

Computing Starter

- Begleitheft
- Activity booklet
- Manuel d'accompagnement
- Cuaderno adjunto
- Folheto

(D) Seite 1-10

Computing Starter

Das Begleitheft zum Baukasten

Beschreibt genau, wie wie du vorgehen musst.
Enthält zahlreiche Programmieraufgaben.

(GB+USA) Page 11-20

Computing Starter

The Activity Booklet for the Construction Kit

Describes exactly what you must do.
Contains numerous programming tasks.

(F) Pagina 21-30

Computing Starter

Le manuel d'accompagnement du jeu de construction

Vous explique en détail comment procéder. Propose un grand nombre de commandes à programmer.

(E) Página 31-40

Computing Starter

El cuaderno adjunto para el kit de construcción

Describe exactamente cómo tienes que proceder.
Contiene numerosas tareas de programación.

(P) Página 41-50

Computing Starter

O auxiliar do kit

Descreve detalhadamente como você deve proceder.
Contém numerosas tarefas de programação.

1. Einleitung	Seite 2
2. Der Einstieg	Seite 2
2.1 Der Baukasten	Seite 2
2.2 Interface und Software	Seite 2
2.3 Erste Schritte beim Programmieren	Seite 3
3. Programmieraufgaben Teil 1	Seite 3
3.1 Händetrockner	Seite 3
3.2 Ampel	Seite 3
3.3 Schiebetür	Seite 4
4. Programmieraufgaben Teil 2	Seite 5
4.1 Temperaturregelung	Seite 5
4.2 Stanzmaschine	Seite 5
4.3 Parkhausschranke	Seite 6
4.4 Schweißroboter	Seite 8
5. Wie geht es weiter?	Seite 9

1. Einleitung

Hallo PC-Freak und fischertechnik-Fan. Herzlich willkommen in unserer „Computing-Welt“. Unter dem Begriff „Computing“ verstehen wir bei fischertechnik das Programmieren und Steuern von Modellen über den PC. Der Baukasten Computing Starter bildet den optimalen Einstieg in dieses Thema. Du kannst 8 verschiedene Modelle, vom Händetrockner über eine Parkhauschranke bis hin zum Schweißroboter, mit Hilfe der Bauanleitung in kürzester Zeit aufbauen. Über das Intelligent Interface Art.-Nr. 30402 verbindest du die Modelle mit dem PC. Schließlich programmierst du die Modelle schnell und einfach mit der grafischen Programmiersoftware LLWin 3.0.

Das vorliegende Begleitheft soll dir helfen, dich in der Computing-Welt schnell zurecht zu finden. Es zeigt dir zunächst, wie du am Anfang vorgehen solltest und was du nacheinander tun musst. Des Weiteren findest du hier Programmieraufgaben für alle Modelle des Baukastens. Natürlich fehlen zu diesen Aufgaben auch Tipps für die richtige Lösung nicht. Es wird genau beschrieben, wie du die Modelle mit der Software LLWin 3.0 programmierst. Du wirst sehen, das macht unheimlich Spaß. Also nichts wie los.

2. Der Einstieg

Was ist denn nun alles im Baukasten drin?

Im „Computing Starter Pack“, dem Komplettpaket mit Interface und Software, ist (fast) alles enthalten was du brauchst. Nur die Stromversorgung ist noch zusätzlich erforderlich.

2.1 Der Baukasten

Zunächst einmal findest du zahlreiche fischertechnik Bausteine, Motor, Lampen und Sensoren, sowie eine farbige Bauanleitung zum Bau von 8 verschiedenen Modellen. Damit beschäftigen wir uns zunächst. Wenn du die Bausteine alle ausgepackt hast, musst du einige Komponenten zuerst montieren, bevor du loslegen kannst (z. B. Kabel und Stecker). Welche das genau sind, ist in der Bauanleitung unter „Montagehilfen und Hinweise“ beschrieben. Erledige das am Besten gleich als Erstes.

Die wichtigsten Bauteile:



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.



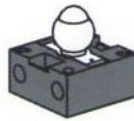
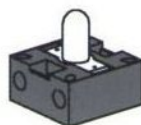
Getriebe: Auf den Motor wird ein Getriebe gesteckt, das die Drehzahl heruntersetzt. Die Untersetzung beträgt einschließlich der Motorschnecke und dem Zahnrad mit der Abtriebswelle 64,8:1.

Lampen:

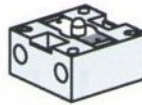
Im Baukasten sind zwei verschiedene Lampen enthalten:

Kugellampe

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



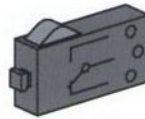
Linse Lampe: In diese Lampe ist eine Linse eingearbeitet, die das Licht bündelt. Sie sieht der Kugellampe sehr ähnlich. Du musst aufpassen, dass du sie nicht verwechselst. Die Linse Lampe benötigst du zum Bauen einer Lichtschranke.



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 musst du auf die richtige Polung achten: Rot = Plus



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 kannst du 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.



NTC-Widerstand: Bei diesem Bauteil handelt es sich um einen Wärmesensor, mit dem man Temperaturen messen kann. Bei 20°C beträgt der Widerstand 1,5kΩ (kilo-Ohm).

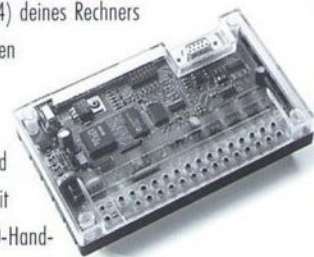
NTC bedeutet **N**egativer **T**emperatur **C**oeffizient. Das heißt einfach, dass der Widerstandswert mit steigender Temperatur sinkt.

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.

