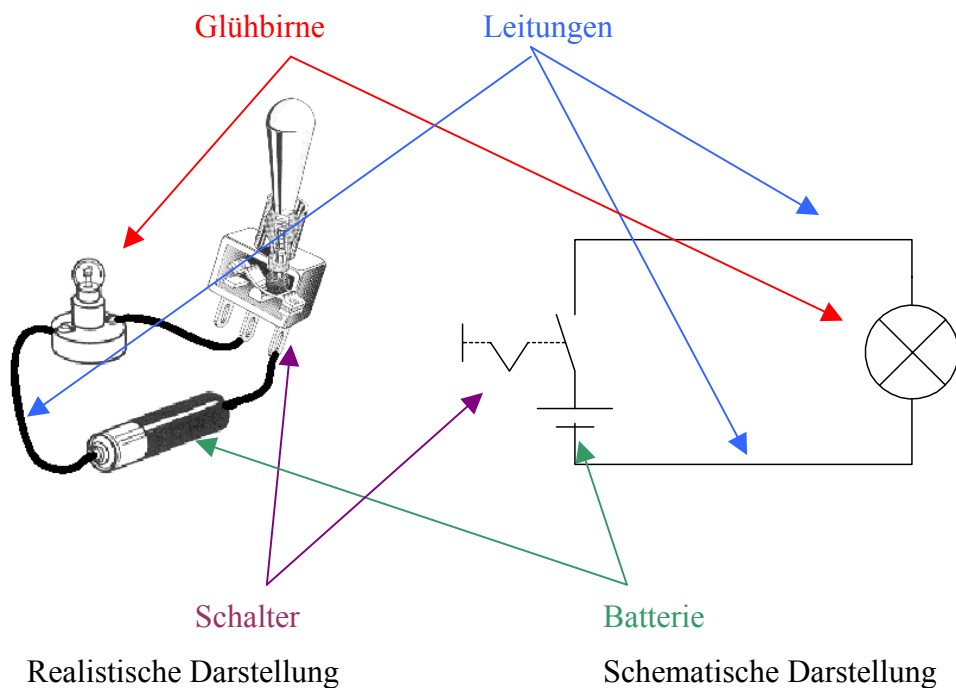
4.3.1. Lernziel

Die Schüler sollen verschiedene Schaltungen mit Fischertechnik Schaltern kennen und verstehen lernen.

4.3.2. Erklärung

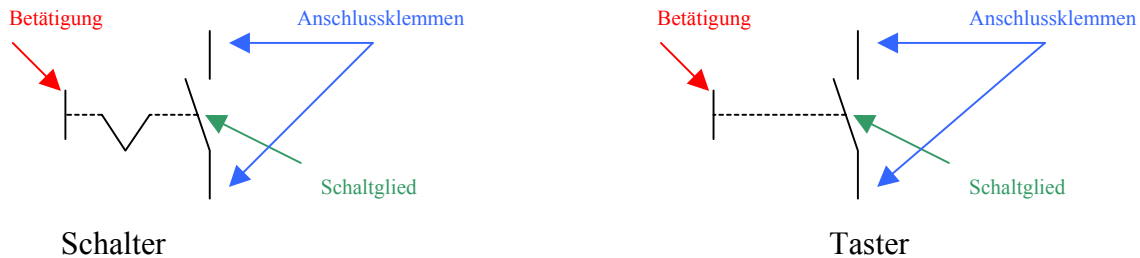
Um ein Ausgangsgerät, wie z.B. eine Glühbirne ein- und auszuschalten, muss ein vollständiger Stromkreis geschlossen werden.

Ein Schalter wird verwendet um einen Stromkreis zu schließen oder zu unterbrechen und damit zu steuern, ob das Ausgabegerät ein- oder ausgeschaltet ist.



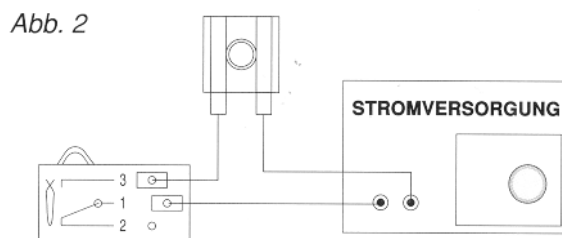
Darstellung mit Fischertechnik

Ein Taster ist ein Schalter, der beim Loslassen von selbst in seine Ausgangslage zurückkehrt.



4.3.3. Aufgabe

- 1) Baue den in Abb. 2 gezeigten Stromkreis auf und antworte auf die folgenden Fragen um den Mini-Taster zu analysieren!



Leuchtet die Glühlampe auf, wenn du den Stromkreis verbindest?

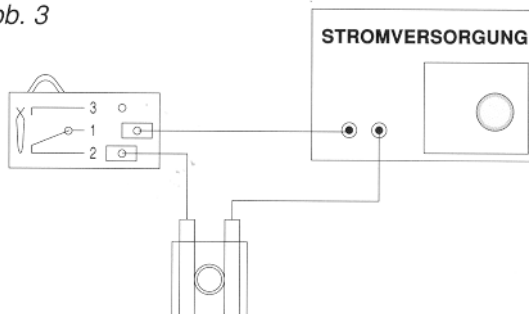
Leuchtet die Glühlampe auf, wenn du den Taster drückst?

4.3.4. Schlussfolgerung

4.3.5. Aufgabe

- 1) Baue den in Abb. 3 gezeigten Stromkreis auf und antworte auf die folgenden Fragen um den Mini-Taster zu analysieren!

Abb. 3



Leuchtet die Glühbirne auf, wenn du den Stromkreis zusammenschaltest?

Leuchtet die Glühbirne auf, wenn du den Mini-Taster drückst?

4.3.6. Schlussfolgerung

4.4. Projekt 4: Elektrische und mechanische Systeme

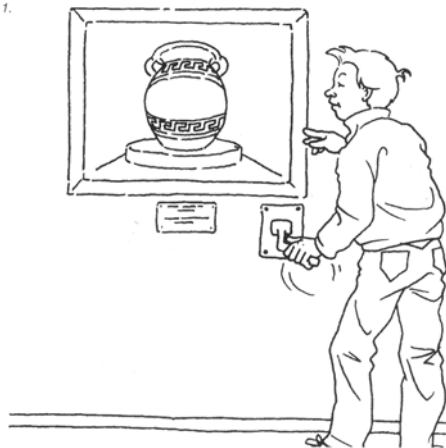
4.4.1. Lernziel

Die Schüler sollen elektrische und mechanische Systeme kombinieren können.

4.4.2. Erklärungen

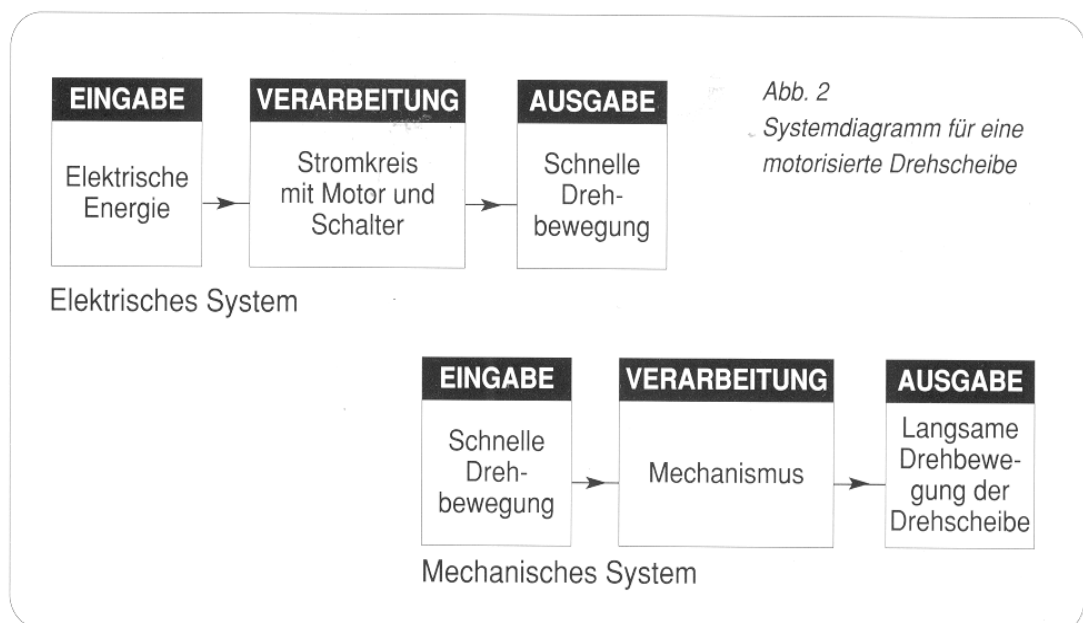
Ein elektrischer Motor könnte verwendet werden, um die Drehscheibe des in Abb. 1 gezeigten Museumsschaukastens zu drehen.

Abb. 1.



Die Drehscheibe muss sich langsam drehen. Ein Schalter würde es den Besuchern ermöglichen die Drehscheibe zu drehen, wenn sie dies möchten.

Das komplette Drehscheibensystem besteht aus zwei Untersystemen, wie in Abb. 2 zu sehen ist.

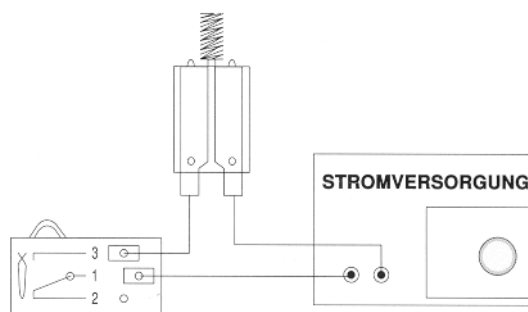


Der Mechanismus muss die Drehbewegung vom Motor auf die Drehscheibe übertragen und sie ändern, indem er sie verlangsamt. Die einfachste Art dies zu tun, ist die Verwendung des Motorgetriebes.

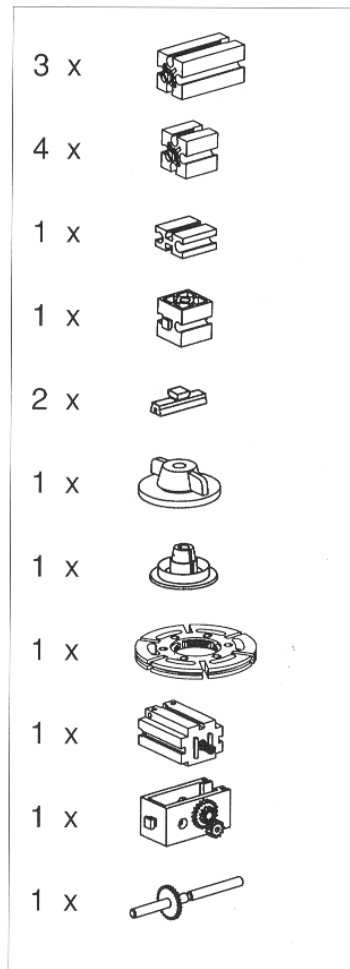
4.4.3. Aufgaben

- 1) Baue Modell 1 auf dem Konstruktionsblatt für die motorisierte Drehscheibe. Baue anschließend den in Abb. 3 gezeigten Stromkreis um die Drehscheibe zu steuern. Erprobe das Modell um festzustellen, ob es richtig funktioniert.