LLWin 2.2 Interface und Software



Bevor du anfängst Modelle zu bauen und Programme zu erstellen, musst du die Steuerungssoftware LLWin 3.0 auf deinem PC installieren. Danach schließt du das Intelligent Interface Art.-Nr. 30402 an einer freien seriellen Schnittstelle (COM1-COM4) deines Rechners an. Wie das geht, ist im LLWin-Handbuch in den Kapiteln 1 und 2 ausführlich beschrieben. Gehe jetzt genau danach vor, und es dürfte dir ohne Schwierigkeiten gelingen Software und Interface zum Laufen zu bekommen. Die CD mit der Software findest du übrigens im LLWin 3.0-Handbuch „versteckt“. Für das Interface benötigst du noch eine fischertechnik Stromversorgung mit einer Spannung von 9V DC und einer Stromstärke von 1000mA (z. B. Energy Set Art.-Nr. 30182 oder Accu Set Art.-Nr. 34969).



Da viele Fischertechniker eine dieser Komponenten bereits besitzen, ist auch im Komplettpaket „Starter Pack“ keine Stromversorgung enthalten. Viel Erfolg jetzt beim Installieren und Anschließen von Software und Interface. Danach geht es hier weiter.

2.3 Erste Schritte beim Programmieren

Nachdem nun Hard- und Software funktionieren, geht es endlich los mit dem Programmieren. Auch dazu verwenden wir erst einmal wieder das LLWin 3.0-Handbuch. Einen besseren Einstieg in die Programmierung als dort in den Kapiteln 3 und 4 beschrieben, gibt es nicht. Deshalb greifen wir an dieser Stelle einfach darauf zurück. Arbeite diese beiden Kapitel sorgfältig durch.

Du wirst auch feststellen, dass in Kapitel 3.5, „Testen des ersten Steuerungsprogramms“, auf das Modell „Motorsteuerung“ des Baukastens Computing Starter verwiesen wird.

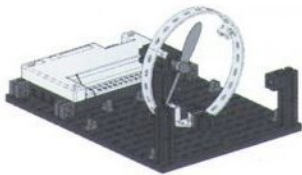
Baue dieses Modell mit Hilfe der Bauanleitung dann auf und teste dein erstes Programm damit. Viel Spaß.



3. Programmieraufgaben Teil 1

Nachdem du die Kapitel 3 und 4 des LLWin 3.0-Handbuchs durchgelesen hast, kannst du jetzt schon einige Modelle des Baukastens Computing Starter programmieren. Deshalb wollen wir auch sofort loslegen. Immer, wenn du ein Modell fertig gebaut und verkabelt hast, prüfe mit Hilfe der Interfacediagnose, ob am Interface alle Aus- und Eingänge richtig angeschlossen sind und die Sensoren, Motoren und Lampen richtig funktionieren.

3.1 Händetrockner (siehe Bauanleitung Seite 6)



In deiner Schule wurden auf der Toilette neben den Waschbecken neue Händetrockner installiert. Diese sind mit einer Lichtschranke versehen, über die man den Lüfter ein- und ausschalten kann. Baue zunächst das Modell auf.

Aufgabe 1: Der Händetrockner soll nun so programmiert werden, dass, sobald die Lichtschranke unterbrochen wird, der Lüfter ein- und nach 5 Sekunden wieder ausgeschaltet wird. Erstelle dazu ein neues Projekt in LLWin 3.0.

Programmier-Tipps:

- Zuerst schaltest du die Lampe an M2 für die Lichtschranke ein (Baustein AUSGANG).
- Danach wartest du eine Sekunde, damit der Fototransistor Zeit hat, auf das Licht zu reagieren (Baustein WARTE).
- Dann fragst du 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) schaltest du den Motor M1 ein (Baustein AUSGANG) und nach 5 Sekunden (Baustein WARTE) wieder aus (Baustein AUSGANG). Danach soll wieder der Fototransistor ab-gefragt werden usw.

Starte dein Projekt mit Run – Start und überprüfe, ob das Programm so funktioniert wie es soll. Wenn ja, dann bist du auf dem besten Weg, ein professioneller LLWin-Programmierer zu werden.

Funktioniert es noch nicht, versuche herauszubekommen, woran es liegt:

- In Run – Init kannst du prüfen ob alle Bausteine mit Linien verbunden und die Eingaben in den Bausteinen richtig sind. Fehlt an einem Bausteine eine Linie oder eine Eingabe, wird der Baustein violett dargestellt.
- Während das Programm im Online-Modus läuft (Run – Start) kannst du dem Programmablauf anhand der rot markierten Bausteine folgen. Damit kannst du ebenfalls schnell erkennen, wo sich ein Fehler eingeschlichen hat (z. B. wenn der Ablauf am Baustein EINGANG nicht weiterläuft, weil vielleicht E2 anstatt E1 abgefragt wird).
- Falls du dein Programm dann immer noch nicht zum Laufen bekommst, kannst du dir auch das fertige Beispielprojekt Händetrockner1.mdl aus dem Unterverzeichnis „Computing Starter“ deines LLWin-Verzeichnisses laden und mit deinem Ablauf vergleichen. Spätestens jetzt wirst du erkennen, wo du einen Fehler eingebaut hast.

Nachdem dieses erste Hürde genommen ist, wollen wir die Aufgabenstellung etwas verändern:

Aufgabe 2:

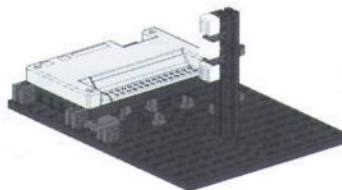
Dem Rektor, der stets darauf bedacht ist Energie zu sparen, gefällt es nicht, dass der Händetrockner immer noch eine gewisse Zeit weiterläuft, obwohl die Hände bereits trocken sind. Er fordert dich auf, das Programm so zu gestalten, dass der Lüfter abschaltet, sobald die Hände zurückgezogen werden. Kein Problem für dich, oder?