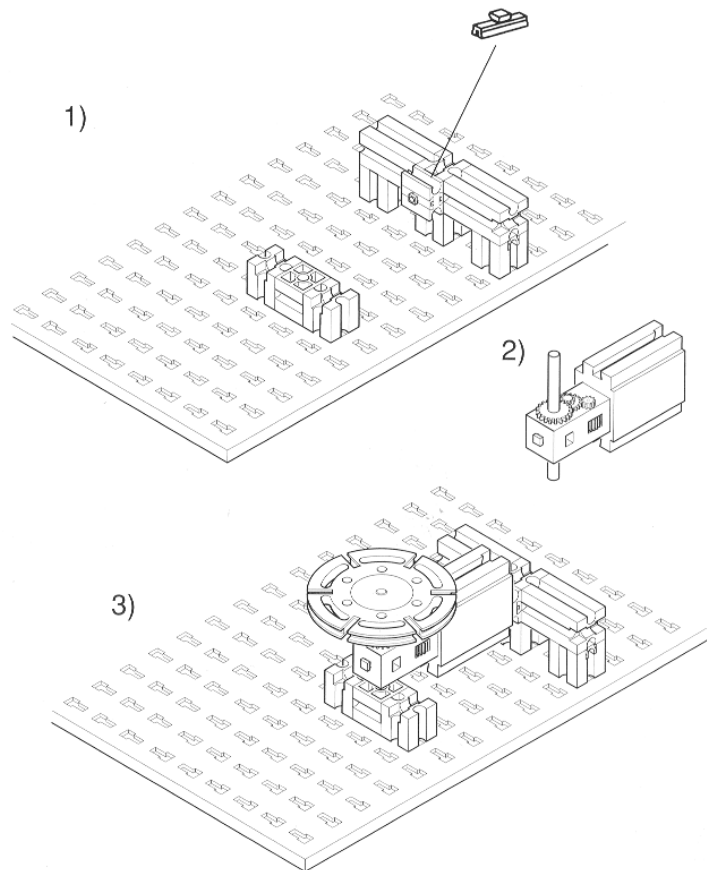
Abb. 3



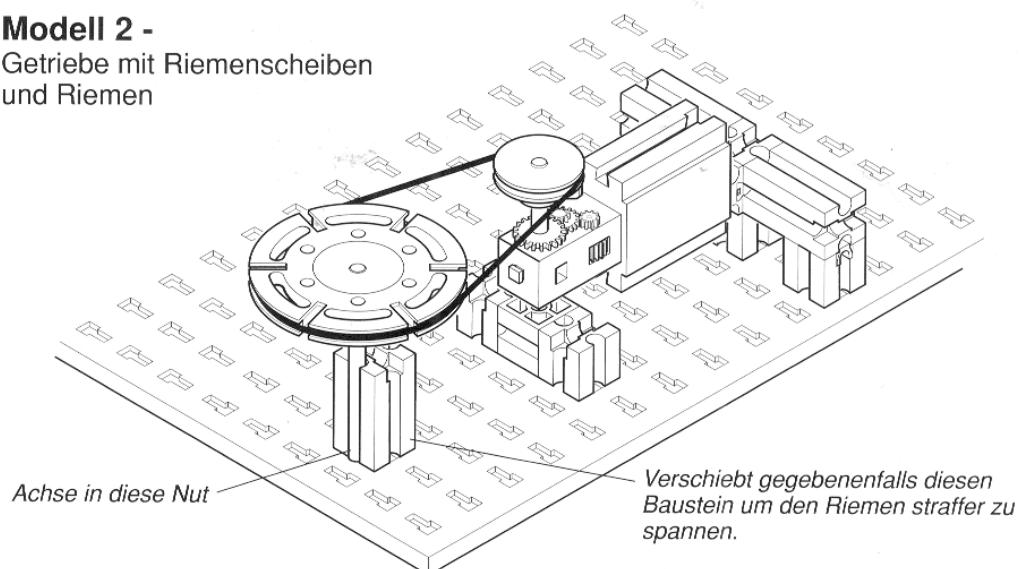
Motorisierte Drehscheibe



Modell 1

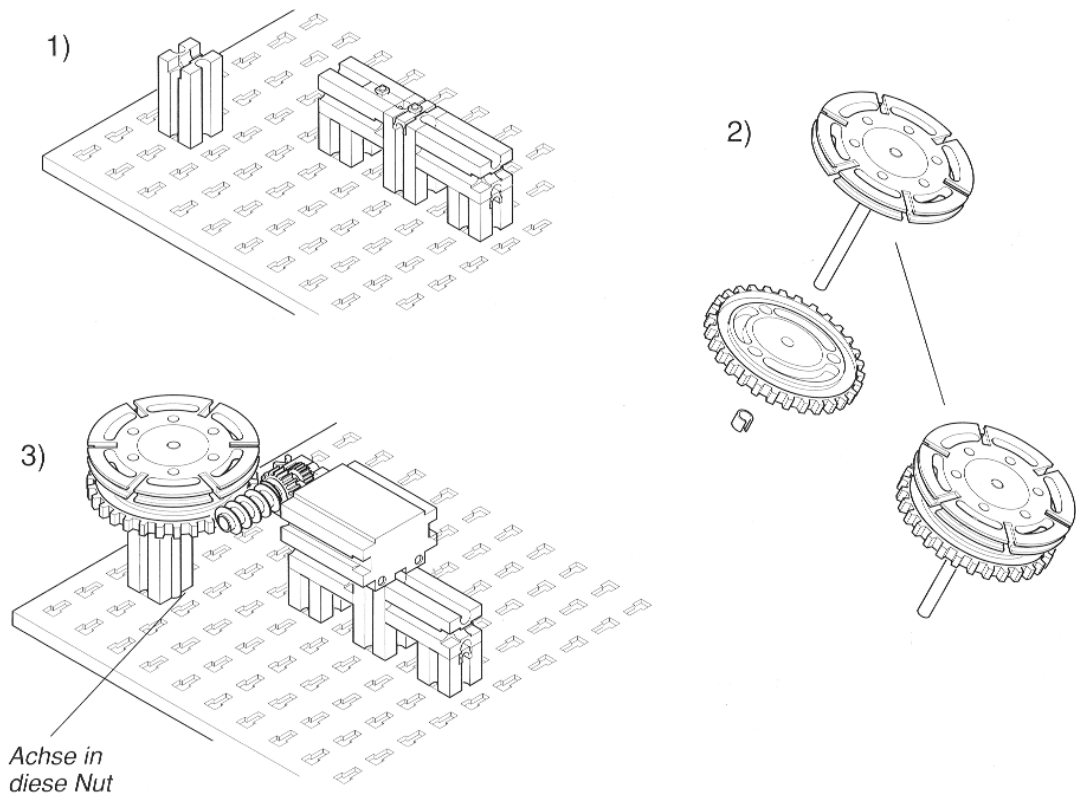


Modell 2 - Getriebe mit Riemenscheiben und Riemen



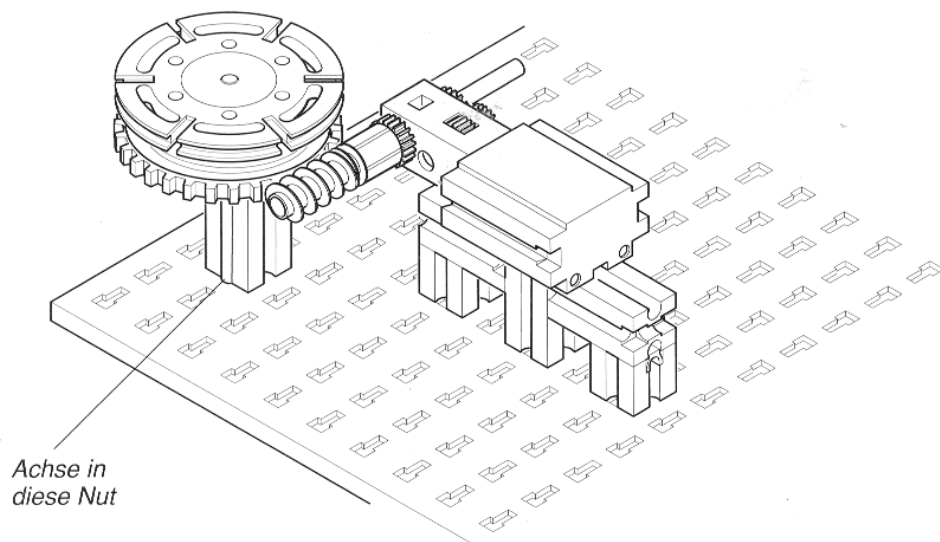
Modell 3

Schneckenradgetriebe



Modell 4

Motorgetriebe plus Schneckenradgetriebe

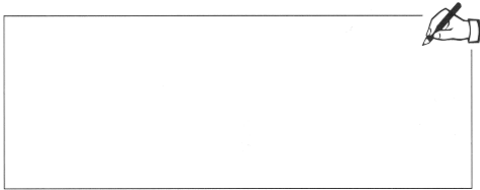


2) Einen geeigneten Mechanismus auswählen.

Das Getriebe reduziert die Geschwindigkeit der vom Motor gelieferten Bewegung, jedoch nicht sehr stark. Auf dem Konstruktionsblatt sind drei verschiedene Mechanismen zum ausprobieren. Baue nacheinander jeden Mechanismus auf und erprobe ihn, um die Geschwindigkeit der Drehscheibe festzustellen.

Wenn du alle Mechanismen ausprobiert hast, wähle den Mechanismus aus, der deiner Meinung nach am besten für das Drehscheibensystem geeignet ist.

Welcher Mechanismus ist am besten für das Drehscheibensystem geeignet?



4.4.4. Schlussfolgerung

4.5. Projekt 5: Schaltkreise

4.5.1. Lernziel

Die Schüler sollen lernen die verschiedenen Fischertechnik Bauteile zu den gewünschten Stromkreisen zusammenzubauen.

4.5.2. Erklärungen

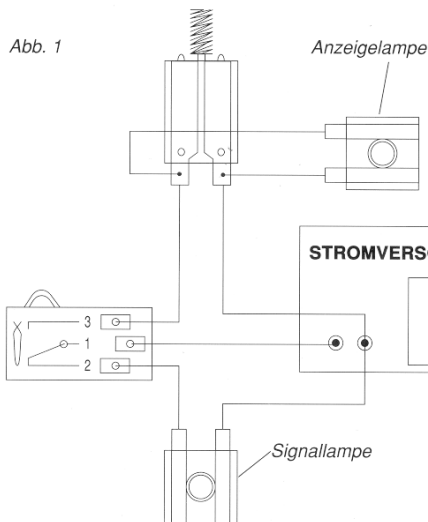
Verwende diese Methode zum Bauen der verschiedenen Stromkreise, die auf diesem Blatt abgebildet sind:

- a) Baue deinen vorhandenen Stromkreis auseinander und legt die Leitungen wieder in den Kasten zurück, bevor du mit einem neuen Stromkreis beginnst.
- b) Beginne jeden Stromkreis von der roten Klemme der Stromversorgung.
- c) Arbeite systematisch und ende, indem du die Leitung wieder an die schwarze Klemme der Stromversorgung anschließt.

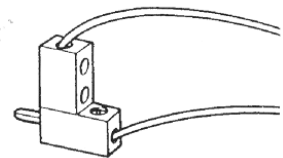
4.5.3. Aufgaben

1) Stromkreis 1

Dieser Stromkreis beinhaltet eine Glühbirne, die eine Anzeige "Hier drücken" neben dem Schalter zum Aufleuchten bringt. Wenn du den Schalter drückst, erlischt diese Glühbirne, der Motor dreht sich und die Anzeigelampe an der Drehscheibe leuchtet auf .



Der Stromkreis ist in Abb. 1 zu sehen. In diesem Stromkreis wird der Mini-Taster als "Zweiwegtaster" verwendet. Baue den Stromkreis, erprobe ihn und trage anschließend die drei fehlenden Symbole in den Schaltplan ein (Abb. 2).



Denkt daran, dass ihr Steckverbinder aufeinander stecken könnt

Abb.2
Trage die 3 fehlenden
Symbole ein

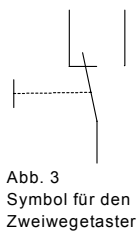
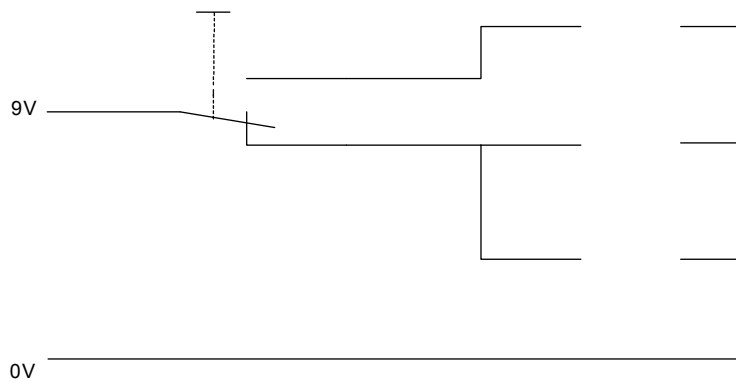


Abb. 3
Symbol für den
Zweiwegtaster

2) Stromkreis 2

Dieser Stromkreis ermöglicht es dem Museumspersonal zu steuern, ob die Besucher die Drehscheibe betätigen können oder nicht. Dabei wird der Polwendeswitcher als Ein-Aus-Schalter verwendet (siehe Abb. 4).

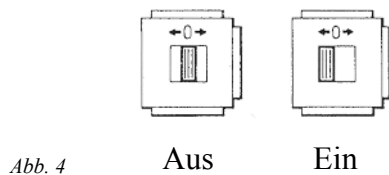


Abb. 4

Ein Ein-Aus-Schalter unterscheidet sich von einem Taster, da er in jedem der beiden Zustände belassen werden kann. Ein Taster springt in seinen normalen Zustand zurück, wenn du ihn los lässt.

In Abb. 5 ist der Stromkreis zu sehen. Schalter und Taster sind in Reihe geschaltet, was bedeutet, dass sie beide eingeschaltet sein müssen, damit der Stromkreis geschlossen wird.

Baue den Stromkreis. Erprobe ihn und trage anschließend die beiden fehlenden Symbole

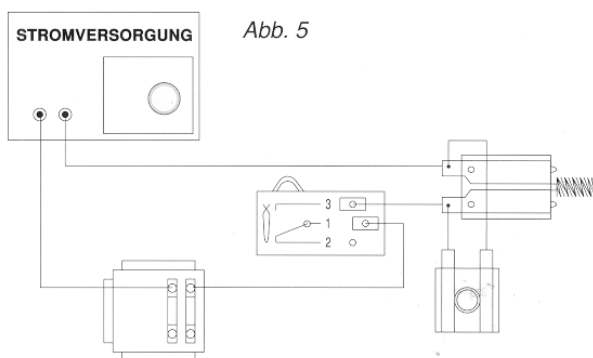


Abb. 5

in den Schaltplan ein (Abb. 6).

Entscheide dich, wer den Schalter und wer den Taster verwenden soll. Kennzeichne die Schalter dann auf dem Schaltplan als "Personal-Schalter" und als "Besucher-Schalter".

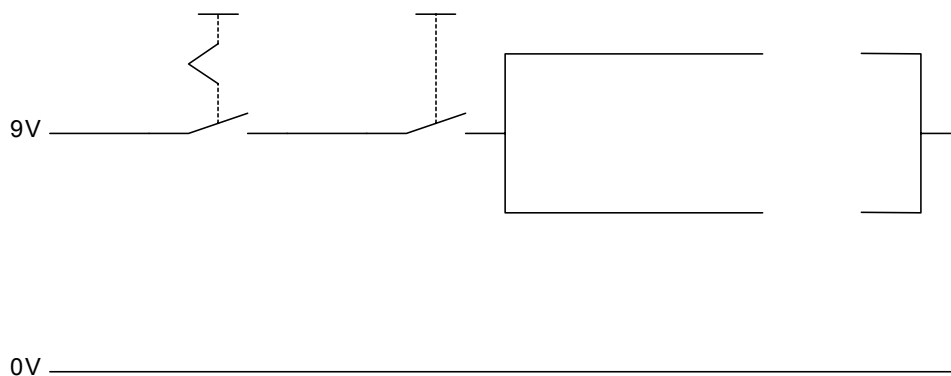
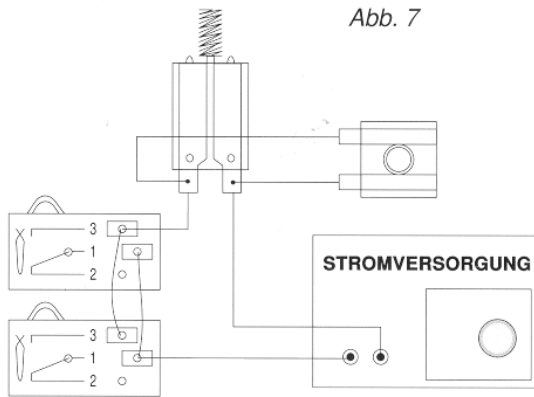


Abb 6. Trage die beiden fehlenden Symbole ein und kennzeichne die Schalter

3) Stromkreis 3

Dieser Stromkreis ermöglicht es den Besuchern, die Drehscheibe von verschiedenen Seiten des Schaukastens aus zu steuern.



In Abb. 7 ist der Stromkreis zu sehen. Die beiden Mini-Taster sind parallel geschaltet, was bedeutet, dass der Stromkreis geschlossen wird, wenn einer der Taster gedrückt wird.

Baue den Stromkreis. Erprobe ihn und trage anschließend die fehlenden Symbole in den Schaltplan ein (Abb. 8).

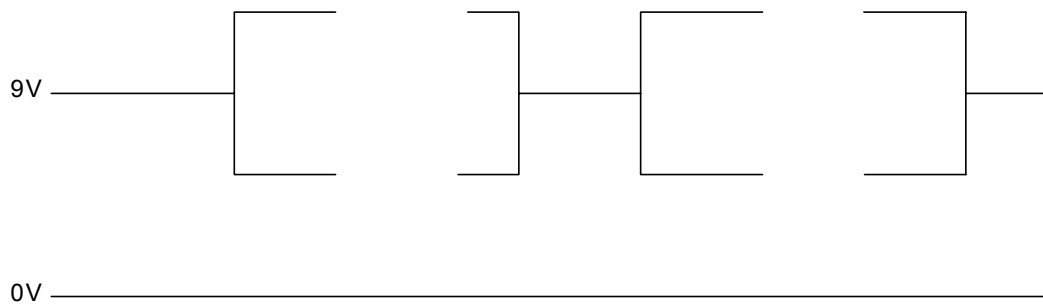


Abb. 8 Trage die 4 fehlenden Symbole ein

4.6. Projekt 6: Einen Motor schalten
















4.6.1. Lernziel

Die Schüler sollen lernen wie bei einem Fischertechnik Motor die Drehrichtung geändert wird.

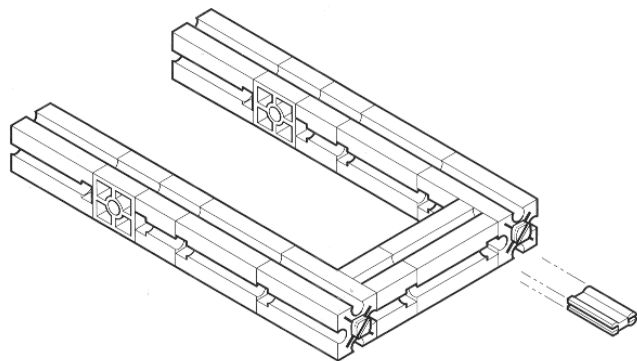
4.6.2. Aufgaben

- 1) Baue das auf dem Konstruktionsblatt für das motorisierte Fahrgestell gezeigte Modell. Verwende den Mini-Taster, um das Modell so zu steuern, dass es vom Motor angetrieben wird, wenn man den Taster drückt.

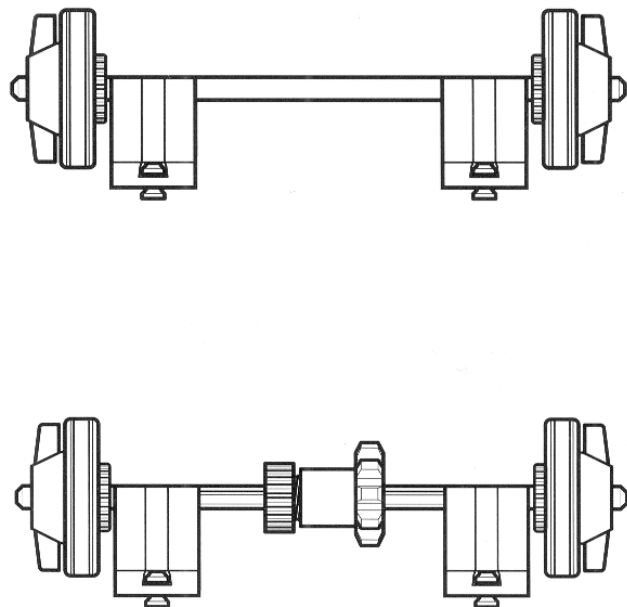
Motorisiertes Fahrgestell

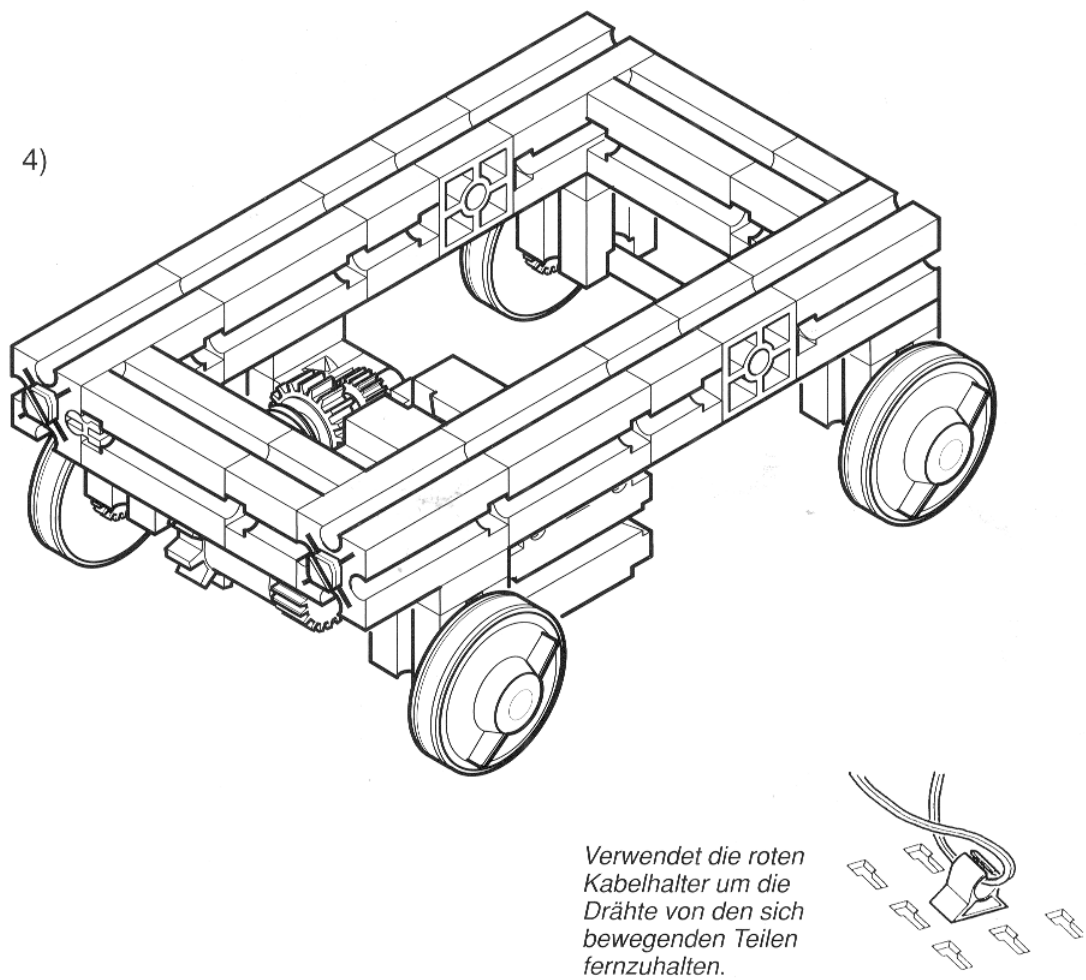
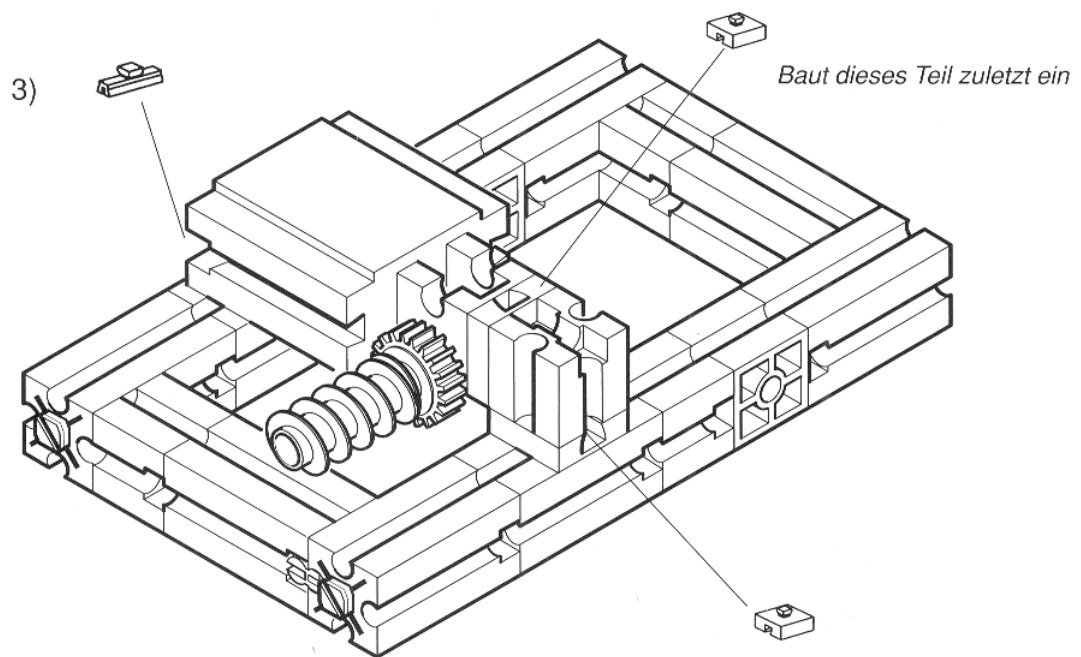
8 x	
9 x	
6 x	
2 x	
1 x	
4 x	
2 x	 110 mm
4 x	
4 x	
4 x	
1 x	
1 x	
1 x	
1 x	
2 x	

1)



2)





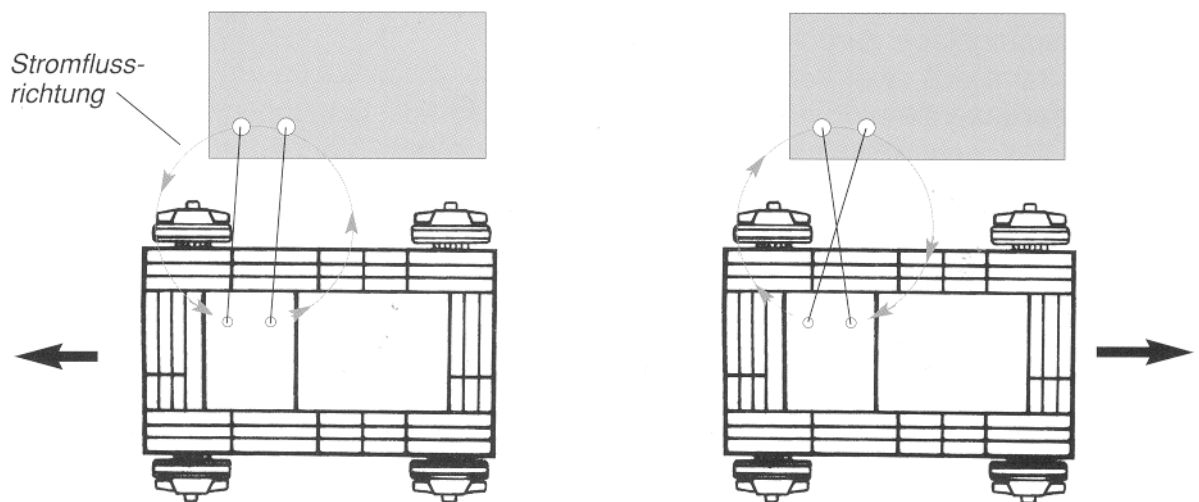
In welche Richtung bewegt sich der Wagen?



Du kannst ihn in die entgegengesetzte Richtung fahren lassen, indem du die Anschlüsse zum Motor wie in Abb. 5 gezeigt umpolst. Probiere es aus!

Abb. 5

Der Motor ändert die Richtung, wenn ihr die Richtung des Stromflusses umkehrt.



Der Polwendeschalter von Fischertechnik ermöglicht die Richtung des Stroms im Motorstromkreis umzukehren. Baue den in der Abbildung gezeigten Stromkreis. Verwende den Polwendeschalter, um den Wagen in beide Richtungen zu steuern.

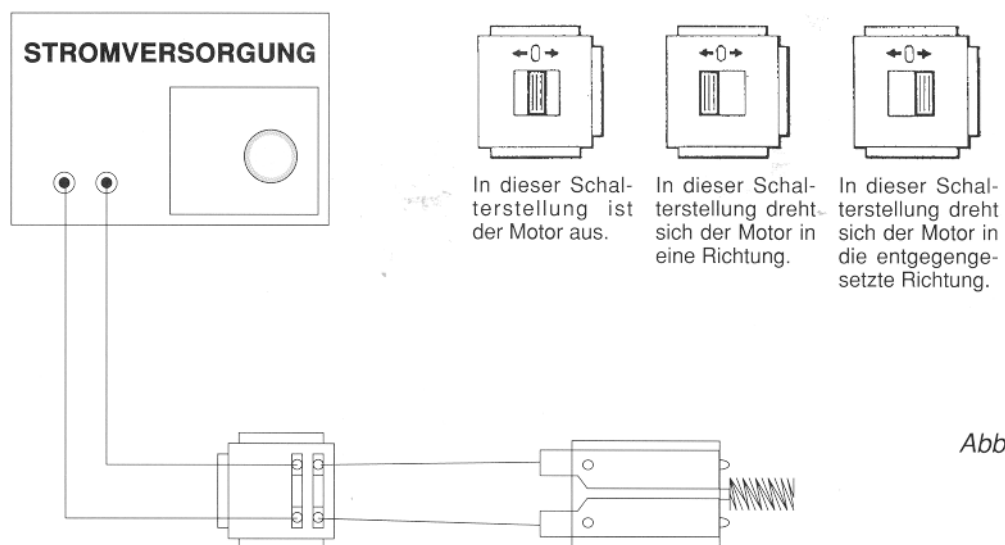


Abb. 6

- 2) Baue die roten und grünen Glühbirnen an den Wagen. Schaltet den Stromkreis so zusammen, dass beide Glühbirnen jedes mal dann voll aufleuchten, wenn sich der Wagen bewegt.

Werden die Glühbirnen durch die Änderung der Richtung des Stromes beeinflusst?

Nein.

Der Strom in einem vollständigen Kreis fließt immer vom Pluspol der Spannungsquelle durch den Verbraucher zum Minuspol der Spannungsquelle. Schließt man einen Fischertechnik Motor an so kann es sein, dass dieser sich in die falsche Richtung bewegt, da am Motor keine Bezeichnungen für Plus- oder Minuspol angebracht sind. Was muss man jetzt tun um die Drehrichtung des Motors zu ändern?

Man muss die Anschlüsse am Motor um tauschen.

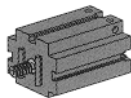
Bei Fischertechnik-Leitungen ist die Plusleitung immer die Rote und die Minusleitung immer die Grüne.

4.6.3. Schlussfolgerung

5. PRAKTISCHE ANWENDUNGEN MIT DEM BAUKASTEN “COMPUTING STARTER“

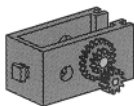
5.1. Die wichtigsten Bauteile

5.1.1. Motor



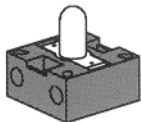
Dieser Motor treibt die Fischertechnik Modelle an. Er wird mit einer Spannung von 9 Volt DC (Gleichspannung) betrieben. Die maximale Leistung liegt bei ca. 1,1 Watt bei einer Drehzahl von 7000 Umdrehungen pro Minute.

5.1.2. Getriebe



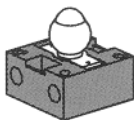
Auf den Motor wird ein Getriebe gesteckt, das die Drehzahl heruntersetzt. Die Untersetzung beträgt einschließlich der Motorschnecke und dem Zahnrad mit der Abtriebswelle 64,8 : 1.

5.1.3. Kugellampe



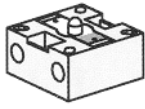
Das ist eine gewöhnliche Glühlampe für eine Spannung von 9 V DC und einem Stromverbrauch von ca. 0,1 A.

5.1.4. Linsenlampe



In diese Lampe ist eine Linse eingearbeitet, die das Licht bündelt. Sie sieht der Kugellampe sehr ähnlich. Man muss aufpassen, dass man sie nicht verwechselst. Die Linsenlampe benötigt man zum Bauen einer Lichtschranke.

5.1.5. Fototransistor

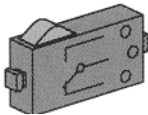


Man bezeichnet den Fototransistor auch als Helligkeitssensor. Das ist ein Fühler der auf Helligkeit reagiert.

Er bildet bei einer Lichtschranke das Gegenstück zur Linse Lampe. Bei großer Helligkeit, also wenn der Transistor von der Linse Lampe angestrahlt wird, leitet er Strom. Wird der Lichtstrahl unterbrochen, leitet der Transistor keinen Strom.

Achtung: Beim Anschluss des Fototransistors an die Stromversorgung muss man auf die richtige Polung achten: Rot = Plus.

5.1.6. Taster



Der Taster wird auch Berührungssensor genannt. Beim Betätigen des roten Knopfes wird mechanisch ein Schalter umgelegt, es fließt Strom zwischen den Kontakten 1 (mittlerer Kontakt) und 3. Gleichzeitig wird der Kontakt zwischen den Anschlüssen 1 und 2 unterbrochen. So kann man den Taster auf zwei verschiedene Arten verwenden.

Als "Schließer": Kontakte 1 und 3 werden angeschlossen.

Taster gedrückt: Es fließt Strom. Taster nicht gedrückt: Es fließt kein Strom.

Als "Öffner": Kontakte 1 und 2 werden angeschlossen.

Taster gedrückt: Es fließt kein Strom. Taster nicht gedrückt: Es fließt Strom.

5.1.7. NTC-Widerstand



Bei diesem Bauteil handelt es sich um einen Wärmesensor, mit dem man Temperaturen messen kann. Bei 20°C beträgt der Widerstand 1,5 k Ω (kilo-Ohm). **NTC** bedeutet Negativer Temperatur Koeffizient. Das heißt, dass der Widerstandswert mit steigender Temperatur sinkt.

Die Informationen die uns die Sensoren liefern (z.B. hell – dunkel, gedrückt – nicht gedrückt, Temperaturwert) kann man, wie wir später noch sehen werden, über das Interface an den PC weiterleiten, und dann mit Hilfe der Software z.B. einen Motor so programmieren, dass er eine Tür öffnet, sobald die Lichtschranke unterbrochen wird.

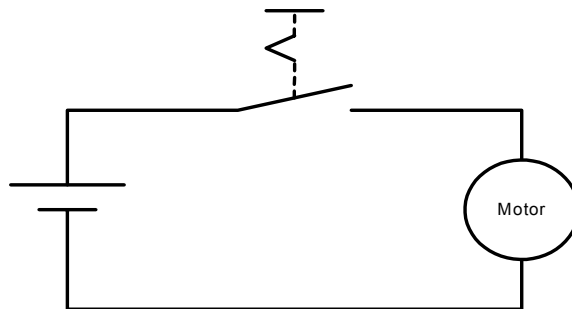
5.2. Projekt 1: Einfache Motorsteuerung

5.2.1. Lernziel

Die Schüler sollen lernen, eine einfache Motorsteuerung mit Hilfe der Fischertechnik Software zu realisieren.

5.2.2. Aufgabe

- 1) Ein Motor soll über einen Schalter ein- bzw. ausgeschaltet werden. Dazu wird eine 9 V-Batterie über einen Schalter, wie im Schema gezeigt, mit dem Motor verbunden. Verwende als Schalter den Polwendeschalter von Fischertechnik.



Was stellst du fest wenn der Schalter geschlossen wird?

Was stellst du fest wenn der Schalter wieder geöffnet wird?

- 2) Dieselbe Schaltung soll jetzt mit Hilfe des PC realisiert werden. Baue dazu das Modell der Motorsteuerung auf (Bauanleitung Seite 4) und antworte auf die folgenden Fragen!

In den Abschnitten a. bis m. wird erklärt was zum Bewältigen der Aufgabe 2 gebraucht wird.

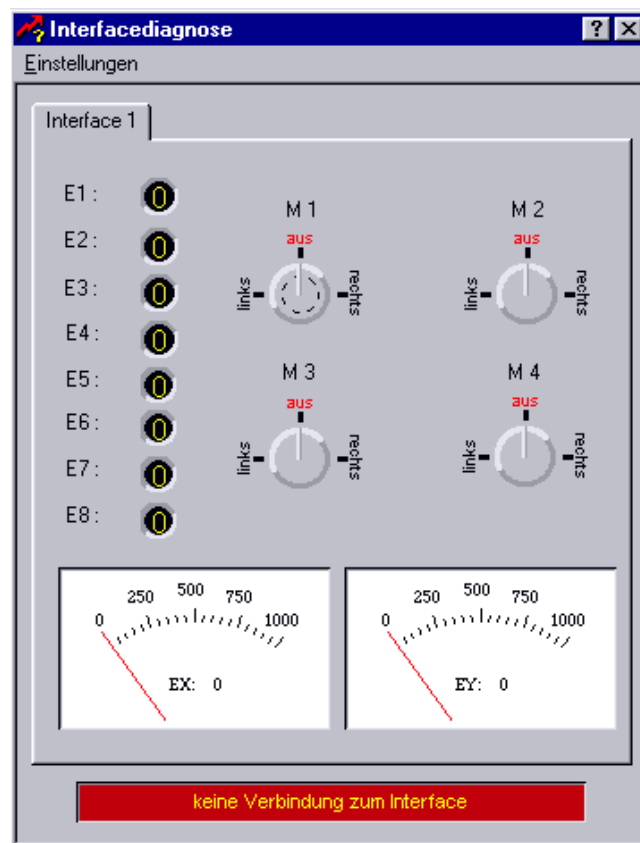
In den Abschnitten a) bis m) wird erklärt was zum Bewältigen der Aufgabe 2 gebraucht wird.

5.2.3. Erklärungen

- a) Test der Hardware

Damit die Steuerungsprogramme die Modelle über das Interface steuern können, muss das Interface durch das Verbindungskabel an eine serielle Schnittstelle (COM1 bis COM4) des PC's angeschlossen werden. Nun muss die Stromversorgung des Interfaces hergestellt werden (Netzgerät oder Batterie).