Programmier-Tipps:

- Du fragst wieder den Fototransistor E1 ab (Baustein EINGANG), ist der Wert 0, schaltest du den Motor M1 ein (Baustein AUSGANG), ist der Wert 1, schaltest du den Motor M1 aus (Baustein AUSGANG) usw.
- Auch zu dieser Aufgabe gibt es für den Notfall ein fertiges Projekt Händetrockner2.mdl.

3.2 Ampel (siehe Bauanleitung Seite 8)

Vor eurem Haus wurde eine Ampel aufgestellt. Da der Monteur von der Ampelfirma sehr unter Zeitdruck steht, bietest du ihm an, die Programmierung der Ampelsteuerung für ihn zu übernehmen.



Der Mann erklärt dir, wie die Steuerung funktionieren soll. Baue aber zuerst das Modell auf.

Aufgabe 1:

Die Ampel soll im Normalfall auf Grün stehen. Wird der Taster E1 von einem Fußgänger gedrückt, soll die Ampel 3 Sekunden später auf gelb und nach weiteren 4 Sekunden auf Rot wechseln. Die Rot-phase soll 10 Sekunden dauern, die anschließende Rot-Gelb-Phase 3 Sekunden, dann soll es wieder Grün werden.

Programmier-Tipps:

- Die verschiedenen Lampen gehören zu folgenden Interfaceausgängen: Grün = M3, gelb = M2, rot = M1. Schalte die Lampen einfach so hintereinander ein und aus (Baustein AUSGANG), dass der geforderte Ablauf zustande kommt.
- Zum Abfragen des Tasters E1 verwendest du den Baustein EINGANG.
- Fertiges Projekt: Ampel1.mdl

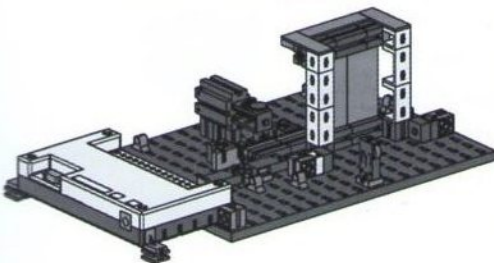
Aufgabe 2:

Am nächsten Tag ruft dich der Monteur der Ampelfirma an. Er hat vergessen dir zu sagen, dass sich in dem Schaltkasten auf dem Gehweg ein Schalter E2 befindet, der die Ampel auf gelbes Blinklicht schalten soll, sobald er betätigt wird. Du sicherst dem Monteur zu, diese Funktion noch schnell in dein Programm zu integrieren.

Programmier-Tipps:

- Frage mit einem weiteren Baustein EINGANG zusätzlich zum Taster E1 den Taster E2 ab. Wird E2 gedrückt, verzweigt du den Ablauf zum Blinklicht, wird E1 gedrückt, läuft die Ampelsteuerung wie in Aufgabe 1 ab.
- Das Blinklicht programmierst du, indem du die Lampe M2 ein- und nach 0,5 Sekunden wieder ausschaltest. Schreibe dafür ein Unterprogramm „Blinklicht“. Wie das geht, hast du ja in Kapitel 4 des LLWin-Handbuchs gelernt.
- Überlege dir genau, an welcher Stelle des Ablaufs du E2 abfragst, damit das Umschalten auf Blinklicht nicht nur einmal beim Starten des Programms, sondern während jeder Grünphase möglich ist.
- Das fertige Projekt heißt übrigens Ampel2.mdl. Aber nicht gleich nachschauen! Versuche erst einmal selbst dein Glück

3.3 Schiebetür (siehe Bauanleitung Seite 10)



Der Supermarkt, in dem du stundenweise hilfst die Regale einzuräumen, hat eine neue Eingangstür bekommen. Für diese muss jetzt noch die Steuerungssoftware erstellt werden. Der Filialleiter weiß, dass

du Experte im Programmieren bist und bittet dich, das zu übernehmen. Zunächst baust du aber das Modell auf.

Aufgabe 1:

Wenn der Taster E3 gedrückt wird, soll sich die Tür öffnen und nach 5 Sekunden wieder schließen.

Programmier-Tipps:

- Zuerst schließt du die Tür. Sie befindet sich dann in ihrer Ausgangsposition. Lasse dazu den Motor M1 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 M1 nach rechts laufen, bis der Endschalter E2 auf „1“ wechselt.
- Nach 5 Sekunden (Baustein WARTE) schließt du die Tür wieder (Schleife zurück zum Anfang).
- Fertiges Projekt: Schiebetür1.mdl

Aufgabe 2:

Deine Türsteuerung funktioniert hervorragend. Als sich jedoch der erste Kunde ein Bein in der Tür einklemmt, weil er genau in dem Moment durch die Tür ging, als sie geschlossen wurde, beschließt du, das Programm noch etwas zu verbessern. Die Tür verfügt nämlich über eine Lichtschranke, die verhindern soll, dass die Tür geschlossen wird, wenn gerade jemand durchgeht. Du willst das Programm so erweitern, dass

1. die Tür nur geschlossen wird, wenn die Lichtschranke nicht unterbrochen ist.
2. die Tür sich wieder öffnet, wenn während des Schließens die Lichtschranke unterbrochen wird
3. die Tür sich, wenn sie bereits geschlossen ist, auch ohne Knopfdruck öffnet, sobald die Lichtschranke unterbrochen wird.

Da hast du dir was vorgenommen! Aber es ist eigentlich gar nicht so schwierig. Vielleicht musst du etwas nachdenken. Schau aber nicht gleich das fertige Projekt an. Etwas Gehirntraining hat noch keinem geschadet!

Programmier-Tipps:

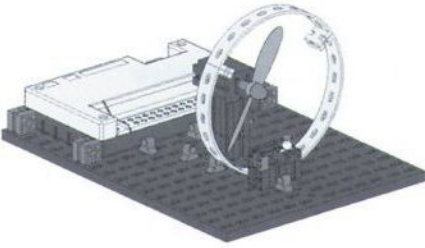
- Schalte zuerst, genau wie zuvor beim Händetrockner, die Lampe für die Lichtschranke ein und warte eine Sekunde, bevor der Ablauf weitergeht.
- Frage überall dort, wo es notwendig ist, den Fototransistor ab und öffne die Tür, wenn der Fototransistor den Wert 0 liefert.
- Fertiges Projekt: Schiebetür2.mdl

Geschafft! Dein Chef ist stolz auf dich! Die Tür funktioniert jetzt tadellos und absolut sicher. Er verdoppelt deinen Stundenlohn.

4. Programmieraufgaben Teil 2

Bevor du dich an den zweiten Teil der Programmieraufgaben wagst, solltest du wieder das LLWin 3.0-Handbuch zur Hand nehmen. Arbeite dort die Kapitel 5 und 6 sorgfältig durch. Langsam werden die Programmieraufgaben etwas anspruchsvoller. Wir verwenden Variablen, mehrere gleichzeitige Abläufe sowie die Analogeingänge EX und EY des Intelligent Interface. Aber wenn du diese beiden Kapitel des LLWin 3.0-Handbuchs aufmerksam liest, wird es dir später leichtfallen damit umzugehen. Außerdem kennst du danach alle Bausteine, die in LLWin 3.0 zur Verfügung stehen. Bis dann.

4.1 Temperaturregelung (siehe Bauanleitung Seite 13)



Bei euch zu Hause wurde eine neue Klimaanlage installiert. Natürlich hast du den Installateur sofort gefragt, wie die Temperaturregelung funktioniert. Er hat dir bereitwillig erklärt, dass ein Temperaturfühler stetig die vorhandene

Temperatur misst. Sobald ein oberer Grenzwert überschritten wird, schaltet die Kühlung ein. Wird hingegen ein unterer Grenzwert unterschritten, schaltet die Kühlung aus und die Heizung ein. Nun willst du anhand des Modells „Temperaturregelung“ versuchen, ebenfalls einen solchen Regelkreis zu programmieren. Baue zuerst das Modell.

Aufgabe:

Die Heizung wird simuliert durch die Linsenlampe M2. Als „Kühlaggregat“ dient das Gebläse am Ausgang M1. Zur Temperaturmessung verwenden wir den NTC-Widerstand am Eingang EX. Programme das Modell so, dass oberhalb einer bestimmten Temperatur die Heizung aus- und das Gebläse einschaltet. Dieses soll so lange kühlen, bis ein unterer Grenzwert erreicht ist. Dann soll das Gebläse aus- und die Heizung eingeschaltet werden. Die beiden Grenzwerte sollen am Baustein TERMINAL mit den Parametern EA und EB eingestellt werden können. Der Wert für die aktuelle Temperatur soll im Display 1 des TERMINALS angezeigt werden.

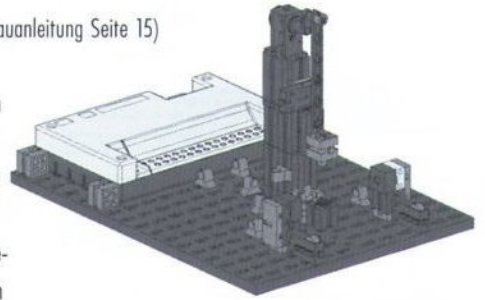
Programmier-Tipps:

- Du fragst in einem separaten Ablauf mit dem Baustein ZUWEISUNG ständig den Analogwert EX ab und speicherst den Wert in der Variablen VART.
- Beachte: Der Widerstandswert des NTC-Widerstands sinkt mit steigender Temperatur. Der obere Temperaturgrenzwert EA ist also der kleinste Wert der Variablen VART. Bei diesem Grenzwert soll das Gebläse einschalten. Der untere Temperaturgrenzwert EB ist der größte Wert von VART. Bei diesem Grenzwert soll die Heizung einschalten.
- Mit zwei Bausteinen VERGLEICH fragst du ständig ab, ob VART 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 wählst du deine Grenzwerte EA und EB. Die Werte können bei verschiedenen NTC-Widerständen etwas schwanken.
- Fertiges Projekt: Temperaturregelung.mdl

4.2 Stanzmaschine (siehe Bauanleitung Seite 15)

Die Werkstatt nebenan hat in eine hochmoderne Maschine zum Ausstanzen von Blechteilen investiert. Die Maschine ist bereits aufgestellt. Leider soll der Programmierer, der die Anlage in Betrieb nimmt, erst in zwei Wochen erscheinen. Da die Werkstatt die Maschine sehr dringend benötigt, fragst dich der Besitzer, ob du nicht in der Lage wärst, die Maschine zum Laufen zu bringen. Da du inzwischen schon ziemlich viel Erfahrung beim Programmieren gesammelt hast, versprichst du ihm, die Anlage bis Morgen zum Laufen zu bringen.



Baue aber zuerst das Modell Stanzmaschine mit Hilfe der Bauanleitung auf.

Aufgabe:

Die Maschine soll ein Teil in einem Arbeitsgang mit 4 Hüben ausstanzen. Sie darf nur starten, wenn der Bediener beide Taster E3 und E4 betätigt (sog. Zweihandbedienung) und gleichzeitig die Lichtschranke nicht unterbrochen ist. Wird die Lichtschranke während eines Arbeitsganges unterbrochen, stoppt die Maschine. Es ertönt ein Warnsignal. Die Anzahl der Hübe soll am TERMINAL mit dem Parameter EA eingestellt werden können. Die Anzahl der bearbeiteten Teile soll am TERMINAL im Display 1 angezeigt werden.