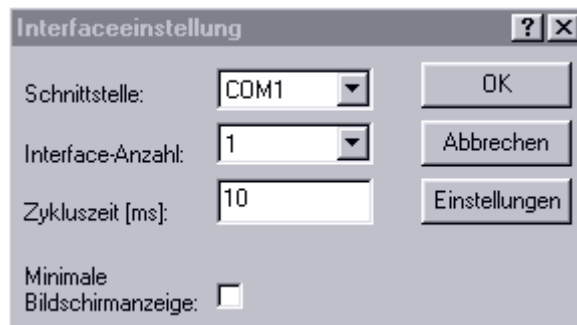
Damit die Verbindung von PC und Interface korrekt funktioniert, muss das Interface in LLWin eingestellt werden. Hierzu kann man das Programm Interfacediagnose verwenden, das im Menü *Optionen* mit dem Befehl *Interfacediagnose* aufgerufen werden kann. Es erscheint folgendes Fenster:



Im Diagnosefenster werden die Ein- und Ausgänge des Interface dargestellt. Der Balken ganz unten zeigt den Verbindungsstatus zwischen PC und Interface an.

- Simulationsmodus
- Kein Verbindung zum Interface
- Verbindung zum Interface o.k.

Zum Anpassen der Verbindungseinstellungen muss man im Menü *Einstellungen* den Befehl *Interfaceeinstellungen* aufrufen, worauf folgendes Fenster erscheint:



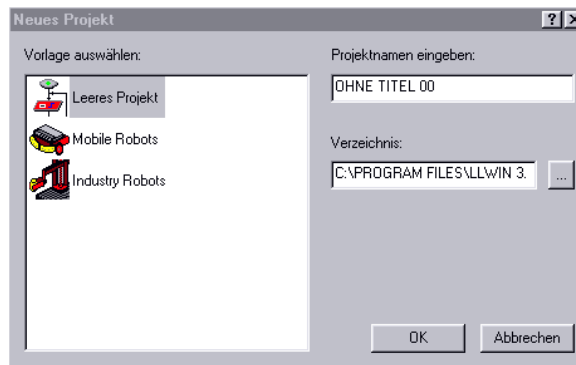
Hier kann man die Schnittstelle auswählen, an der man das Interface angeschlossen hat. Mit der Interface-Anzahl wird festgelegt, ob man auf ein Interface allein (1) oder ein Interface mit Extension Module (2) zugreifen möchte. Nach dem Schließen des Dialogs versucht das Diagnoseprogramm, auf das angeschlossene Interface zuzugreifen. Ist dies gelungen so erscheint im Fenster Interfacediagnose "Verbindung zum Interface o.k.".

b) Erstes Steuerungsprogramm

Weil die Beschreibung von Steuerungsprogrammen als Text umständlich und auch nicht sehr übersichtlich ist, verwendet man für die Darstellung der nacheinander auszuführenden Aktionen Ablaufpläne. Mit Hilfe der Software LLWin kann man Abläufe durch Verbinden von Bausteinen zeichnen und damit das Steuerungsprogramm für die angeschlossene Hardware erstellen.

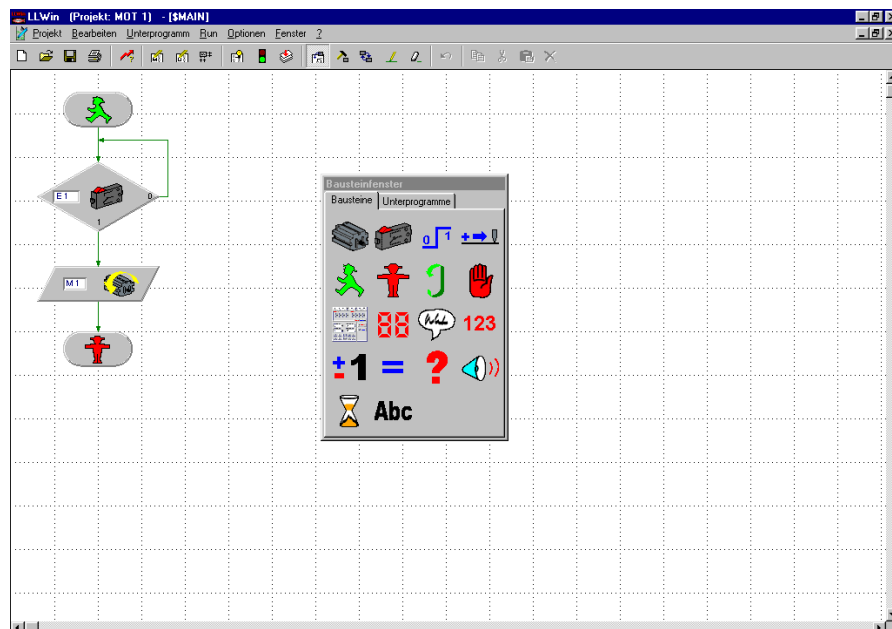
Im ersten Steuerungsprogramm soll ein Motor gesteuert werden, der nach dem Drücken einer Taste anläuft (Aufgabe 2).

Zum Erstellen dieses Programms ruft man den Befehl *Neu* im Menü *Projekt* auf. Im Dialogfenster wählt man in der Liste der Projektvorlagen den Eintrag *Leeres Projekt* aus.



In den Projektvorlagen sind bereits Unterprogramme für verschiedene Fischertechnik-Modelltypen vorhanden. Will man keine der vorhandenen Vorlagen nutzen wählt man *leeres Projekt*. Es kann gleich zu Beginn ein Projektname eingegeben werden. Will man das nicht, erhält das neue Projekt automatisch den Namen *OHNE TITEL*. Weiterhin kann man angeben, in welchem Verzeichnis das neue Projekt abgespeichert werden soll. Bestätigt man mit OK, wird das neue Projekt angelegt und ein Editierfenster öffnet sich.

Im Fenster ist bereits das erste Element des Ablaufplans, der Baustein *START*, enthalten. Weitere Bausteine können über das Bausteinfenster eingefügt werden. Sollte dieses nicht geöffnet sein, muss man den Befehl *Baustein einfügen* im Menü *Bearbeiten* aufrufen. Ist das Fenster immer noch nicht zu finden, so aktiviert man es mit dem Befehl *Bausteinfenster* im Menü *Optionen*.



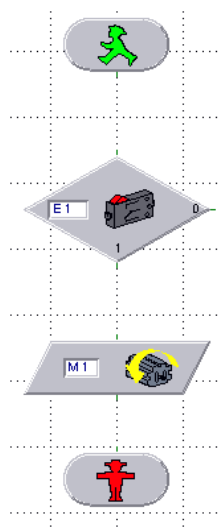
Aus dem Bausteinfenster können die Bausteine aufgerufen und auf der Arbeitsoberfläche eingefügt werden. Dazu bewegt man die Maus auf das Symbol des gewünschten Bausteins. Der Name des Bausteins wird dann zusätzlich eingeblendet.

Jetzt drückt man die linke Maustaste und zieht den Baustein bei gedrückter Maustaste an die gewünschte Stelle auf der Arbeitsoberfläche. Lässt man die Maustaste los, erscheint das Dialogfenster des jeweiligen Bausteins. Da der Zustand von einem Taster abgefragt werden soll, muss man den Baustein *EINGANG* einfügen.

Im Dialog kann man nun einstellen, an welchem Eingang des Interface der Taster angeschlossen werden soll. Der Eingang E1, der hier verwendet werden soll, ist bereits voreingestellt. Der Typ des Eingangs soll auf Taster eingestellt und die Verzweigung rechts bei 0 hell unterlegt sein. Nach Eingabe aller benötigten Werte wird der Dialog mit einem Klick auf OK geschlossen und der Baustein ist auf der Arbeitsoberfläche platziert.

Bausteine lassen sich auch nach dem Einfügen bei gedrückter linker Maustaste an die gewünschte Stelle verschieben. Das Ändern der Bausteinwerte ist ebenfalls möglich. Ein Klick mit der rechten Maustaste auf den Baustein öffnet den bereits bekannten Dialog.

Nun müssen noch die Bausteine *AUSGANG* (Ausgang M1, Motor, links drehend) und *ENDE* eingefügt werden, so dass der folgende Ablaufplan entsteht.



c) Verbindungslinien

Was nun noch fehlt, sind die Verbindungslinien zwischen den einzelnen Bausteinen. Vielleicht sind die Bausteine bereits so platziert, dass die Verbindungslinien automatisch gezogen wurden. Dieses automatische Verbinden funktioniert beim

Einfügen eines neuen Bausteins, wenn der Abstand zum vorhergehenden Baustein höchstens 9 Rasterpunkte beträgt.

Zum manuellen Zeichnen von Linien muss das Bleistiftsymbol in der Symbolleiste ausgewählt werden. Aus dem Cursor wird ein Stift, und man kann zwischen 2 Bausteine eine Linie ziehen, indem man zuerst mit der linken Maustaste auf den Ausgang des ersten Bausteins klickt. Lässt man die Maustaste wieder los, wird aus dem Cursor ein Fadenkreuz. Daran erkennt man, dass der Ausgang getroffen wurde. Klickt man nun auf den Eingang des zweiten Bausteins, wird die Linie gezogen.

Verbindungslinien lassen sich auch durch Verschieben der Bausteine erzeugen. Wenn ein Baustein so verschoben wird, dass sein Eingang genau auf dem Ausgang eines andern liegt, wird zwischen beiden Bausteinen eine Verbindungsline erzeugt. Danach kann man den Baustein in seine Endposition schieben.

Das Löschen von Verbindungslinien wird durch das Klicken mit der linken Maustaste auf den Radiergummi in der Symbolleiste erreicht. Betätigt man jetzt die linke Maustaste auf einer Verbindungsline, so wird diese gelöscht.

d) Baustein *START*



Symbol im Bausteinfenster

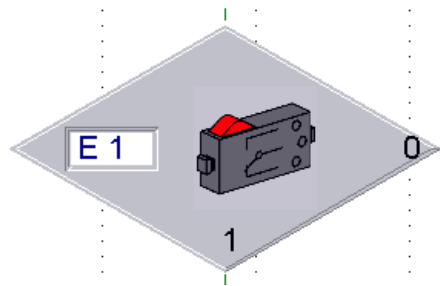


Ein Ablauf beginnt immer mit einem Startbaustein. Fehlt dieser Baustein am Anfang, wird der Ablauf nicht abgearbeitet. Wenn ein Projekt mehrere Abläufe enthält, muss jeder dieser Abläufe einen Startbaustein enthalten. Die verschiedenen Abläufe werden dann gleichzeitig gestartet.

e) Baustein *EINGANG*

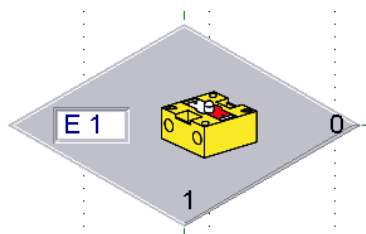


Symbol im Bausteinfenster

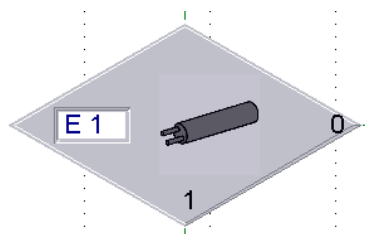


Der Baustein *EINGANG* fragt den Zustand eines digitalen Eingangs E1-E8 am Interface ab. Ein digitaler Eingang kann nur zwei Zustände annehmen, nämlich 0 und 1.

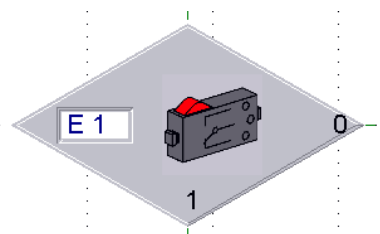
An den digitalen Interfaceeingängen kann man folgende digitale Fischertechnik Sensoren anschließen:



Fototransistor



Reedkontakt



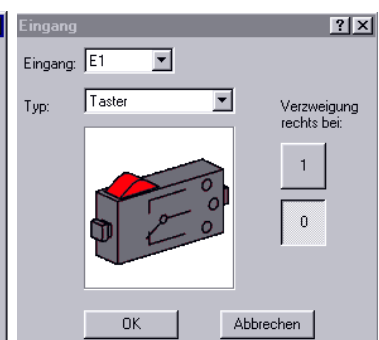
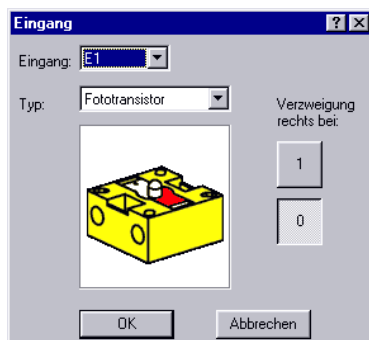
Taster

Fototransistor: hell – dunkel

Reedkontakt (Magnetsensor): geschaltet – nicht geschaltet

Taster: gedrückt – nicht gedrückt

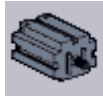
Beim Einfügen des Bausteins wird nachfolgender Dialog aufgerufen:



Nach Auswahl des Sensortyps wird das entsprechende Symbol im Baustein abgebildet. Je nach Zustand (0 oder 1) des Eingangs kann der Ablauf nach rechts verzweigen oder am unteren Anschluss des Bausteins fortgesetzt werden. Ob die

Verzweigung nach rechts bei 0 oder 1 erfolgen soll, kann im Dialog ausgewählt werden.

f) Baustein *AUSGANG*



Symbol im Bausteinfenster



Mit dem Baustein *AUSGANG* schaltet man einen der Ausgänge M1-M4 des Interface. An einem Ausgang am Interface kann entweder ein Motor, eine Lampe oder ein Elektromagnet angeschlossen werden.



Beim Einfügen des Bausteins wird nachfolgender Dialog aufgerufen:



Motor

Lampe

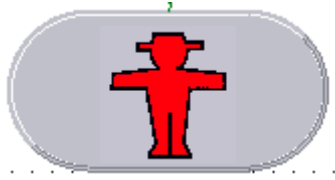
Elektromagnet

Im Dialogfeld Typ kann man zwischen den drei Typen Motor, Lampe und Elektromagnet auswählen. Außerdem stellt man im Dialog den gewünschten Zustand des Ausgangs ein: Bei einem Motor: links, rechts oder aus. Bei einer Lampe oder einem Elektromagneten: ein oder aus. Der eingestellte Zustand wird ebenfalls im Bausteinsymbol abgebildet.

g) Baustein *ENDE*



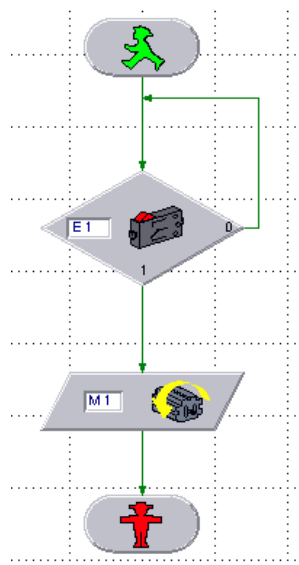
Symbol im Bausteinfenster



Soll ein Ablauf beendet werden, verbindet man den Ausgang des letzten Bausteins mit dem Baustein *ENDE*. Ein Ablauf kann auch an verschiedenen Stellen mit einem Baustein *ENDE* beendet werden. Es besteht weiterhin die Möglichkeit, mehrere Ausgänge mit einem einzigen Baustein *ENDE* zu verbinden.

Bei vielen Anwendungen wird ein Ablauf als Endlosschleife ausgeführt. Dann wird kein Baustein *ENDE* benötigt.

Der fertige Ablaufplan sollte nach dem Verbinden der Bausteine so aussehen:



Die Verbindung vom Ausgang "0" des Tasters zurück zu dessen Eingang wird gezeichnet, indem zunächst auf den Ausgang und anschließend auf die Verbindungslinie START-TASTER geklickt wird. Sie bedeutet nichts anderes, als dass solange gewartet wird, bis der Taster betätigt und damit der Motor eingeschaltet wird.

h) Testen

Zum Testen des Steuerungsprogramms benutzt man das bereits aufgebaute Modell. Danach ruft man den Befehl *Init* im Menü *Run* auf. Beim Eintritt in den Init-Modus wird getestet, ob bei allen Bausteinen die Anschlüsse verbunden sind. Nicht verbundene Bausteine leuchten violett auf. Dann muss man zurück in den Editiermodus wechseln (Taste F6). Ist das Steuerungsprogramm fehlerfrei, wählt man im Menü *Run* den Befehl *Start*. Zunächst wird das Steuerungsprogramm übersetzt, wobei eine Balkenanzeige über den Bearbeitungsstand informiert. Nach der Übersetzung wird das Projekt im Online-Modus gestartet.

Der Baustein Eingang leuchtet rot auf. Es wird angezeigt, dass der Ablauf in diesem Baustein auf ein Ereignis wartet, nämlich dass der Taster gedrückt wird. Drückt man den am Eingang 1 angeschlossenen Taster, ist die Bedingung zum Weiterschalten erfüllt. Der Motor wird eingeschaltet und der Ablauf ruft den nächsten Schritt den unteren Baustein *ENDE* auf.

i) Symbolleiste

Die wichtigsten Menübefehle können auch direkt durch Anklicken des entsprechenden Icons in der Symbolleiste ausgeführt werden. Je nach Programmzustand werden die möglichen Kommandos als Icon eingeblendet.