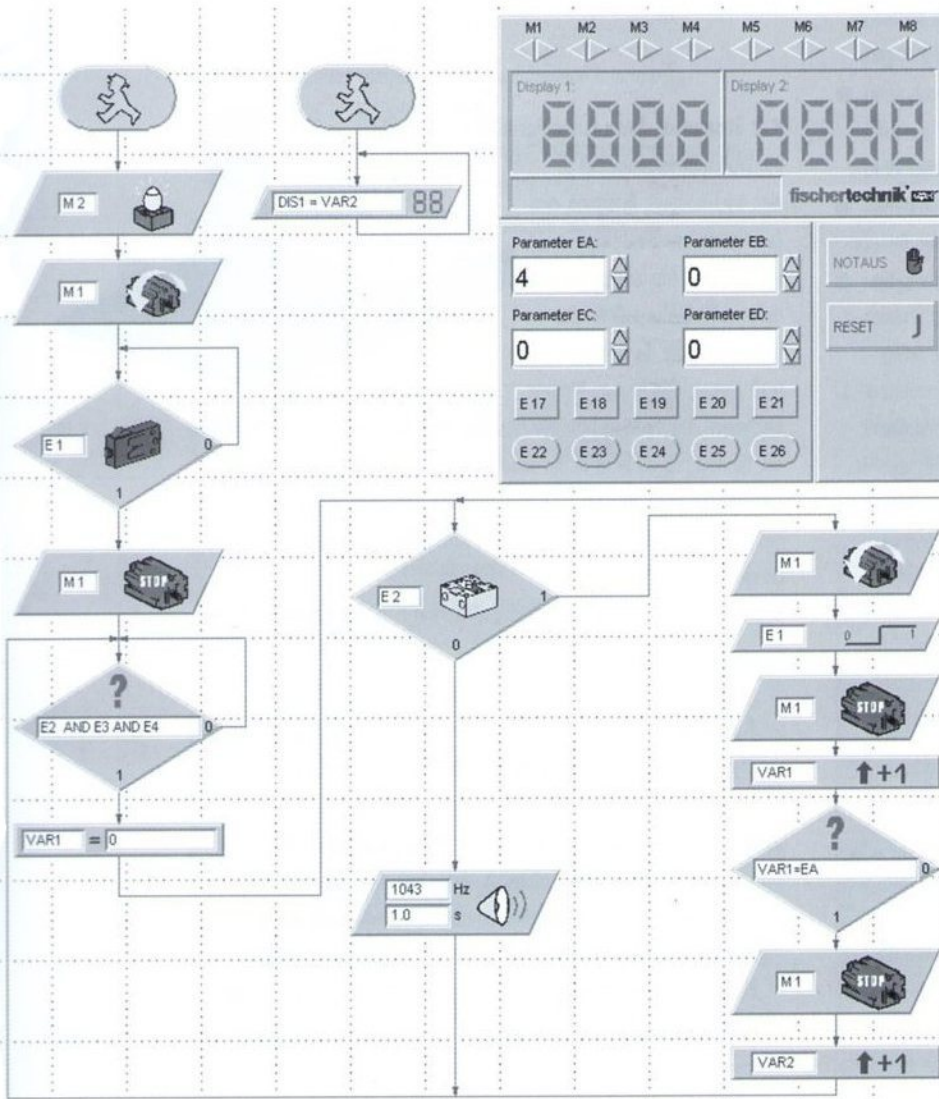
Programmier-Tipps:

Da hast du dich auf etwas eingelassen. Aber keine Sorge, mit LLWin 3.0 ist das gar nicht so schwierig:

- 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; siehe auch LLWin-Handbuch Kapitel 6)
- Ist dies der Fall, schaltest du den Motor M1 ein und lässt den Exzenter 4 Hübe ausführen. Aber nur, wenn die Lichtschranke nicht unterbrochen ist!
- Du zählst eine Variable VART 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 Hübe erreicht ist. Nach jedem Hub schaltest du den Motor ab und prüfst

ob die Lichtschranke unterbrochen ist. Wenn ja, gibst du über der 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 Variablen zeigst du im Display 1 an. Er entspricht der Anzahl der gefertigten Teile.
- Fertiges Projekt: Stanzmaschine.mdl

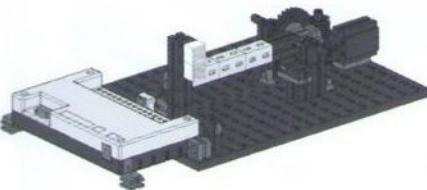


Hoffentlich hat alles geklappt, denn dein Nachbar ist überglücklich, wenn seine Maschine funktioniert, bevor der Programmierer erscheint. Du weißt ja, er braucht sie dringend.

4.3 Parkhausschranke (siehe Bauanleitung Seite 18)

Nächsten Samstag soll in der Stadt das neue Parkhaus eröffnet werden.

Heute wurde die Schranke für die Zufahrt eingebaut. Da inzwischen bekannt ist, dass du der beste Programmierer der Stadt bist, hat man dich gebeten, die Programmierung zu übernehmen. Natürlich bist du stolz darauf und machst dich sofort an die Arbeit. Baue das Modell auf.



Aufgabe 1:

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.

Programmier-Tipps:

- 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 Tür 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 vorne.

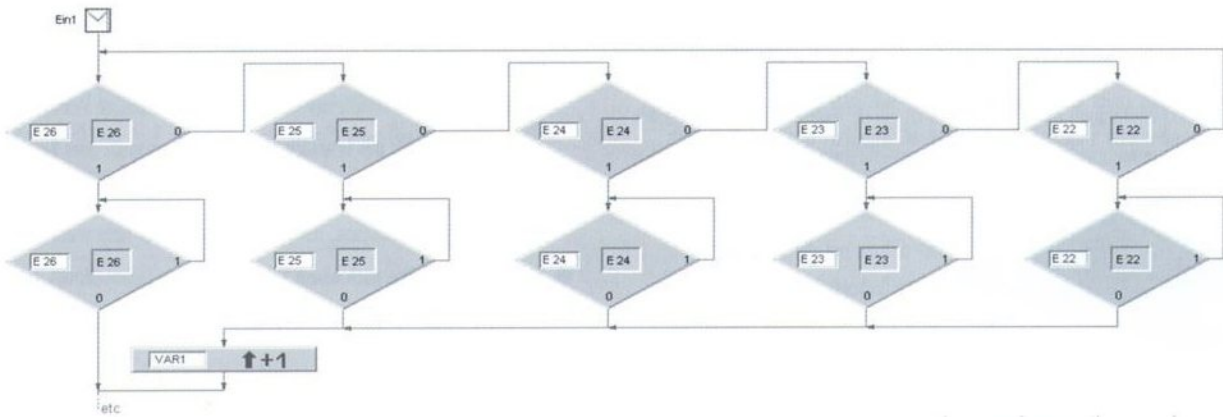
Aufgabe 2:

Das Parkhaus soll am Eröffnungstag für prominente Gäste freigehalten werden. Dazu erhalten die Parkberechtigten eine geheime Zahlenkombination mit 3 Ziffern. Sie müssen drei der 5 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.

Programmier-Tipps:

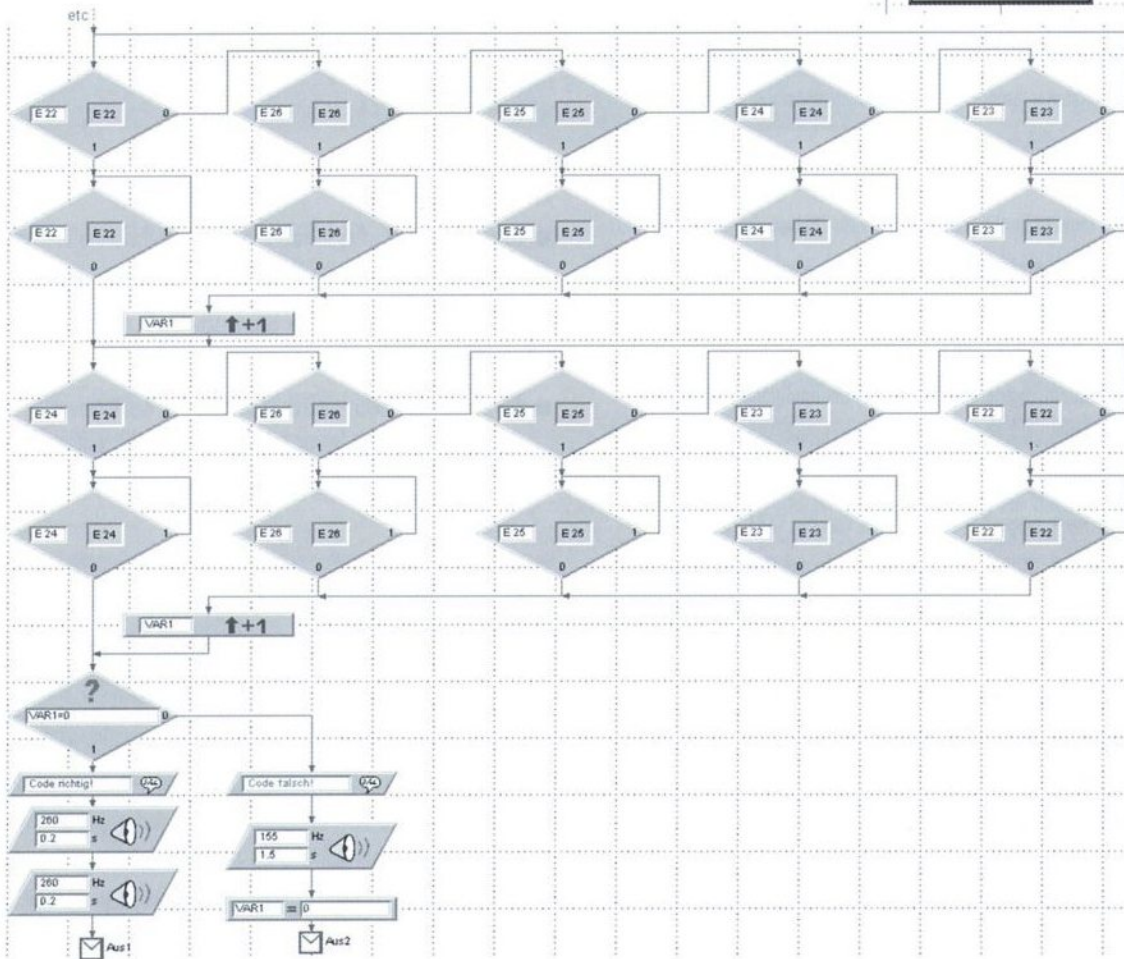
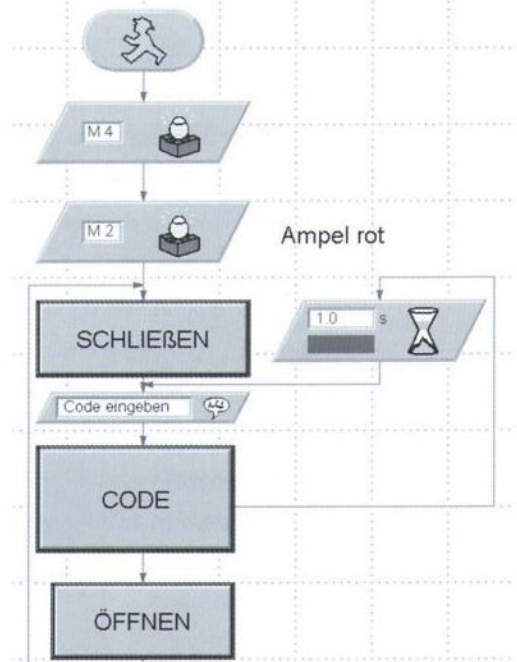
- Das Unterprogramm „Code“ kannst du ganz unterschiedlich aufbauen: Am Einfachsten ist es, mit dem Baustein FLANKE die drei Eingänge E26, E22 und E24 nacheinander abzufragen. Wurden die drei Tasten gedrückt, erscheint die Meldung „Code richtig“ und du verlässt das Unterprogramm. Bei dieser Lösung können 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 wie folgt 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:

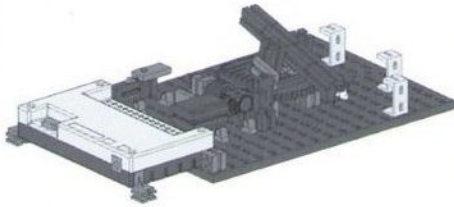
War der Code richtig, öffnest du die Schranke, war er falsch, muss ein neuer Code eingegeben werden. ►

Diese Programmierung ist schon ziemlich anspruchsvoll. Aber versuche es trotzdem. Du wirst sehen, es funktioniert. Und wenn nicht? Keine Sorge, auch hier haben wir wieder das fertige Projekt unter Parkhausschranke2.mdl abgespeichert.



4.4 Schweißroboter (siehe Bauanleitung Seite 21)

Die bereits zuvor erwähnte Werkstatt hat sich jetzt auch noch einen Schweißroboter zugelegt. Da der Besitzer begeistert war, wie du neulich



seine Stanzmaschine programmiert hast, kommt er auch jetzt wieder auf dich zu mit der Bitte, seinen Schweißroboter in Gang zu setzen. Baue zuerst das Modell gemäß der Bauanleitung auf.

Aufgabe 1:

Der Roboter soll an drei verschiedenen Positionen jeweils an einem Metallgehäuse den Deckel mit einem Schweißpunkt fixieren. Die Schweißelektrode wird durch eine Linse simuliert, die drei Metallgehäuse durch gelbe Bausteine. Der Roboter soll die 3 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 den Taster E8 ein Reset ausgelöst werden können, d.h. das Programm nach Drücken und wieder Loslassen des Tasters von vorne starten.

Programmier-Tipps:

- Fahre den Roboter zunächst in seine Ausgangsposition (M1 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 Linse (M2) ein und gleich wieder aus. Diesen Vorgang wiederholst du ein paar Mal, indem du eine Schleife programmierst und eine Variable VAR1 hochzählst, 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 lasse 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.
- Das fertige Beispielprojekt findest du unter Schweißroboter1.mdl

Das Programm für den Schweißroboter ist noch nicht sonderlich groß. Stell dir aber vor, der Roboterarm würde außer im Kreis auch noch nach oben und unten sowie vor und zurück fahren können. Würde man dann jede Position so wie im Beispielprojekt Schweißroboter1 mit einem eigenen Baustein Position anfahren, und müsste man dann nicht nur 3, sondern mehr Positionen anfahren, würde das Programm sehr groß. Werden Programme zu groß, funktionieren sie irgendwann nicht mehr im Download-Modus. Außerdem ist es nicht sehr elegant, dass wir das Unterprogramm „Schweißen“ dreimal aufrufen. Wir bauen das Programm deshalb einmal völlig anders auf, so wie es die richtigen Profis tun:

Aufgabe 2:

Gestalte das Programm so, dass möglichst wenig Bausteine dazu benötigt werden und das Unterprogramm „Schweißen“ nur einmal im Ablauf vorkommt.

Programmier-Tipps:

- Bevor du loslegst, ist es hilfreich, im LLWin 3.0-Handbuch das Kapitel 10.4, „Bausteine sparen – Effizienteres Programmieren in LLWin“, durchzulesen. Dann bist du der Lösung schon sehr nahe. Unser Programm wird sogar etwas einfacher
- Mit der Variable VAR3 steuerst du die Aktionen des Roboters. Nach jeder Aktion wird die Variable um 1 erhöht. Der Roboter soll nacheinander folgendes tun:

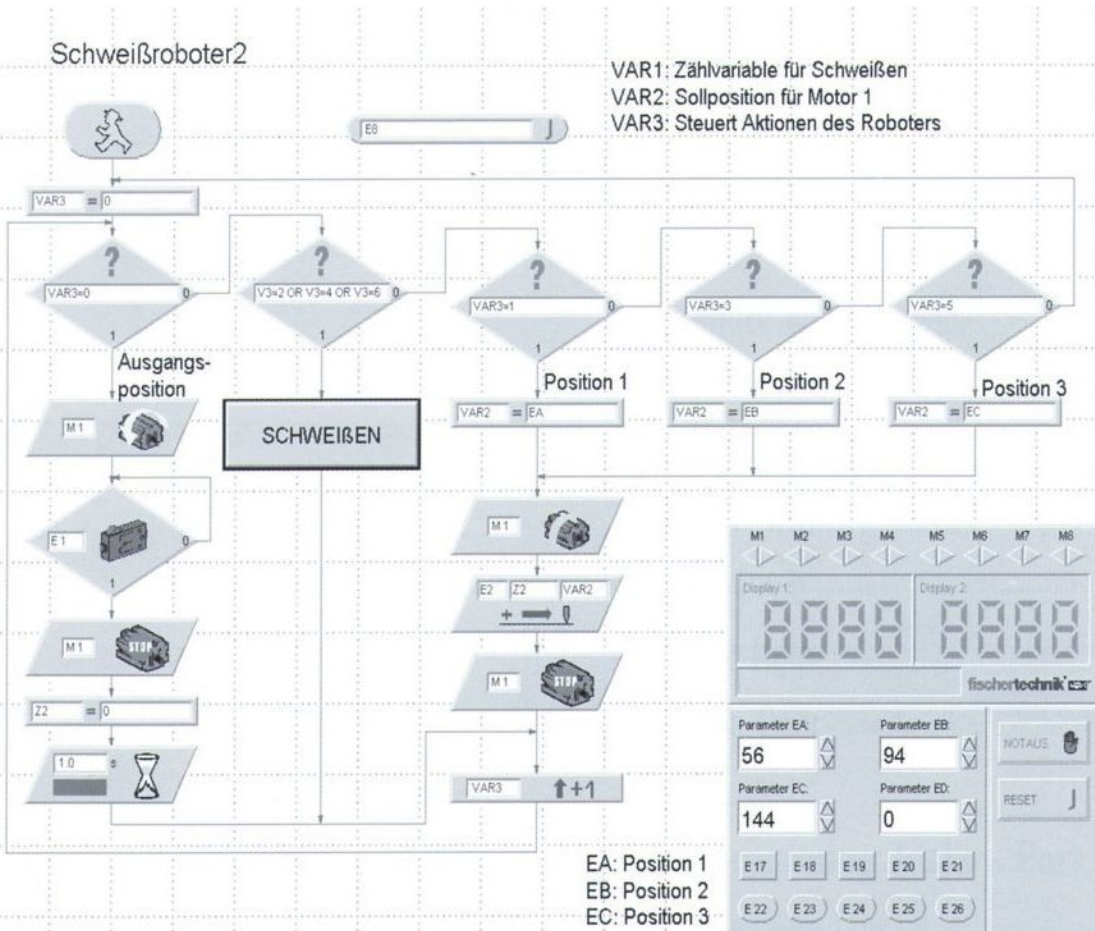
Wert von VAR3	Aktion
0	In Ausgangsposition fahren
1	Zu Position 1 fahren
2	Schweißen
3	Zu Position 2 fahren
4	Schweißen
5	Zu Position 3 fahren
6	Schweißen