	Neues Projekt		Bausteine einfügen
	Projekt öffnen		Bausteine löschen
	Projekt speichern		Bausteine ersetzen
	Seite drucken		Linie zeichnen
	Interfacediagnose		Linie löschen
	Hauptprogramm		Rückgängig
	Unterprogramm		Kopieren
	Unterprogramm-Design		Ausschneiden
	Init		Einfügen



Start-Stop



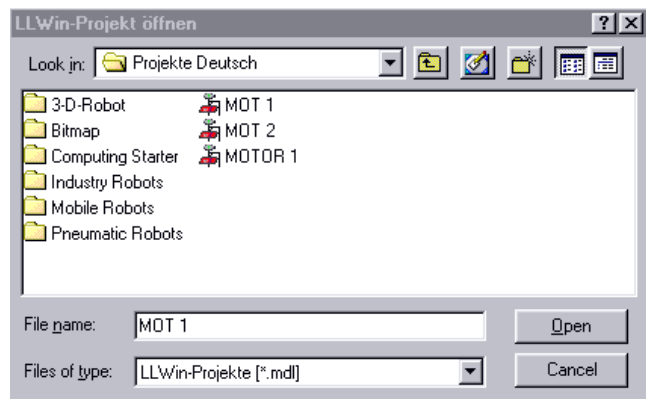
Löschen



Download

j) Öffnen (Projekt Menü)

Mit diesem Befehl wird ein bereits vorhandenes Projekt geöffnet. Man wählt ein bereits erstelltes Projekt aus und bestätigt mit Öffnen.

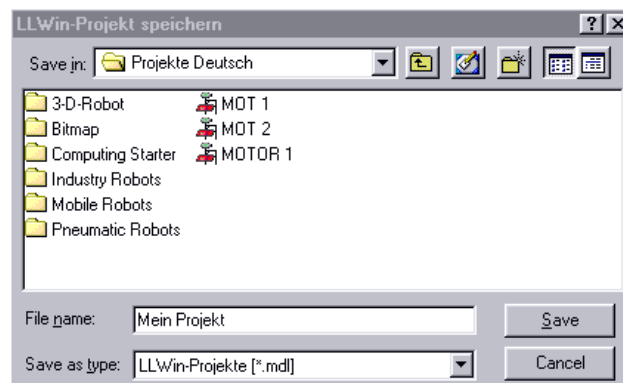


k) Speichern (Menü Projekt)

Mit diesem Befehl speichert man ein Projekt ab. Hat dieses Projekt noch keinen Namen, muss man spätestens jetzt einen Namen eingeben.

l) Speichern unter (Menü Projekt)

Will man ein vorhandenes Projekt unter einem neuen Namen speichern, z.B. weil man es geändert hat und das Projekt ohne Änderungen noch behalten möchte, wählt man diesen Befehl aus.



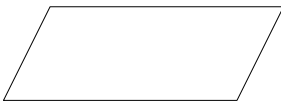
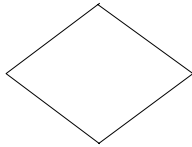
Man wählt dabei das gewünschte Verzeichnis, gibt den gewünschten Dateinamen ein und bestätigt mit Speichern.

m) Schließen (Menü Projekt)

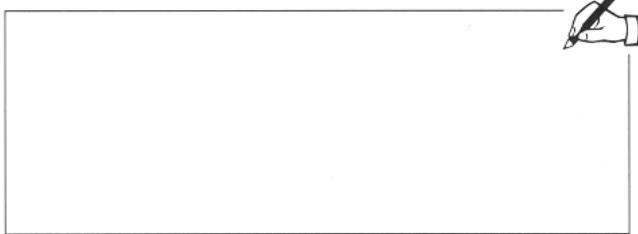
Mit diesem Befehl wird ein gerade geöffnetes Projekt geschlossen. Wurde es seit dem letzten Abspeichern verändert, fragt LLWin, ob die Änderung gespeichert werden soll.

5.2.4. Aufgabe

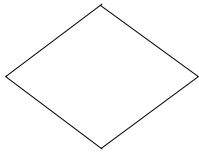
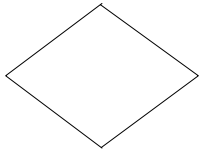
- 1) Suche die benötigten Bausteine unter deiner Tafel zusammen und erstelle dein erstes Steuerungsprogramm zuerst einmal auf deiner Magnettafel. Vervollständige danach das folgende Diagramm und erstelle anschließend dasselbe Programm am PC.



Was stellst du fest?



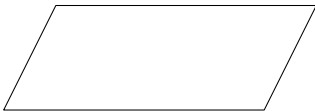
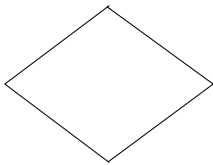
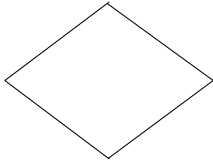
- 2) Suche die benötigten Bausteine unter deiner Tafel zusammen und verändere das Programm, damit der Motor beim Betätigen von E2 abgestellt wird, zuerst einmal auf deiner Magnettafel. Vervollständige das folgende Diagramm und erstelle dasselbe Programm an PC.



Was stellst du fest beim nochmaligem Betätigen des Tasters E1?

Es passiert nichts.

- 3) Suche die benötigten Bausteine unter deiner Tafel zusammen und verändere das Programm, damit der Motor bei jedem Betätigen des Tasters E1 wieder anläuft, zuerst einmal auf deiner Magnettafel. Vervollständige das Diagramm und erstelle dasselbe Programm an PC.



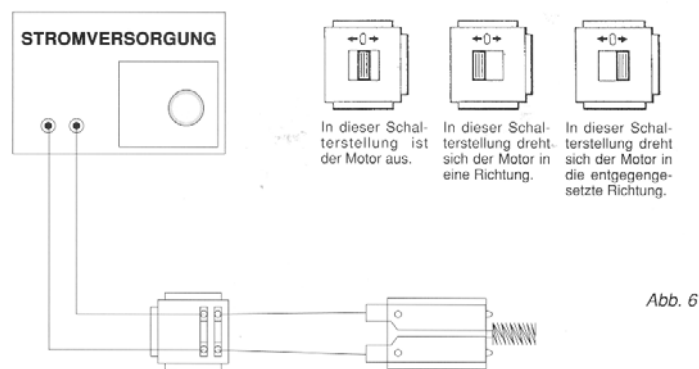
5.3. Projekt 2: Erweiterte Motorsteuerung

5.3.1. Lernziel

Die Schüler sollen lernen mit der Fischertechnik Software einen Motor im Links-Rechtslauf zu betreiben.

5.3.2. Aufgaben

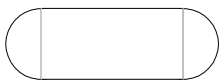
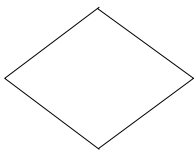
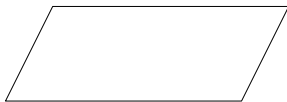
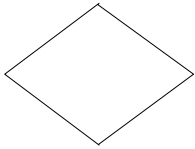
- 1) Ein Motor soll über einen Schalter von Rechts- in Linkslauf umgeschaltet werden können. Dazu soll der Stromkreis wie in der Abbildung aufgebaut werden.



Was stellst du fest, wenn der Schalter nach rechts geschaltet wird?

Was stellst du fest, wenn der Schalter nach links geschaltet wird?

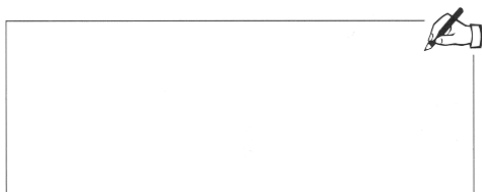
- 2) Versuche nun die gleiche Motorsteuerung zuerst auf deiner Magnettafel und dann mit Hilfe von LLWin am PC zu realisieren. Vervollständige das Diagramm.



Was passiert beim Betätigen von E1?



Was passiert beim Betätigen von E2?




Was passiert beim nochmaligem Betätigen von E1 oder E2?




Erstelle jetzt ein Programm mit dem man hin und her schalten kann.

Was stellst du jetzt fest?



Erstelle jetzt ein Programm mit dem man hin und her schalten kann. Achte jedoch darauf, dass wie in Teil 1 der Aufgabe, der Motor vor jedem Umschalten ausgeschaltet wird.


Was passiert beim Betätigen von E1?




Was passiert beim Betätigen von E2?



Was muss vor dem Umschalten in den Rechtslauf getan werden?

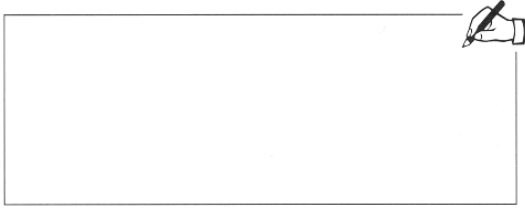


Was stellst du weiter fest?



Versuche ein Programm zu entwickeln, mit dem man beliebig hin und her schalten kann.

Was passiert, wenn jetzt während dem Linkslauf E2 gedrückt wird?



Dies darf jedoch nicht geschehen.

5.3.3. Erklärungen

a) Grundlagen zur Verwendung von Variablen

Variablen sind Zwischenspeicher für ganze Zahlen im Bereich von - 32767 bis 32767. Sie können z.B. zum Speichern von Programmezuständen oder als Zählvariable verwendet werden. Über Variablen können Informationen zwischen mehreren parallel arbeitenden Abläufen ausgetauscht werden. Der Wert aller Variablen wird beim Start des Steuerungsprogramms auf 0 gesetzt.

In LLWin-Projekten wird zwischen Variablen und Standard-Zählvariablen unterschieden.

Die 99 Variablen werden mit VAR1-VAR99 bezeichnet. Die Bezeichnungen VAR1-VAR99 werden als Adressen benötigt, um in den Bausteinen auf eine bestimmte Variable zugreifen zu können.

Die Standard-Zählvariablen Z1-Z16 werden als Zählwert im Baustein *POSITION* verwendet. Sie können aber auch von anderen Bausteinen ausgewertet und überschrieben werden.

Während ein Steuerungsprogramm im Online-Modus läuft, kann der aktuelle Wert einer Variablen mit dem Baustein *WERTE ANZEIGEN* angezeigt werden.

b) Steuern paralleler Abläufe mit Variablen

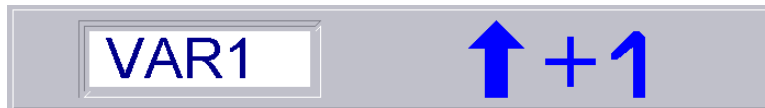
Man kann innerhalb eines Steuerungsprogramms auch mehrere Abläufe erstellen, die dann gleichzeitig oder parallel, wie man das in der Fachsprache nennt, durchlaufen werden. Meistens wird es aber so sein, dass die beiden Abläufe nicht völlig unabhängig voneinander sind. Zum Beispiel kann bei einem Ablauf an einer Stelle erst dann fortgefahren werden, wenn beim zweiten Ablauf eine bestimmte

Bedingung oder ein Zustand erreicht wurde. Dieses gegenseitige Steuern von Abläufen lässt sich sehr einfach mit Variablen realisieren. Dazu werden folgende Bausteine benötigt:

c) Baustein *VARIABLE* +/-1

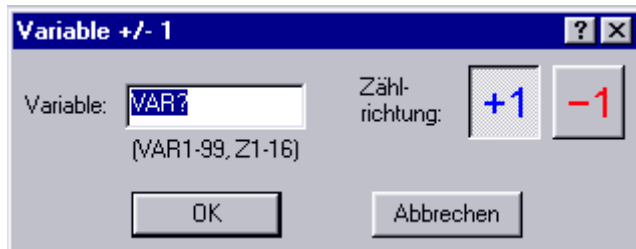


Symbol im Bausteinfenster



Mit diesem Baustein kann man den Wert einer Variablen um eins erhöhen oder um eins verringern.

Beim Einfügen des Bausteins wird nachfolgender Dialog aufgerufen:



Im Dialogfeld gibt man an, welche Variable bearbeitet werden soll. Außerdem wählt man aus, ob der Wert um eins erhöht oder um eins verringert werden soll. Diese Zählrichtung wird im Symbol des Bausteins angezeigt.

d) Baustein *VERGLEICH*

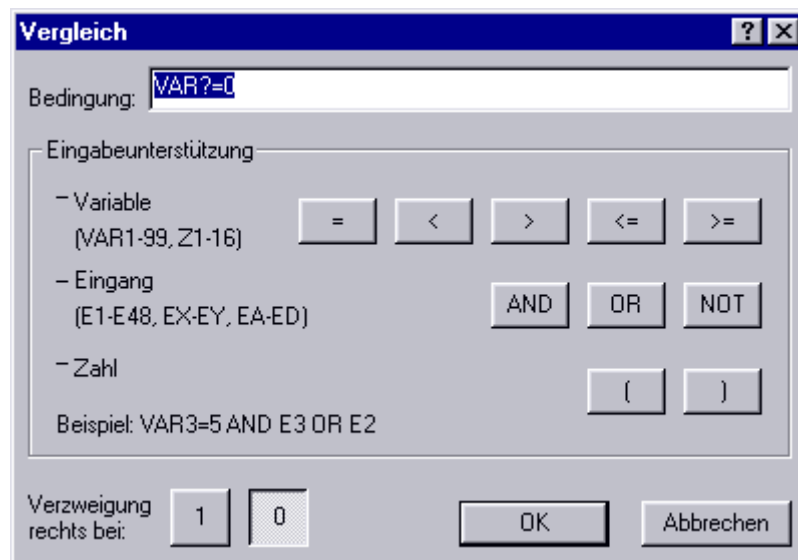


Symbol im Bausteinfenster



Im Baustein *VERGLEICH* wird eine Bedingung ausgewertet. Abhängig davon, ob die Bedingung erfüllt ist oder nicht, verzweigt der Ablauf nach rechts oder wird am unteren Ausgang des Bausteins fortgesetzt. Die Zahlen 0 und 1 an den Ausgängen

stehen für "Bedingung erfüllt" (1) bzw. "Bedingung nicht erfüllt" (0). Ob bei 1 oder 0 nach rechts verzweigt wird, kann im Dialog des Bausteins festgelegt werden.



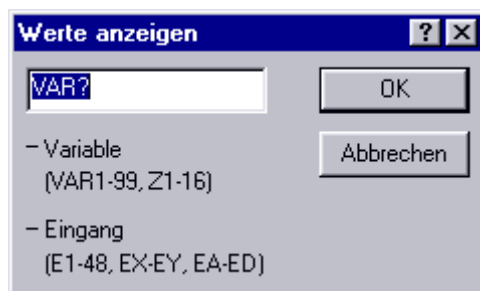
Beim Einfügen des Bausteins gibt man im dargestellten Dialogfenster eine maximal 40 Zeichen umfassende Bedingung ein.

Mit der Eingabeunterstützung des Dialogs können die Vergleichsoperatoren, die logischen Verknüpfungen und die Klammern durch Mausklick eingefügt werden. Die ganze Formel kann aber auch komplett über die Tastatur eingetippt werden.

e) Baustein *WERTE ANZEIGEN*

123

Symbol im Bausteinfenster

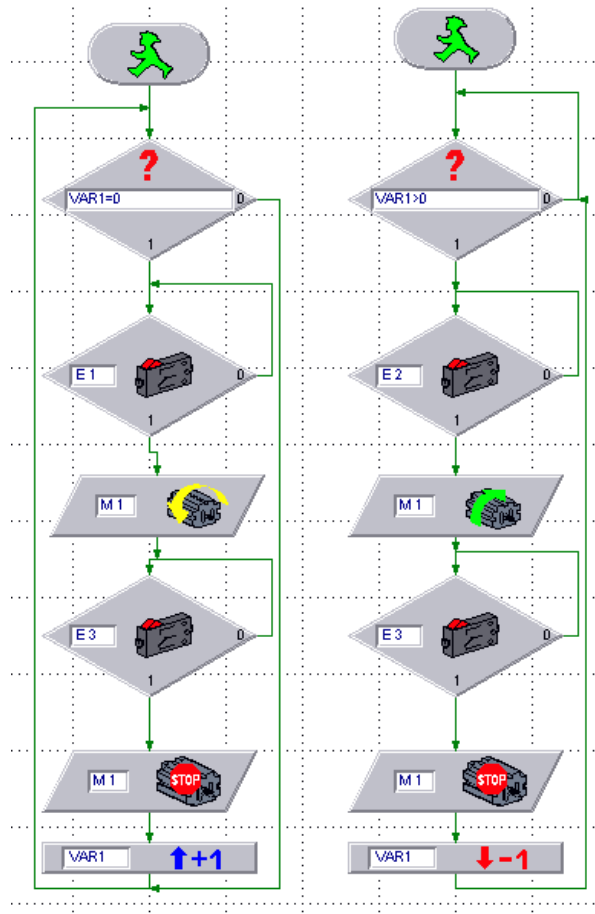


Im Bausteindialog musst du angeben, welche Variable oder welcher Eingang angezeigt werden soll. Dieser Baustein zeigt, während das Steuerungsprogramm im Online-Modus läuft, stets den aktuellen Wert eines digitalen oder analogen Eingangs

oder einer Variablen an. Der Baustein wird ohne Verbindungen zu anderen Bausteinen auf der Programmieroberfläche eingefügt.

Ohne den Online-Modus zu unterbrechen, kann ein anderer Wert im Baustein angezeigt werden. Dazu ruft man mit der linken Maustaste den Dialog auf und verändert den Eintrag.

Das Programm welches mit diesen neuen Bausteinen realisiert werden soll müsste wie folgt aussehen:



Zu Beginn jedes Ablaufes ist die Variable 1 immer null, somit ist die Bedingung im ersten Baustein Vergleich erfüllt und man kann mit E1 in den Linkslauf schalten. Solange die Variable 1 gleich 0 ist, ist die Bedingung im Baustein Vergleich im rechten Ablauf nicht erfüllt und somit kann der Rechtslauf nicht eingeschaltet werden. Nachdem der Linkslauf nun aber mit E3 gestoppt ist wird im Baustein Variable +/-1 die Variable 1 um 1 erhöht, damit ist sie größer als 0 und die Bedingung zum Einschalten des Rechtslaufs ist somit erfüllt. Wird der Rechtslauf gestoppt so wird die Variable 1 um 1 tiefer also wieder auf 0 gestellt und man kann wieder in den Linkslauf schalten.

5.4. Projekt 3: Motorsteuerung mit Zeitverzögerung

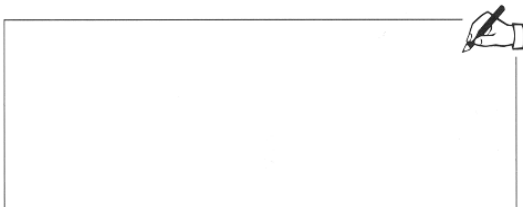
5.4.1. Lernziel

Die Schüler sollen lernen mit der Fischertechnik Software eine Motorsteuerung mit Zeitverzögerung zu realisieren.

5.4.2. Aufgaben

- 1) Schalte den Motor aus Aufgabe 5.3.2. in den Rechtslauf und 10 Sekunden später in den Linkslauf.

Was musst du dazu alles beachten?



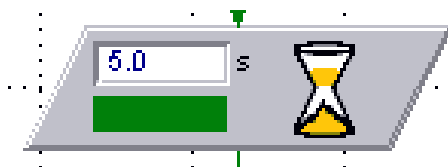
- 2) Versuche diese Schaltung mit Hilfe von Fischertechnik zu bewältigen.

5.4.3. Erklärungen

- a) Baustein *WARTE*



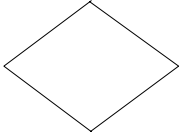
Symbol im Bausteinfenster



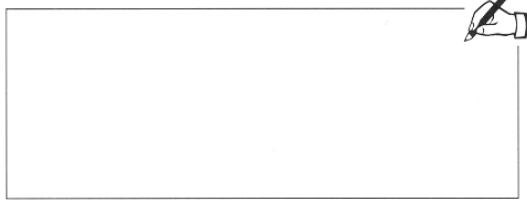
Im Dialogfenster des Bausteins *WARTE* kann man die gewünschte Zeit eingeben. Für unseren Fall muss man als "Typ" *definiert* einstellen (was man mit einer zufällig langen Zeit anfangen kann wird später erklärt).



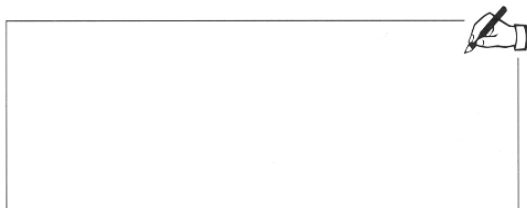
Dieses Programm könnte dann so aussehen:



Was passiert nachdem die zehn Sekunden abgelaufen sind?



Was passiert bei nochmaligem Betätigen von E1?



5.5. Projekt 4: Verkehrsampel

5.5.1. Lernziel

Die Schüler sollen mit der Fischertechnik Software eine Verkehrsampelsteuerung realisieren

5.5.2. Aufgabe

- 1) Baue das Modell der Verkehrsampel auf! (Bauanleitung Seite 8)

5.5.3. Funktionsbeschreibung

Durch Betätigung des Schiebers am Taster E2 wird die Ampelanlage in Betrieb geschaltet und die rote Lampe leuchtet. Wird jetzt der Taster E1 gedrückt so erlischt nach 2 Sekunden das rote Licht und das Orange leuchtet auf. Wiederum nach 2 Sekunden erlischt das orange Licht und das Grüne leuchtet. Nachdem man nun 10 Sekunden gewartet hat wird die Anlage in der umgekehrten Reihenfolge wieder auf rot geschaltet. Dieser Vorgang kann beliebig oft durch Betätigen von E1 wiederholt werden.

Erstelle diesen Ablauf:

5.5.4. Aufgabe

Jetzt soll das Programm so umgeändert werden, dass bei jedem Hin -und Herschalten zwischen Rot und Grün die orange Lampe mit einem Zeitabstand von einer Sekunde fünfmal aus und dann wieder eingeschaltet wird. Hierzu wird ein Unterprogramm eingesetzt.

5.5.5. Erklärungen

- a) Baustein *UNTERPROGRAMM*

Größere Ablaufpläne können so umfangreich werden, dass man die Übersicht verliert. So hat man bei einer großen Anzahl von Bausteinen und Verbindungslinien Schwierigkeiten, die Struktur des gesamten Projektes zu überblicken. Um große Projekte übersichtlicher erstellen und bearbeiten zu können, gibt es in LLWin die Möglichkeit, Unterprogramme zu verwenden.

In LLWin ist ein Unterprogramm ein spezieller Bausteintyp. Im Bausteinfenster kann man beispielsweise den Unterprogrammbaustein UP1 auswählen, der in jedem neu geöffneten Projekt existiert.

Unterprogrammbausteine können wie jeder andere Baustein auf die Arbeitsoberfläche gezogen und dort abgelegt werden.

Die Besonderheit der Unterprogrammbausteine ist, dass man die Funktion selbst programmieren kann. So wie man die Ablaufpläne für Steuerungsprogramme erstellt, kann man für die Funktion dieses Bausteins ebenfalls einen Ablaufplan erstellen. Soll ein Ablauf im Unterprogramm mit dem Ablauf im Hauptprogramm über Linien verbunden werden, benötigt das Unterprogramm hierzu spezielle Ein- und Ausgänge. Ein Unterprogramm kann aber auch einen eigenständigen Ablauf enthalten, der parallel zu anderen Abläufen abgearbeitet wird und keine Verbindungen zu anderen Abläufen hat.

Wird die Funktion, die ein Unterprogramm erfüllt, häufiger benötigt, muss sie nicht jedes Mal neu eingegeben werden. Es genügt, das entsprechende Unterprogramm einmal zu erstellen. Anschließend lässt es sich mehrfach in einem Steuerungsprogramm verwenden.

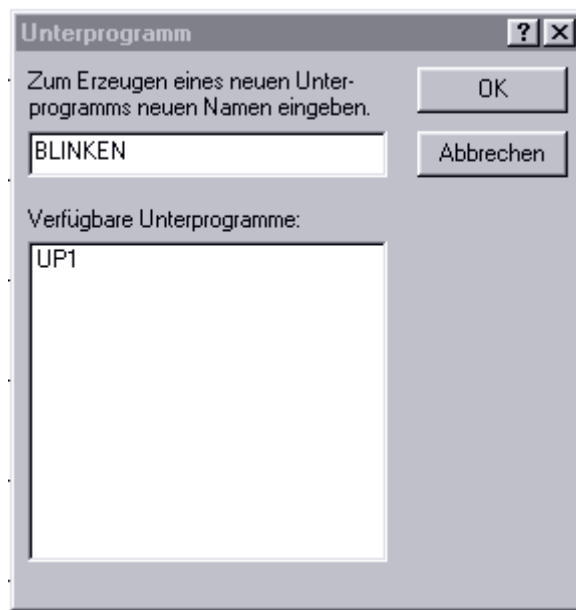
Die einzelnen Arbeitsschritte zum Erstellen und Bearbeiten eines Unterprogramms sowie zum Einfügen des Unterprogrammbausteins in das Hauptprogramm können an einem praktischen Beispiel nachvollzogen werden.

b) Beispiel für die Verwendung eines Unterprogramms

Die Verwendung von Unterprogrammen soll nachfolgend am Beispiel der Ampelsteuerung erläutert werden.

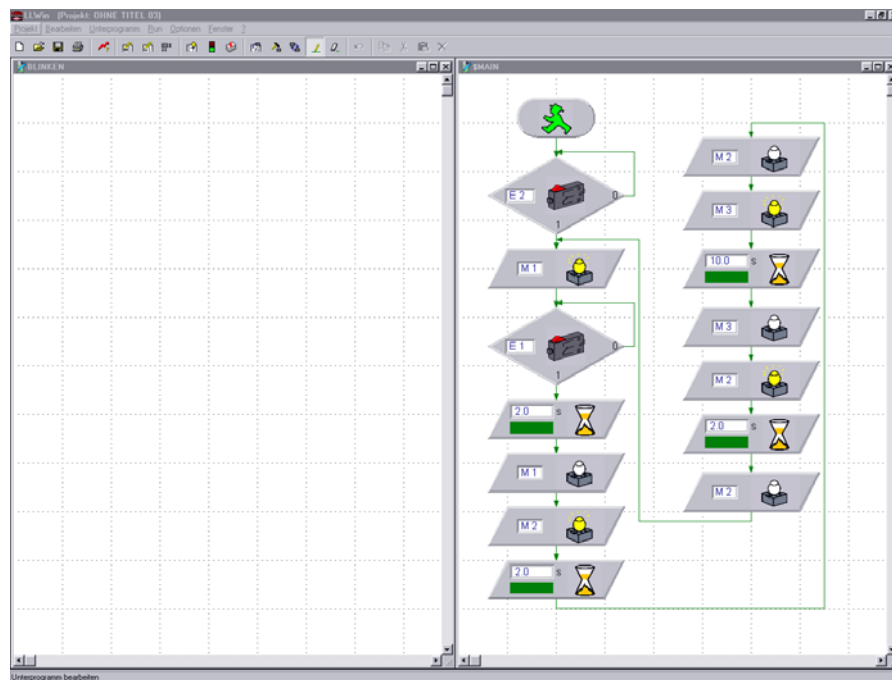
Zum Erstellen oder Bearbeiten von Unterprogrammen ruft man den Befehl *Unterprogramm* im Menü *Bearbeiten* auf.

Im Eingabefeld des Dialoges wird als Unterprogrammname BLINKEN eingegeben.



Eine leere Arbeitsoberfläche erscheint, auf dem der Ablaufplan für das Unterprogramm erstellt werden kann.

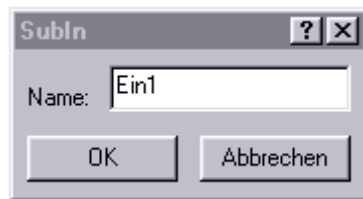
Mit dem Befehl *Nebeneinander* im Menü *Fenster* kann man die beiden Fenster für das Hauptprogramm und das Unterprogramm nebeneinander anordnen. Das Fenster Hauptprogramm trägt den Namen \$MAIN, während das Unterprogramm den Namen BLINKEN besitzt.



Die Verbindung zum Hauptprogramm wird durch die Bausteine SUBIN und SUBOUT hergestellt. Diese beiden Bausteine sind im Bausteinfenster nur verfügbar, wenn man ein Unterprogramm bearbeitet.

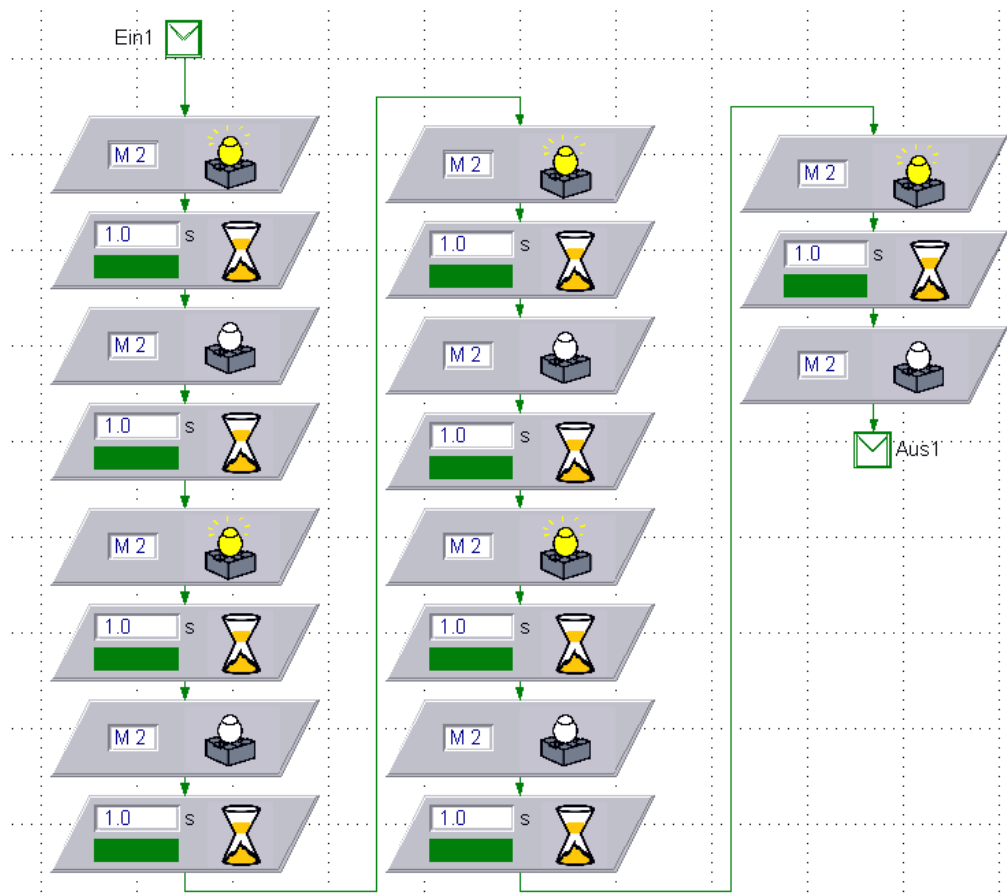


Beim Platzieren wird jeweils nach einem Namen für den Eingang bzw. Ausgang gefragt. Man kann die vorgeschlagenen Namen verwenden und mit OK bestätigen.

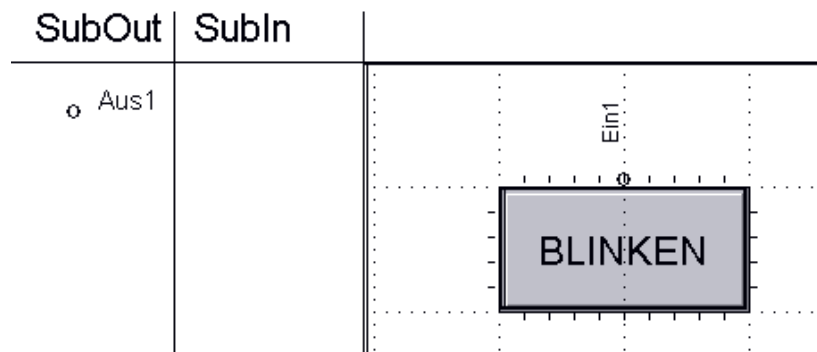


Als nächstes wird der Ablauf des Blinkens programmiert.

Nach Zeichnen der Verbindungslinien sollte das Unterprogramm folgendes Aussehen haben:

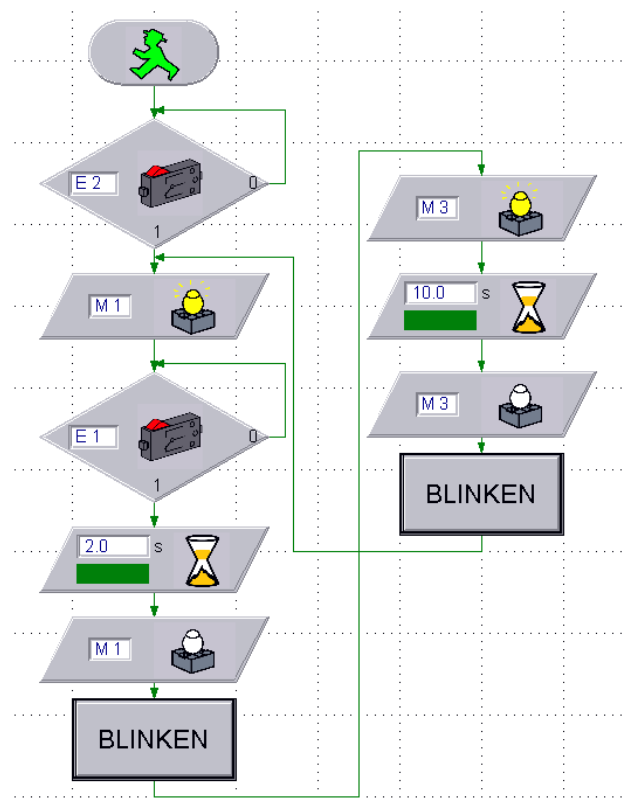


Nach Aufruf des Befehls *Design* im Menü *Unterprogramme* werden im Unterprogrammbaustein *BLINKEN* die Anschlüsse auf dem Bausteinrahmen platziert. Dies ist erforderlich, um den Unterprogrammbaustein mit den restlichen Bausteinen des Hauptprogramms verbinden zu können.