- Mit dem Baustein Vergleich fragst du ab, welchen Wert die Variable VAR3 hat. Bei 0 fährt der Roboter in Ausgangsposition, bei 1 fährt er in Position 1, bei 2, 4 und 6 wird geschweißt usw. Der Ablauf sieht dann wie folgt aus:

5. Wie geht es weiter?

Mit etwas Phantasie kannst du dir zu den Modellen des Baukastens Computing Starter sicherlich noch weitere Aufgabenstellungen ausdenken und

die Programme dazu schreiben. So z. B. könnte der Schweißroboter noch an einer vierten Position schweißen, oder die drei vorhandenen Positionen in einer anderen Reihenfolge mehrmals anfahren. Falls du noch weitere Bauteile aus anderen Baukästen besitzt, könntest du z. B. mit zusätzlichen Lampen die Ampel zu einer ganzen Straßenkreuzung mit einer umfangreichen Ampelsteuerung ausbauen. Mit dem Baustein BEEP kannst du auch eine Melodie programmieren, die z.B. ertönt, wenn die Schiebetür geöffnet wird. Lass dir einfach etwas einfallen, Möglichkeiten gibt es noch viele.



- Jede Aktion kommt nur einmal im Programm vor. Sie wird immer dann abgearbeitet, wenn die Variable VAR3 den Wert aufweist, der zu dieser Aktion gehört.
- Der Variablen VAR2 wird die jeweilige Endposition des Motors M1 an den Positionen 1, 2 und 3 zugewiesen. So taucht auch der Ablauf MOTOR RECHTS - POSITION - MOTOR AUS nur einmal auf.
- Das fertige Projekt heißt Schweißroboter2.mdl.

Falls die Ein- und Ausgänge des Intelligent Interface nicht mehr ausreichen, besorgst du dir einfach das Extension Module Art.-Nr. 16554. Diese Erweiterungsbaugruppe wird direkt am Interface angeschlossen. Damit stehen dir weitere 4 Ausgänge und 8 digitale Eingänge zur Verfügung. Im LLWin 3.0-Handbuch sind in den Kapiteln 7 bis 9 alle Bausteine, Menübefehle, die Symolleiste und Tastaturkombinationen beschrieben. Diese Kapitel sind als Nachschlagewerk sehr hilfreich. Im Kapitel 10 sind wertvolle Tipps und Tricks für die Programmierung mit LLWin 3.0 enthalten. Lesen lohnt sich!

Zugegeben, diese sog. zustandsorientierte Programmierung erscheint zunächst komplizierter als das einfache Hinetereinandersetzen der Bausteine wie im Projekt Schweißroboter 1. Vielleicht ist es auch tatsächlich etwas übertrieben, diesen einfachen Roboter so zu programmieren. Andererseits, je mehr der Roboter kann und je mehr Positionen angefahren werden, desto sinnvoller wird sie. Deshalb ist es auf jeden Fall hilfreich, sie zunächst einmal an einem einfachen Modell anzuwenden. Dann hast du später bei den großen „Industry Robots“, bei denen 3 Motoren gleichzeitig eine Position anfahren, damit keine Probleme mehr.

Dann gibt es auch noch weitere Computing Baukästen von fischertechnik. Mobile Robots sind fahrbare Roboter, die man so programmieren kann, dass sie z. B. Hindernissen ausweichen oder nicht vom Tisch fallen. Mit dem Baukasten Pneumatic Robots lassen sich pneumatisch angetriebene Maschinen mit Hilfe von elektromagnetischen Ventilen über das Interface steuern. Aus dem Baukasten Industry Robots lassen sich Greifroboter mit drei Achsen bauen, die im dreidimensionalen Raum Gegenstände greifen, versetzen und stapeln können. Das bedeutet Bau- und Programmierspaß ohne Ende. Und das Beste daran, es kommt ständig etwas Neues dazu.

Computing starter

- Begleitheft
- Activity booklet
- Manuel d'accompagnement
- Cuaderno adjunto
- Folheto

5.0 s



fischerwerke
Artur Fischer
GmbH & Co. KG
Weinhalde 14-18,
D-72178 Waldachtal
Telefon 0 74 43/12-43 69
Telefax 0 74 43/12-45 91
<http://www.fischertechnik.de>
email: info@fischertechnik.de

fischertechnik[®] [®]