Diese Anschlüsse kann man im neu geöffneten Fenster aus der Tabelle auf der linken Seite herausziehen und auf dem Rahmen des Unterprogrammbausteins absetzen.

Den Unterprogrammbaustein *BLINKEN* kann man nun im Bausteinfenster auswählen und in das Hauptprogramm einfügen. Sind alle Linien zum Unterprogrammbaustein gezogen, ist das erste Steuerungsprogramm unter Verwendung eines Unterprogramms fertiggestellt.



5.6. Projekt 5: Händetrockner

5.6.1. Lernziel

Die Schüler sollen mit der Fischertechnik Software eine Händetrocknersteuerung mit Fotozelle realisieren.

5.6.2. Aufgabe

- 1) Baue das Modell des Händetrockners auf! (Bauanleitung Seite 6)

5.6.3. Funktionsbeschreibung

Der Händetrockner soll nun so programmiert werden, dass, sobald die Lichtschranke unterbrochen wird, der Lüfter ein- und nach 5 Sekunden wieder ausgeschaltet wird.

Programmiertipps:

- Zuerst schaltet man die Lampe an M2 für die Lichtschranke ein (Baustein *AUSGANG*).
- Danach wartet man eine Sekunde, damit der Fototransistor Zeit hat, auf das Licht zu reagieren (Baustein *WARTE*).
- Dann fragt man den Fototransistor E1 ab (Baustein *EINGANG*). Ist der Wert 1 (Lichtschranke nicht unterbrochen), soll der Fototransistor in einer Schleife dauernd abgefragt werden. Ist der Wert 0 (Lichtschranke unterbrochen) schaltet man den Motor M1 ein (Baustein *AUSGANG*) und nach 5 Sekunden (Baustein *WARTE*) wieder aus (Baustein *AUSGANG*). Danach soll wieder der Fototransistor abgefragt werden usw.

Erstelle nun diesen Ablauf:

5.6.4. Aufgabe

- 1) Da man stets darauf bedacht sein sollte Energie zu sparen soll das Programm nun so geändert werden, dass der Lüfter abschaltet sobald die Hände zurückgezogen werden.

Erstelle nun diesen Ablauf:

5.7. Projekt 6: Schiebetür

5.7.1. Lernziel

Die Schüler sollen mit der Fischertechnik Software eine Schiebetürsteuerung mit Verriegelungen realisieren.

5.7.2. Aufgabe

- 1) Baue das Modell der Schiebetür auf! (Bauanleitung Seite 10)

5.7.3. Erklärungen

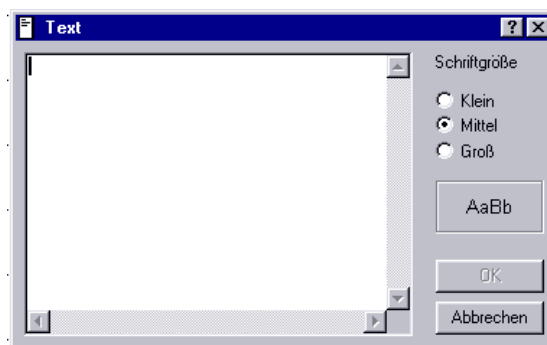
Um die Abläufe übersichtlicher zu gestalten, kann man mit dem Baustein *TEXT* verschiedene Bemerkungen in die Projekte einfügen.

- a) Baustein *TEXT*



Symbol im Bausteinfenster

Der hier eingetragene Text kann als Kommentar an einer beliebigen Stelle auf der Programmieroberfläche platziert werden.



Im Dialog kann außerdem die Schriftgröße eingestellt werden. Ein bereits eingefügter Text kann geändert werden. Man klickt dazu mit der rechten Maustaste auf den Text und aktiviert damit den Dialog, in welchem dann die Änderungen vorgenommen werden können.

5.7.4. Funktionsbeschreibung

Die Schiebetür soll nun so programmiert werden, dass sie sich beim Betätigen von E3 öffnet und nach 5 Sekunden wieder schließt.

Programmiertipps:

- Zuerst schließt du die Tür. Sie befindet sich dann in ihrer Ausgangsposition. Lasse dazu den Motor nach links laufen bis der Endschalter E1 "0" wird (Baustein *EINGANG*).
- Frage danach den Taster E3 ab (Baustein *EINGANG*). Wird er betätigt, öffne die Tür. Dazu lässt du den Motor nach rechts laufen, bis der Endschalter E2 auf "1" wechselt.
- Nach 5 Sekunden (Baustein *WARTE*) schließt du die Tür wieder.

Erstelle nun diesen Ablauf:

5.7.5. Aufgabe

- 1) Da man sich aber in der Tür einklemmen kann wenn sie schließt, soll das Programm folgendermaßen erweitert werden:
 - a. Die Tür darf nur geschlossen werden wenn die Lichtschranke nicht unterbrochen ist.
 - b. Die Tür soll sich wieder öffnen wenn während dem Schließen die Lichtschranke unterbrochen wird.
 - c. Die Tür soll sich, wenn sie bereits geschlossen ist, auch ohne Knopfdruck öffnen, sobald die Lichtschranke unterbrochen wird.

Programmiertipps:

- Schalte zuerst genau wie zuvor beim Händetrockner, die Lampe für die Lichtschranke ein und warte eine Sekunde, bevor der Ablauf weitergeht.
- Frage überall dort, wo es notwendig ist, den Fototransistor ab und öffne die Tür wenn der Fototransistor den Wert 0 liefert.

Erstelle nun diesen Ablauf:

5.8. Projekt 7: Temperaturregelung

5.8.1. Lernziel

Die Schüler sollen mit der Fischertechnik Software einen Regelkreis für die Temperatur aufbauen.

5.8.2. Aufgabe

- 1) Baue das Modell der Temperaturregelung zusammen! (Bauanleitung Seite 13)

5.8.3. Erklärungen

- a) Verarbeiten von Analogwerten mit Variablen

Neben den digitalen Eingängen bietet das Interface auch zwei analoge Eingänge EX und EY. Worin besteht dabei der Unterschied? Nun, mit digitalen Eingängen können nur zwei Zustände ausgewertet werden, nämlich "0" und "1" bzw. Kontakt geschlossen oder Kontakt geöffnet. Möchte man aber z.B. eine Temperatur messen, benötigt man einen Eingang, mit dem man mehrere Zustände (je nach Temperatur) auswerten kann. Hierfür dienen die beiden Analogeingänge des Interfaces. Der am Interface an EX und/oder EY angeschlossene Widerstand im Bereich von 0 ... 5 K Ω wird beim Intelligent Interface in einen Zahlenwert zwischen 0 und 1024 umgewandelt und kann dann im Steuerungsprogramm verwendet werden.

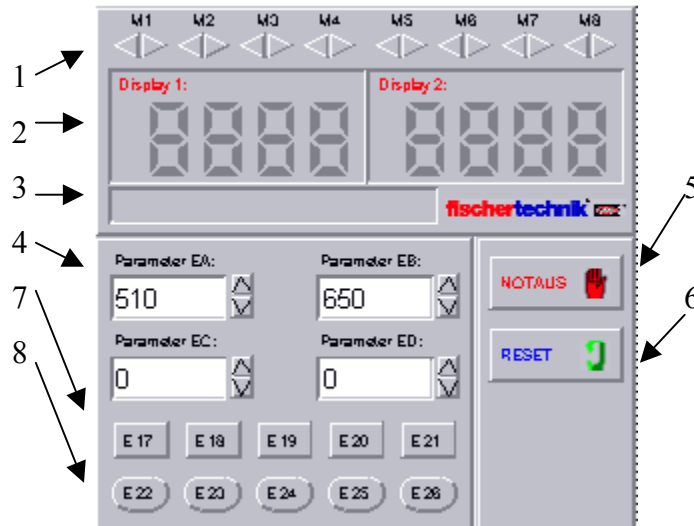
Bei der Erstellung des Ablaufplans ergibt sich allerdings ein kleines Problem! Wie lässt sich der Analogwert des Wärmesensors verwerten? Bei unserem bisher verwendeten Baustein *EINGANG* können wir nur EI bis E48, nicht aber EX oder EY einstellen. Probiere es einmal aus!

Mit dem Baustein *EINGANG* lässt sich nur eine "digitale" Abfrage erstellen! Hier hilft uns nur die Verwendung einer Variablen weiter. Wenn wir der Variablen den Wert des Analogeingangs zuweisen, können wir mit dem Baustein *VERGLEICH* unseren Steuerungsablaufplan erstellen. Zur Anzeige des Werts der Variablen braucht man einen neuen Baustein.

b) Baustein *TERMINAL*



Symbol im Bausteinfenster



Der Baustein *TERMINAL* wird zur Anzeige und Eingabe von Werten während des Online-Modus verwendet.

Ganz oben im Baustein wird dargestellt, welche der Interface-Ausgänge M1-M8 eingeschaltet sind und in welche Richtung sich die Motoren drehen (1).

Darunter befinden sich zwei Displays (2), in denen aktuelle Werte von Variablen, Analogwerten oder Konstanten angezeigt werden können. Welche Werte angezeigt werden sollen, muss über den Baustein *DISPLAY* festgelegt werden. Es kann nur der Wertebereich von -999 bis +9999 dargestellt werden.

Im Feld darunter (3) kann ein Text mit einer Länge von bis zu 17 Zeichen eingeblendet werden. Der Text wird im Baustein *MELDUNG* festgelegt.

Die 4 Terminalparameter EA-ED (4) sind Platzhalter für Werte, die man auch im Online-Modus verändern kann. So z. B. kann man im Baustein *POSITION* festlegen, dass ein Motor immer bis zum Endwert EA fahren soll. Den Wert EA stellt man dann im *TERMINAL*-Baustein ein und kann ihn, während das Steuerungsprogramm läuft, beliebig oft korrigieren.

Mit einem Mausklick auf den Terminalschalter *NOTAUS* (5) können im Online-Modus alle Ausgänge abgeschaltet werden. Das Steuerungsprogramm läuft weiter. Die Ausgänge werden wieder zugeschaltet, wenn der Schalter *NOTAUS* rückgesetzt wird. Mit dem Baustein *NOTAUS* kann die gleiche Funktion in Abhängigkeit von

einer Bedingung ausgeführt werden. Falls der Baustein *NOTAUS* zusammen mit dem *TERMINAL*-Baustein eingesetzt wird, müssen zur Ansteuerung der Ausgänge sowohl der Terminalschalter *NOTAUS* rückgesetzt sein, als auch die im *NOTAUS*-Baustein eingetragene Bedingung den Wert 0 liefern. Im Download-Modus des Intelligent Interface wird nur die Bedingung im Baustein *NOTAUS* ausgewertet.

Mit einem Mausklick auf den Terminalschalter *RESET* (6) können im Online-Modus alle Ausgänge abgeschaltet werden und das Steuerungsprogramm wird gestoppt. Das Steuerungsprogramm wird wie über den Befehl *START* im Menü *RUN* neu gestartet, wenn der Schalter *RESET* rückgesetzt wird. Mit dem Baustein *RESET* kann die gleiche Funktion in Abhängigkeit von einer Bedingung ausgeführt werden. Falls der Baustein *RESET* zusammen mit dem *TERMINAL*-Baustein eingesetzt wird, müssen, damit das Steuerungsprogramm gestartet werden kann, sowohl der Terminalschalter *RESET* rückgesetzt sein, als auch die im *RESET*-Baustein eingetragene Bedingung den Wert 0 liefern. Im Download-Modus des Intelligent Interface wird nur die Bedingung im Baustein *RESET* ausgewertet.

Der *TERMINAL*-Baustein besitzt Terminalschalter und Terminaltaster, die als zusätzliche 10 digitale Eingänge E17-E26 im Steuerungsprogramm abgefragt werden können. Die Terminalschalter E17-E21 (7) wechseln bei jedem Mausklick in den jeweils anderen Zustand. Die Terminaltaster E22-E26 (8) können durch Mausklick auf 1 gesetzt und wechseln nach Loslassen der Maustaste wieder auf 0. Die Terminalschalter E17-E21 können vor dem Start des Steuerungsprogramms in Init-Modus voreingestellt werden (gedrückt oder nicht gedrückt). Die Terminaltasten E22-E26 können nur im Online-Modus betätigt werden.

Nur wenn ein Projekt einen *TERMINAL*-Baustein enthält, können im Init-Modus Parameter EA-ED und die Terminalschalter E17-E21 gesetzt und im Projekt gespeichert werden. Auch im Download-Modus stehen die Werte der Parameter nur zur Verfügung, wenn der Baustein eingefügt wurde.

Die Parameter EA-ED und die Zustände der Terminalschalter E17-E21 können im Online-Modus auch in dem Bedienterminal, das über die Symbolleiste aufgerufen wird, geändert werden.

Sind mehrere *TERMINAL*-Bausteine im Projekt enthalten, so werden Eingaben von einem Baustein in alle anderen übernommen.

5.8.4. Funktionsbeschreibung

Dies wollen wir gleich an Hand Ventilatorsteuerung ausprobieren. Überlegen wir uns zunächst, wie eine solche Steuerung funktioniert. Als Erstes müssen wir mit einem Temperatur- oder Wärmesensor die Temperatur messen. Ist die Temperatur höher als die von uns eingestellte "kritische Temperatur", muss der Ventilator eingeschaltet werden. Sinkt die Temperatur wieder unter den kritischen Punkt, muss der Ventilator abgeschaltet werden. Der Vorgang Messen – Vergleichen – Ventilator aus-/einschalten beginnt dabei immer wieder von vorne, der Ablaufplan bildet also einen Kreis. In der Technik nennt man diese Steuerung deshalb auch einen **Regelkreis!**

Um die Temperatur messen zu können, benötigen wir einen Wärmesensor. Das ist ein temperaturabhängiger Widerstand. Je höher die Temperatur, desto geringer der Widerstand. Wir verwenden dazu den NTC-Widerstand 1,5 k Ω aus dem Baukasten. Schließe ihn zum Test an den Eingang EX des Interfaces an. Über die Interface-Diagnose kannst du feststellen, welchen Wert EX dann bei Zimmertemperatur besitzt. Wenn man den Wärmesensor erwärmt, wird man feststellen, dass der Wert von EX kleiner wird.

Programmiertipps:

- Du fragst in einem separaten Ablauf mit dem Baustein *ZUWEISUNG* ständig den Analogwert EX ab und speicherst den Wert in der Variablen VAR1.
- Beachte: Der Widerstandswert des NTC-Widerstands sinkt mit steigender Temperatur. Der obere Temperaturgrenzwert EA ist also der kleinste Wert der Variablen VAR1. Bei diesem Grenzwert soll das Gebläse einschalten. Der untere Temperaturgrenzwert EB ist der größte Wert von VAR1. Bei diesem Grenzwert soll die Heizung einschalten.
- Mit zwei Bausteinen *VERGLEICH* fragst du ständig ab, ob VAR1 unter dem kleinsten bzw. über dem größten Wert liegt und schaltest je nach Ergebnis das Gebläse oder die Heizung ein.
- Welchen Wert EX bei Zimmertemperatur besitzt, kannst du leicht mit der Interfacediagnose herausfinden. Du schaltest die Lampe M2 ein und siehst, wie der Wert nach unten geht. Dann schaltest du das Gebläse ein (so, dass es in Richtung Wärmesensor bläst) und beobachtest, wie der Wert ansteigt. Dementsprechend

wählt man die Grenzwerte EA und EB. Die Werte können bei verschiedenen NTC-Widerständen etwas schwanken.

Erstelle nun diesen Ablauf:

5.9. Projekt 8: Stanzmaschine

5.9.1. Lernziel

Die Schüler sollen lernen mit der Fischertechnik Software die Steuerung einer Stanzmaschine zu realisieren.

5.9.2. Aufgabe

- 1) Baue das Modell der Stanzmaschine zusammen! (Bauanleitung Seite 15)

5.9.3. Funktionsbeschreibung

Die Maschine soll ein Teil in einem Arbeitsgang mit 4 Hügen ausstanzen. Sie darf nur starten, wenn der Benutzer beide Taster E3 und E4 betätigt und gleichzeitig die Lichtschranke nicht unterbrochen ist. Wird die Lichtschranke unterbrochen, stoppt die Maschine. Es ertönt ein Warnsignal. Die Anzahl der Hüge soll am Terminal mit dem Parameter EA eingestellt werden können. Die Anzahl der bearbeiteten Teile soll am Terminal im Display 1 angezeigt werden.

Programmiertipps:

- Zuerst schaltest du die Lampe für die Lichtschranke ein. Dann fährst du die Maschine in ihre Ausgangsposition (der Exzenter betätigt in seinem oberen Umkehrpunkt den Taster E1).
- Mit einem Baustein *VERGLEICH* fragst du danach ab, ob E2 und E3 und E4 den Wert "1" aufweisen (Formel: E2 AND E3 AND E4).
- Ist dies der Fall, schaltest du den Motor M1 ein und lässt den Exzenter 4 Hüge ausführen. Aber nur, wenn die Lichtschranke nicht unterbrochen ist!
- Du zählst eine Variable VAR1 jedes Mal, wenn der Taster E1 betätigt wird, um eins hoch und fragst anschließend über den Baustein *VERGLEICH*, ob die bei EA eingestellte Anzahl der Hüge erreicht ist. Nach jedem Hub schaltest du den Motor ab und prüfst ob die Lichtschranke unterbrochen ist. Wenn ja, gibst du über den Baustein *BEEP* ein Warnsignal aus und brichst den Arbeitsgang ab.
- Danach folgt der nächste Arbeitsgang. Nach jedem erfolgreich ausgeführten Arbeitsgang zählst du eine Variable VAR2 hoch. Den Wert der Variable zeigst du im Display 1 an. Er entspricht der Anzahl der gefertigten Teile.

5.9.4. Erklärungen

a) Baustein *BEEP*

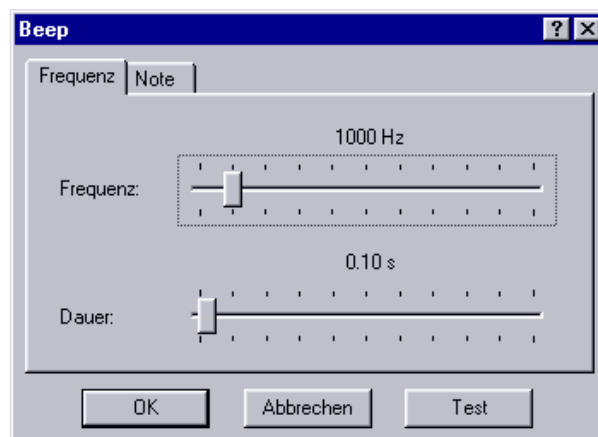


Symbol im Bausteinfenster



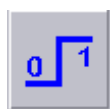
Der Baustein *BEEP* gibt einen Signalton über den Lautsprecher des PC's aus.

Beim Einfügen des Bausteins wird nachfolgender Dialog aufgerufen:

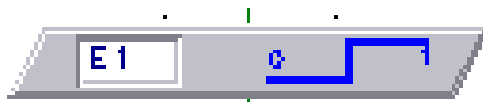


Im Dialog kann die Tonhöhe und Tondauer eingestellt werden. Die Tonhöhe kann entweder als Frequenz (50 – 10000 Hz) oder für Musiker auch als Note (c – h in 5 Oktaven) eingegeben werden. Die Tondauer kann 0 – 5 Sekunden betragen.

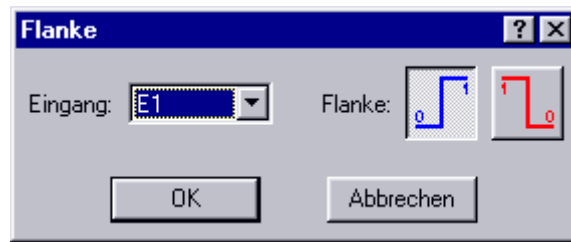
b) Baustein *FLANKE*



Symbol im Bausteinfenster



Beim Einfügen des Bausteins wird nachfolgender Dialog aufgerufen:



Der Baustein *FLANKE* wartet darauf, dass ein digitaler Eingang entweder von 0 auf 1 oder von 1 auf 0 wechselt. Diesen Übergang nennt man Flanke. Wurde der eingestellte Flankentyp erkannt, wird der Ablauf mit dem nächsten Baustein fortgesetzt.

Im Dialog des Bausteins kann eingestellt werden, auf welche Flanke (1 - 0 oder 0 - 1) an einem digitalen Eingang gewartet werden soll.

Die Art der ausgewählten Flanke wird im Bausteinsymbol dargestellt.

Erstelle nun diesen Ablauf:

5.10. Projekt 9: Parkhausschranke

5.10.1. Lernziel

Die Schüler sollen lernen mit der Fischertechnik Software die Steuerung einer Parkhausschranke mit Codierung zu realisieren.

5.10.2. Aufgabe

- 1) Baue das Modell der Parkhausschranke zusammen!(Bauanleitung Seite 18)

5.10.3. Funktionsbeschreibung

Durch Betätigen des Tasters E3 soll die Schranke geöffnet werden. Ist die Schranke offen, leuchtet die Ampel grün. Erst wenn die Lichtschranke passiert wurde, springt die Ampel auf Rot und die Schranke schließt wieder.

Programmiertipps:

- Schreibe zum Öffnen und Schließen der Schranke jeweils ein Unterprogramm "Öffnen" und "Schließen".
- Schalte im Ablauf als Erstes die Lampe für die Lichtschranke ein und danach die Ampel auf Rot.
- Schließe die Schranke. Sie befindet sich dann in ihrer Ausgangsposition.
- Nach Betätigen des Tasters E3 (Baustein Flanke) wird die Schranke geöffnet. Danach schaltet die Ampel auf Grün.
- Wird die Lichtschranke erst unterbrochen und dann wieder geschlossen/1-0 und 0-1 Flanke an E4) soll die Ampel auf Rot wechseln und die Schranke schließen.
- Dann beginnt der Ablauf von vorn.

Erstelle nun diesen Ablauf:

5.10.4. Funktionsbeschreibung

Das Parkhaus soll am Eröffnungstag für prominente Gäste freigehalten werden. Dazu erhalten die Parkberechtigten eine geheime Zahlenkombination mit 3 Ziffern. Sie müssen drei der Eingänge E22-E26 am Baustein *TERMINAL* in der richtigen Reihenfolge drücken. Nur dann darf sich die Schranke öffnen. Programmiere diese Funktion in einem Unterprogramm "Code". Am Terminal soll eine Meldung ausgegeben werden ob der Code richtig oder falsch war.

Der Code soll lauten: E26-E22-E24.

Programmiertipps:

- Das Unterprogramm "Code" kannst du ganz unterschiedlich aufbauen: Am Einfachsten ist es, mit dem Baustein *FLANKE* die drei Eingänge E26, E22 und E24 nacheinander abzufragen. Wurden die drei Tasten gedrückt, erscheint die Meldung "Code richtig" und du verlässt das Unterprogramm. Bei dieser Lösung können dann allerdings zwischendurch auch falsche Tasten gedrückt werden, ohne, dass das Programm dies bemerkt.
- Willst du die Codeeingabe so absichern, dass wirklich nur drei Tasten gedrückt werden können, und wenn eine davon falsch ist, die Meldung "Code falsch" erscheint, wird dies etwas umfangreicher. Du verwendest dann den Baustein *EINGANG* und fragst im Unterprogramm alle 5 Eingänge E26-E22 ab.
- Als erstes soll ja E26 gedrückt und wieder losgelassen werden. Wird die richtige Taste gedrückt, geht der Ablauf nach unten weiter. Wird eine falsche Taste gedrückt (E25-E22) wird eine Variable VAR1 hochgezählt und der Ablauf erst dann fortgeführt.
- Diese Abfolge benötigst du insgesamt dreimal untereinander. Links steht immer die richtige Taste, rechts davon die falschen. Jedes Mal, wenn eine falsche Taste gedrückt wird, wird die Variable VAR1 um 1 erhöht.
- Am Ende fragst du mit dem Baustein *VERGLEICH* ab, ob VAR1=0 ist, denn nur dann wurde der richtige Code eingegeben. Dann kannst du noch eine Meldung ausgeben und unterschiedliche Töne erzeugen. Danach verlässt du das Unterprogramm über zwei verschiedene Bausteine *SUBOUT*.

Erstelle nun diesen Ablauf:

5.11. Projekt 10: Schweißroboter

5.11.1. Lernziel

Die Schüler sollen lernen mit der Fischertechnik Software einen Schweißroboter zu steuern.

5.11.2. Aufgabe

- 1) Baue das Modell des Schweißroboters zusammen!(Bauanleitung Seite 21)

5.11.3. Funktionsbeschreibung

Der Roboter soll an drei verschiedenen Positionen jeweils an einem Metallgehäuse den Deckel mit einem Schweißpunkt fixieren. Die Schweißelektrode wird durch eine Linsenlampe simuliert, die drei Metallgehäuse durch gelbe Bausteine. Der Roboter soll die drei Positionen nacheinander anfahren und an jeder Position eine Schweißung durchführen. Danach soll er in seine Ausgangsposition zurückkehren und wieder von vorne beginnen. Da der Roboter auch unabhängig vom PC im Downloadbetrieb arbeiten muss, soll über Taster E8 ein Reset ausgelöst werden können, d.h. das Programm nach Drücken und wieder Loslassen des Tasters von vorne starten.

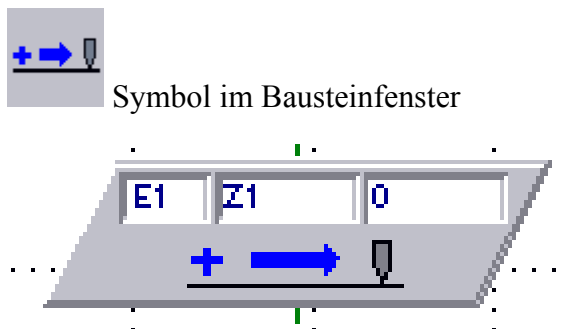
Programmiertipps:

- Fahre den Roboter zunächst in seine Ausgangsposition (M 1 links bis E1 gedrückt).
- Fahre dann in die Position 1. Verwende dazu den Baustein *POSITION* und zähle die Impulse am Eingang E2. Wie viele Zählimpulse notwendig sind, bis die Position erreicht ist, musst du ausprobieren. Am Besten verwendest du als Endwert im Baustein *POSITION* den Parameter EA. Dann kannst du den Wert ständig am *TERMINAL* ändern.
- Für den Schweißvorgang schreibst du ein Unterprogramm "Schweißen". In diesem Unterprogramm schaltest du die Linsenlampe (M2) ein und gleich wieder aus. Diesen Vorgang wiederholst du ein paar Mal, indem du eine Schleife programmierst und eine Variable VAR1 hoch zählst, bis deren Wert z.B. 20 ist. Dann erst verlässt du das Unterprogramm. Vergiss nicht, die Variable vorher wieder auf Null zu setzen (Baustein *ZUWEISUNG*).
- Verfahre genauso mit den Positionen 2 und 3. Verwende als Endwerte in den Bausteinen Position die Parameter EB und EC.

- Für den Reset verwendest du natürlich den Baustein *RESET*, den du ohne Verbindung zu anderen Bausteinen auf der Programmieroberfläche platzierst (Formel die im Dialog eingegeben wird: E8).
- Teste das Programm im Online-Modus, bis du alle Positionen am *TERMINAL* richtig eingestellt hast. Lade es danach auf das Interface (Run-Download) und lass es im Download-Modus ablaufen. Es funktioniert dort genau wie im Online-Modus. Allerdings kannst du, da das Interface keine Verbindung mehr zum Rechner hat, die Terminalparameter nicht mehr verändern (d.h. verändern kannst du sie am Bildschirm schon, nur reagiert der Roboter nicht mehr darauf).
- Nicht vergessen: Um nach einem Download in den Online-Modus zurückzukehren, muss man die Stromversorgung am Intelligent Interface für ca. 3 Sekunden unterbrechen.

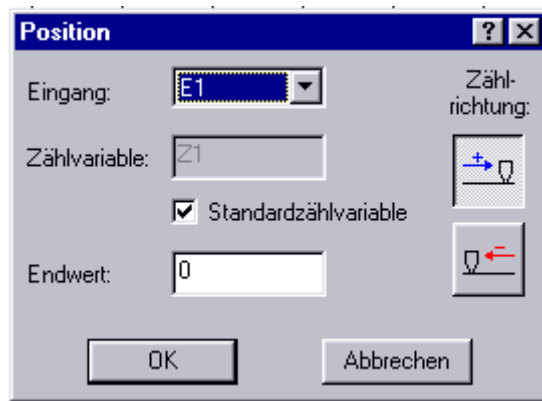
5.11.4. Erklärungen

a) Baustein *POSITION*



Der Baustein *POSITION* zählt Flanken an einem digitalen Eingang: Wurde der zugewiesene Endwert erreicht, kommt der nächste Baustein an die Reihe. Gezählt wird jede Flanke am Eingang, d.h. wird ein Taster gedrückt und wieder losgelassen, werden zwei Flanken gezählt.

Beim Einfügen des Bausteins wird nachfolgender Dialog aufgerufen:



Der aktuelle Zählwert wird in einer Zählvariablen gespeichert. Aktiviert man im Dialog den Schalter "Standardzählvariable" wird dem ausgewählten digitalen Zähleingang E1-E16 automatisch eine Zählvariable Z1-Z16 zugeordnet (bei E1 wird Z1 zugewiesen, bei E2 Z2 usw.). Dann muss man nicht lange überlegen, welche Variable man für diesen Zähleingang verwenden soll, und ob diese schon anderweitig vergeben ist.

Die 16 Zählvariablen können auch von allen anderen Bausteinen, in denen Variablen verwendet werden können, verarbeitet werden. Will man trotzdem lieber eigene Zählvariablen vergeben, deaktiviert man die Funktion "Standardzählvariable" und trägt im Feld "Zählvariable" eine der Variablen VAR1 bis VAR99 ein, die in LLWin zur Verfügung stehen.

Schließlich stellt man im Dialog noch die Zählrichtung ein. Bei + wird der Wert der Zählvariable bei jeder Flanke um 1 erhöht, bei - um 1 verringert.

Im Bausteinsymbol werden der Zähleingang, die Zählvariable, der Endwert und die Zählrichtung dargestellt.

Am Häufigsten wird der Baustein *POSITION* dazu verwendet, einen Motor in eine bestimmte Position zu fahren. Man schaltet den Motor ein, zählt über einen Taster, der von einem Impulsrad gedrückt wird, die Flanken und schaltet den Motor ab, nachdem der gewünschte Endwert erreicht wurde.

Will man eine Zählvariable wieder auf Null setzen, verwendet man dazu den Baustein *ZUWEISUNG* und trägt dort z.B. $Z1 = 0$ ein.

b) Baustein *RESET*



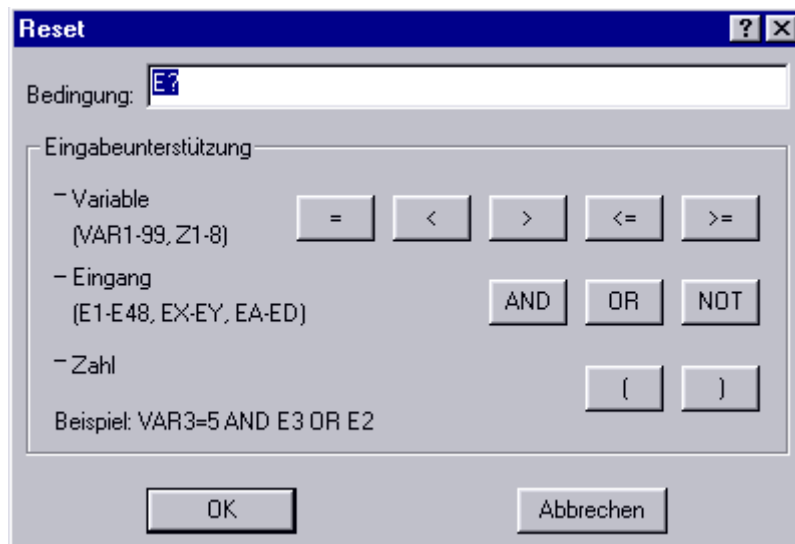
Symbol im Bausteinfenster



Der Baustein *RESET* setzt alle Abläufe eines Projekts zurück auf den Baustein *START*, sobald die zugewiesene Bedingung erfüllt ist. Außerdem werden alle Ausgänge abgeschaltet.

Der Baustein *RESET* wird ohne Verbindungslinien zu anderen Bausteinen auf der Programmieroberfläche platziert. In einem Projekt darf nur ein Baustein *RESET* enthalten sein. Sind mehrere *RESET*-Bausteine eingefügt worden, so werden sie, um auf den Fehler hinzuweisen, im Init-Modus violett dargestellt.

Beim Einfügen des Bausteins gibt man im nachfolgend dargestellten Dialogfenster eine maximal 40 Zeichen umfassende Bedingung ein, bei der *RESET* ausgelöst werden soll.



Hinweis:

Wurde nach der Eingabe der Bedingung eine Fehlermeldung angezeigt, können mögliche Ursachen unter "Fehlermeldungen in Berechnungen und Bedingungen" nachgelesen werden.

Die einfachste Reset-Bedingung kann beispielsweise E1 heißen. Dann wird *RESET* ausgelöst, sobald diese Bedingung erfüllt ist, E1 also 1 wird. Das Steuerungsprogramm startet wieder neu, sobald die Bedingung nicht mehr zutrifft, E1 also den Wert 0 annimmt.

So kann man über einen digitalen Eingang ein Projekt, das auf dem Intelligent Interface im Download-Modus (ohne Verbindung zum PC) läuft, anhalten und neu starten, ohne dass man es erneut laden muss.

Erstelle nun diesen Ablauf: