

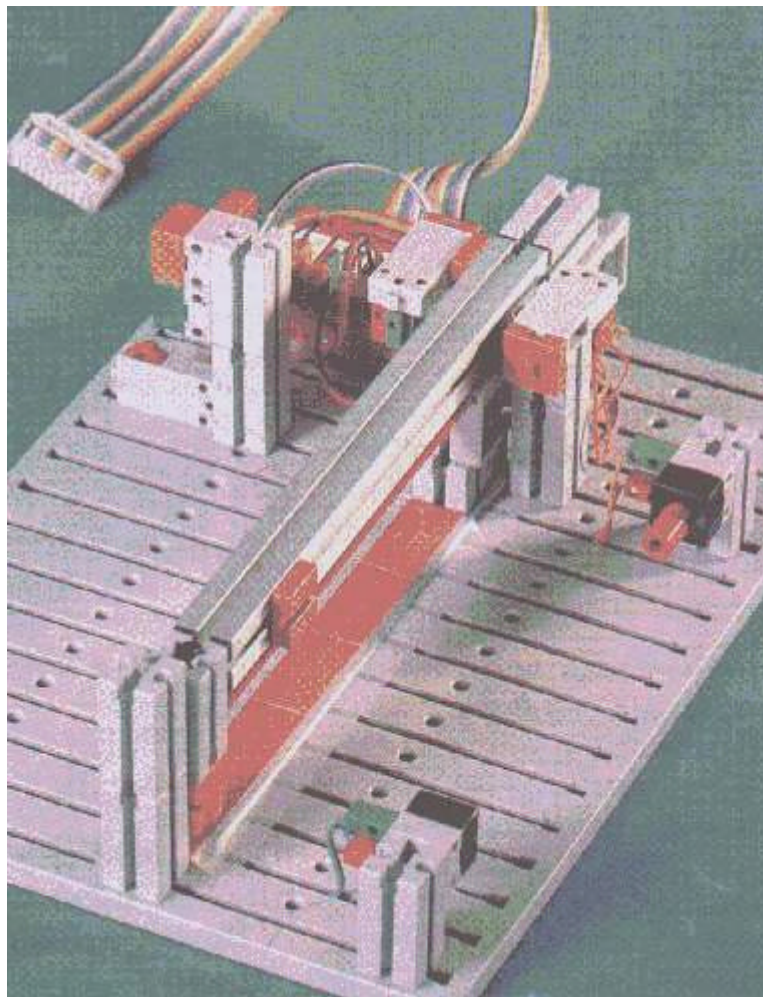
Qbasic-Kurs

"Steuern und Regeln mit dem Computer – Steuerung einer Parkhaus-Schranke"

von Rolf Düren

E-Mail: rolf.dueren@rd.bm.shuttle.de

PDF-Konvertierung: Thomas Antoni 17.4.2004 - www.qbasic.de



Unterrichtseinheit: Steuern und Regeln mit dem Computer**Thema: Nutzung von Interface und Programmiersprache**

Für das Steuern und Regeln mit dem Computer benötigst Du - neben dem Computer und der üblichen Peripherie (Monitor, Maus usw.) -

ein **Interface** und eine **Programmiersprache**.

Ein Interface ist ein Zusatzgerät, durch das der Computer mit Maschinen und Geräten (eine Ampel, eine Leuchtreklame, eine Bohrmaschine usw.) verbunden werden kann. Über das Interface kann er die Maschinen dann ansprechen und steuern.

Wir benutzen im Unterricht das FISCHER - Interface

Von den verschiedenen Anschlußmöglichkeiten dieses Interface sind für uns zunächst nur die Ein- und Ausgänge wichtig.

An die **8 Eingänge** können Sensoren angeschlossen werden und der Computer kann dann z.B. feststellen ob eine Lichtschranke unterbrochen wurde, eine Temperatur überschritten ist oder ein Auto vor einer Ampel steht.

An die **8 Ausgänge** können Aktoren angeschlossen werden und der Computer kann dann z.B. Lampen einschalten, eine Tür öffnen oder ein Fließband in Bewegung setzen. **EX und EY** werden später behandelt.

All das kann - nein - könnte der Computer. Es gibt nur ein Problem:

Der Computer weiß ja zunächst nicht das er es tun soll und noch nicht einmal wann und wie er es tun soll. Irgendjemand muß ihm das erst einmal sagen und beibringen.

Aber auch da gibt es ein Problem: **Der Computer versteht uns nicht**

Zumindest kannst Du ihm nichts so erklären wie Deinem Nachbarn.

Damit der Computer Anweisungen entgegennehmen kann muß er zunächst eine Sprache lernen - **eine Programmiersprache**.

Und:

Der Mensch, der Ihm Anweisungen erteilen will muß diese Sprache - leider - auch erst mal lernen.

Um dem Computer Befehle zu erteilen benutzen wir die Programmiersprache **QBASIC**. Du mußt folgende Schritte durchführen, um sie dem Computer beizubringen:

1. Schalte Computer und Monitor ein.

Warte, bis auf dem Monitor „Windows 95 wird gestartet...erscheint, drücke schnell F8, wähle Nr. 4 (Einzelbestätigung). Jetzt startet Windows im DOS-MODUS, der **Prompt** erscheint. Oder unter WIN95: Startmenü -> Programme -> MS-DOS-Eingabeaufforderung.

2. Lege Deine Arbeitsdiskette in das Laufwerk A: ein.

3. Tippe ein: „A:“ und drücke die <ENTER> - Taste.

Du hast jetzt das Laufwerk A: ,das Diskettenlaufwerk, angewählt. Der Bildschirm zeigt jetzt das Bereitschaftszeichen von Laufwerk A:.

4. Gib ein: „QBASIC“ und drücke die <ENTER> - Taste

5. Warte auf weitere Anweisungen!

*Unterrichtseinheit: **Steuern und Regeln mit dem Computer***
*Thema: **Nutzung von Interface und Programmiersprache***

Nach dem Laden von QBASIC ist das Schreibfeld des Bildschirms leer. Oben steht die Menüleiste mit den Wörtern:

Datei Bearbeiten Ansicht Suchen Ausführen Debug Aufrufe

Für Deine ersten Schritte sind folgende Kommandos wichtig:

[Das Datei Menü]

In diesem Menü stehen alle Kommandos, die irgendwie mit dem Zugriff auf Dateien zu tun haben.

1. Lade die Datei: **Grundpr.bas** Mit dieser Option wird das Programm von der Diskette in den Rechner geladen. Du kannst dieses Programm dann weiterbearbeiten oder später ausführen lassen.
2. Nach dem Öffnen sofort die Datei Grundpr.bas unter einem anderen Namen (Datei -> Speichern unter.. , nicht mehr als 8 Zeichen) **abspeichern**. Dein neues Programm enthält jetzt alle Befehle zur Ansteuerung des Schul-Interfaces.

Das Gedächtnis des Computers

Das Gedächtnis eines Computers besteht aus vielen einzelnen Speicherzellen, den sogenannten Bits. In jeder dieser Speicherzellen kann nur einer von zwei möglichen Werten gespeichert werden: eine 0 oder eine 1. Ein heutiger Computer enthält mehrere Millionen dieser Speicherzellen, die zusammengeschaltet das Gedächtnis des Computers bilden, den Arbeitsspeicher. Den einzelnen Bits können verschiedene Aufgaben zugewiesen sein. Wird der Computer zum Steuern eingesetzt, dann sind acht dieser Bits für die Ausgänge des Interface zuständig. Diese Bits können auf 0 oder 1 gesetzt werden und dadurch Lampen oder Motoren aus- bzw. eingeschaltet werden.

Ebenso sind acht Bits für die Eingänge des Interface zuständig. Der Wert dieser Bits kann durch Schalter, die mit dem Interface verbunden werden, geändert werden. Der Computer „merkt“ so, daß ein Schalter betätigt wurde.

Man kann die Eingänge mit den menschlichen Sinnen wie Auge, Ohr usw. vergleichen. Die Ausgänge übernehmen die Funktion der Gliedmaßen wie Hände, Beine usw.

Vokabeln der Computersprache

In einer Computersprache werden verschiedene Arten von Anweisungen unterschieden:

Befehle

Strukturen

Bedingungen.

Die Befehle:

CALL MOTOR (Nummer, Richtung)

Die Ausgangsbits sind von 1 bis 8 belegt: Motor 1 bis 4; dazu kommen die Richtungen (aus, links, rechts, ein=rechts). Also insgesamt 8 Ausgänge.

Beispiel: CALL MOTOR(1, links). Damit kann man auch acht Lampen ansteuern, die dann alle mit Masse verbunden werden müssen.

SLEEP [Sekunden]

Dieser Befehl weist den Computer an, eine bestimmte Zeit zu warten.

Beispiel: SLEEP 2

Mit diesen beiden Befehlen können wir unsere erstes kleines Programm schreiben. Zuvor muß jedoch das Interface mit dem Computer verbunden (LPT1), das Interface mit Strom versorgt(max. 8 Volt) und ein Motor an die Verteilerbox angeschlossen werden.

*Unterrichtseinheit: **Steuern und Regeln mit dem Computer***
*Thema: **Nutzung von Interface und Programmiersprache***

Vorbereitung des Interface

Für unseren ersten Versuch muß das Interface auf die Schnittstelle des Computers eingestellt werden. 1) Lade das Programm INSTALL.BAS

2) Drücke die Funktionstaste „F5“, das Programm wird ausgeführt. Nach erfolgreicher Installation können wir jetzt unser erstes Programm schreiben.

Programmieraufgabe 1:

Dein Programm soll
den Motor1 für 2 Sekunden links einschalten und anschließend den Motor1 ausschalten.

Programm	Kommentar
CLS	Bildschirm löschen
CALL Motor(1, links)	Motor1 dreht sich links
SLEEP 2	Programm wartet 2 Sekunden
CALL Motor(1, aus)	Motor1 wird ausgeschaltet

Programmieraufgabe 2:

Dein Programm soll
Motor1 für 2 Sekunden nach links einschalten und anschließend den Motor1 für 2 Sekunden nach rechts einschalten, dann Motor1 ausschalten.

Deine Lösung:

_____	_____
_____	_____
_____	_____
_____	_____

Programmieraufgabe 3:

Dein Programm soll
das Programm 2 endlos wiederholen. Dazu kann man in QBASIC den Sprungbefehl **GOTO** verwenden. Mit GOTO wird das Programm in der mit <Zeilennummer> angegebenen Programmzeile fortgesetzt. Ergänze dein Programm am Anfang mit der Zeilennummer 50 und setze an das Ende den Befehl: GOTO 50. Jetzt bist du in der Schleife „gefangen“ und kannst ihr nur noch mit der Tastenkombination „CTRL + BREAK“ entfliehen.

Unterrichtseinheit: Steuern und Regeln mit dem Computer
Thema: Nutzung von Interface und Programmiersprache

Hier noch einmal eine Lösung von Programmieraufgabe 3:

*** Beginn des Benutzerprogramms ***

REM ****Motor links, Motor rechts Endlosschleife****

REM **** Abbrechen mit Ctrl-Break ****

50 :

CALL Motor(1, links)

SLEEP 2

CALL Motor(1, aus)

SLEEP 2

CALL Motor(1, rechts)

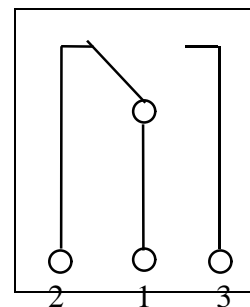
SLEEP 2

CALL Motor(1, aus)

SLEEP 2

GOTO 50

Der Schalter



STRUKTUREN

Wir ergänzen unseren Versuchsaufbau und wollen nun zusätzlich einen Schalter an das Interface anschließen (E1). Anschließend prüfen wir, ob der Schalter funktioniert. Wir laden das Programm „DIAGNOSE.BAS“ und prüfen damit die Funktionstüchtigkeit unseres Modells („F5“ drücken). Nun kannst du sehen, ob der Schalter richtig arbeitet. Verdrahte den Schalter mit seinen Anschlüssen 1 und 2 (s. oben) an die Verteilerbox (E1 und Masse).

Ein wichtiger **Strukturbefehl** in QBASIC lautet:

IF <Bedingung> THEN <Anweisung> ELSE <Anweisung>

Der Computer wird angewiesen einen bestimmten Programmteil nur dann auszuführen, wenn eine Bedingung erfüllt ist.

BEDINGUNGEN

Da wir hier mit einem Taster arbeiten, brauchen wir den Ausdruck, um ihn ansprechen zu können. Die Bedingung fordert einen bestimmten **Wert** (NULL oder EINS, mehr kennt der Computer nicht!). Für eNummer muß die Nummer des Eingangs und für x eine **0 für ausgeschaltet** oder eine **1 für eingeschaltet** eingesetzt werden.

Die Bedingung heißt: **EINGANG%(eNummer) = x** Beispiel: Eingang%(E1) = 0

Schau dir folgendes Programm an und versuche vorherzusagen, was passieren wird, wenn das Programm gestartet wird:

*** Beginn des Benutzerprogramms ***

REM ***Am Schalter wurde 1 und 2 belegt (0=Schalter gedrückt)

50 :

IF Eingang%(E1) = 0 THEN 100

CALL Motor(1, links)

GOTO 50

100 CALL Motor(1, aus)

GOTO 50

Was hast du beobachtet: _____

*Unterrichtseinheit: **Steuern und Regeln mit dem Computer***
*Thema: **Nutzung von Interface und Programmiersprache***

FÜR DIE, DIE MEHR WISSEN WOLLEN (EXPERTENSEITE):

Hier noch ein paar Ergänzungen zu dem Strukturbefehl:

IF <Bedingung> THEN <Anweisung> ELSE <Anweisung>

Du brauchst einen zweiten Taster, schließe ihn an Eingang 2 an.

Programm 5:

```
*****
***  Beginn des Benutzerprogramms      ***
*****
REM ***Der Motor dreht sich solange, bis Taster 1 und Taster 2 gedrückt wird***
REM ***Am Schalter wurde 1 und 2 belegt (0=Schalter gedrückt)
50 :
  IF Eingang%(E1) = 0 AND Eingang%(E2) = 0 THEN 100
  CALL Motor(1, links)
  GOTO 50
100 CALL Motor(1, aus)
GOTO 50
```

Programm 6:

```
*****
***  Beginn des Benutzerprogramms      ***
*****
REM ***Der Motor dreht sich solange, bis Taster 1 oder Taster 2 gedrückt wird***
REM ***Am Schalter wurde 1 und 2 belegt (0=Schalter gedrückt)
50 :
  IF Eingang%(E1) = 0 OR Eingang%(E2) = 0 THEN 100
  CALL Motor(1, links)
  GOTO 50
100 CALL Motor(1, aus)
GOTO 50
```

Zwei Beispiele, die zeigen, daß die IF ... THEN ...ELSE -Struktur mit **logischen Operatoren** (z.B. AND, OR, NOT) verknüpft werden kann.

Programm 5: erst wenn Taster 1 **UND** Taster 2 betätigt werden, wird der Motor ausgeschaltet.

Programm 6: der Motor wird ausgeschaltet, wenn Taster 1 **ODER** Taster 2 betätigt wird.

Ebenfalls ist es möglich, die BEDINGUNG mit verschiedenen **Vergleichsoperatoren** zu belegen.
Hier eine tabellarische Übersicht:

Operator	Bedeutung
=	gleich
>	größer
>=	größer gleich
<	kleiner
<=	kleiner gleich
<>	ungleich



*Unterrichtseinheit: **Steuern und Regeln mit dem Computer***
*Thema: **Nutzung von Interface und Programmiersprache***

Die Analogeingänge EX und EY

Zum Abschluß unseres Einführungskurses wollen wir uns mit den Analogeingängen beschäftigen. Du brauchst dazu aus dem Computingkasten den Fototransistor und die Linsenstecklampe.

Achtung: beim Fototransistor wird die rote Markierung (+) mit EX oder EY auf der Verteilerbox verbunden.

Einlesen eines analogen Wertes

Viele Meßwerte von elektronischen Meßgliedern sind analoge Größen, zumeist Spannungsschwankungen, die sich in Abhängigkeit von der Meßgröße verändern. Mit den Spannungsschwankungen kann aber ein digital arbeitender Computer nichts anfangen. Aus diesem Grunde wurde in unserem Interface noch zusätzlich zwei Analogeingänge eingebaut. Das Interface wandelt die Spannungsschwankungen in digitale Werte um (Einzelheiten interessieren uns hier nicht).

Der Fotowiderstand

In der Elektronik gibt es Widerstände, die ihren Wert bei Beleuchtung ändern. Solche Widerstände heißen Fotowiderstände (**LDR**: Light-Dependent-Resistor = lichtabhängiger Widerstand). Sie bestehen aus Cadmiumsulfid. Je stärker der LDR beleuchtet wird, desto geringer ist sein Widerstand.

In der Automation findet seine Eigenschaft vielfältige Anwendung. So wird z.B. in den automatischen Waschstraßen das sich nähernde Auto in seiner Breite und Höhe „vermessen“ und die Waschbügel entsprechend eingestellt und angepaßt (die Meßwerte werden dem Computer mitgeteilt, der dann die Motoren steuert...“fahre hoch“, „fahre vor“ usw.).

Versuch1:

Baue eine Lichtschranke auf (Linsenlampe plus Fototransistor; Abstand ca. 15cm).

Lade das Programm „DIAGNOSE.BAS“. Schalte damit die Linsenlampe ein. Beobachte die digitalen Werte, wenn du durch die Lichtschranke „fährst“ (abdunkeln).

Beobachtung:

Um einen Widerstandswert in QBASIC einlesen zu können, müssen wir kurz noch den Begriff „**VARIABLE**“ kennenlernen. Man kann die Variable in einem Computerprogramm als Transportbehälter ansehen, die mit veränderbaren Werten gefüllt werden kann. Computertechnisch gesehen, vertritt der Name einer Variable einen bestimmten Speicherplatz, dem „Gedächtnis“ des Computers. Wenn wir eine neue Variable benennen, setzt QBASIC diesen Namen in die Nummer einer Speicheradresse um und reserviert hier einen Speicherbereich, der fortan ausschließlich für diese Variable reserviert ist.

Versuch2:

Schreibe folgendes kleine Programm:

CALL Motor(1, EIN)

FW = EX%

PRINT FW

Kommentar

Linsenlampe einschalten.

Die Speicheradresse FW wird mit dem Wert, den der Fototransistor liefert, besetzt.

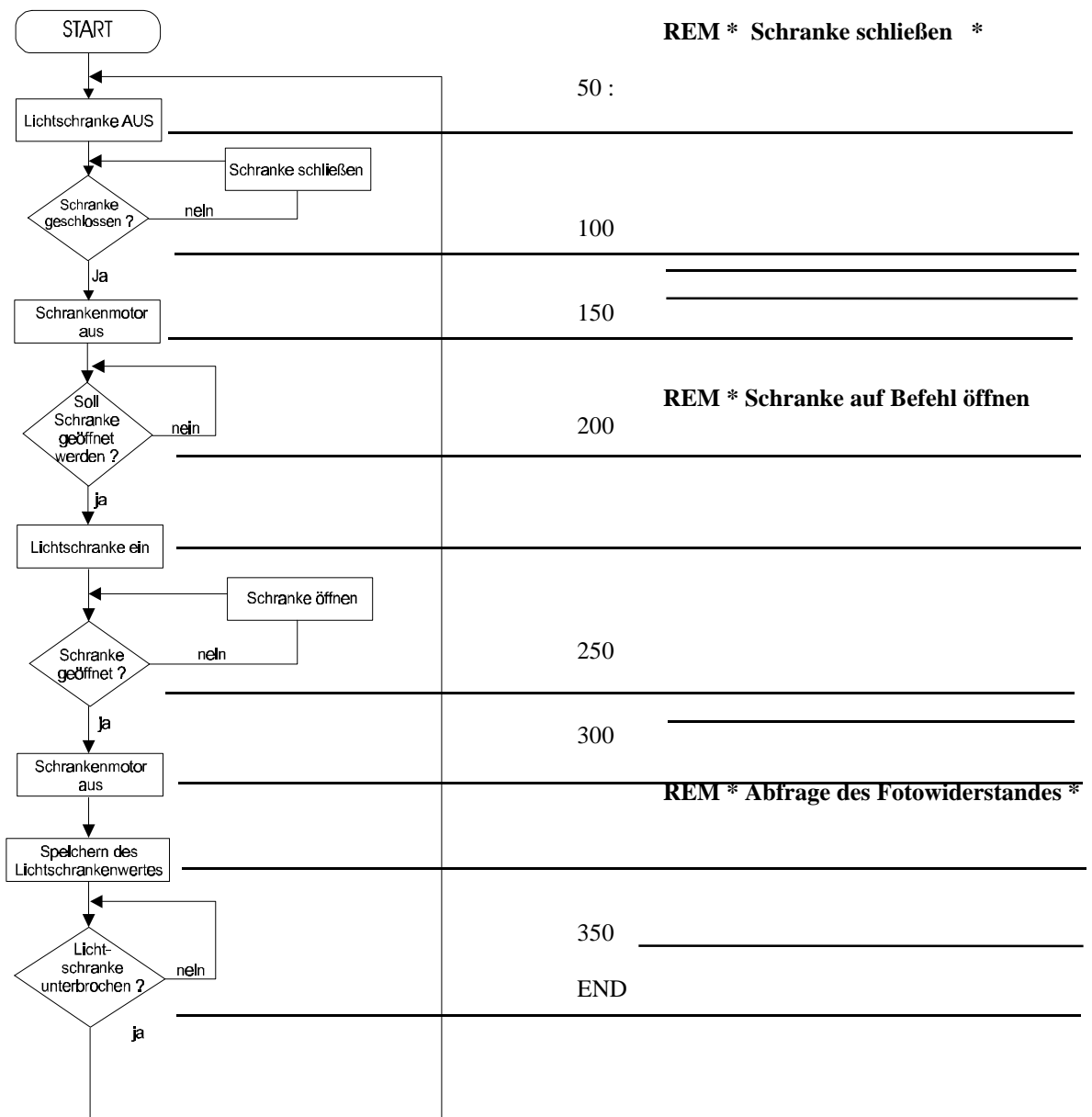
Der digitale Wert von FW wird angezeigt

Zusammenfassung: die Speichervariable kannst du selbst benennen. **Das Gleichheitszeichen** weist den momentanen Wert EX der Variable FW zu. Damit kann sie (FW) zur weiteren Verarbeitung jederzeit abgerufen werden.

Unterrichtseinheit: Steuern und Regeln mit dem Computer
Thema: Nutzung von Interface und Programmiersprache

DEINE AUFGABE :

Du sollst ein Programm entwickeln, daß den Computer anweist, die Schranke nur dann zu öffnen, wenn die Lichtschranke unterbrochen und die Schranke geschlossen wurde. Die einzelnen Ablaufschritte sind hier noch einmal zeichnerisch dargestellt. Schreibe neben den einzelnen Arbeitsbefehlen die entsprechenden Computerbefehle.

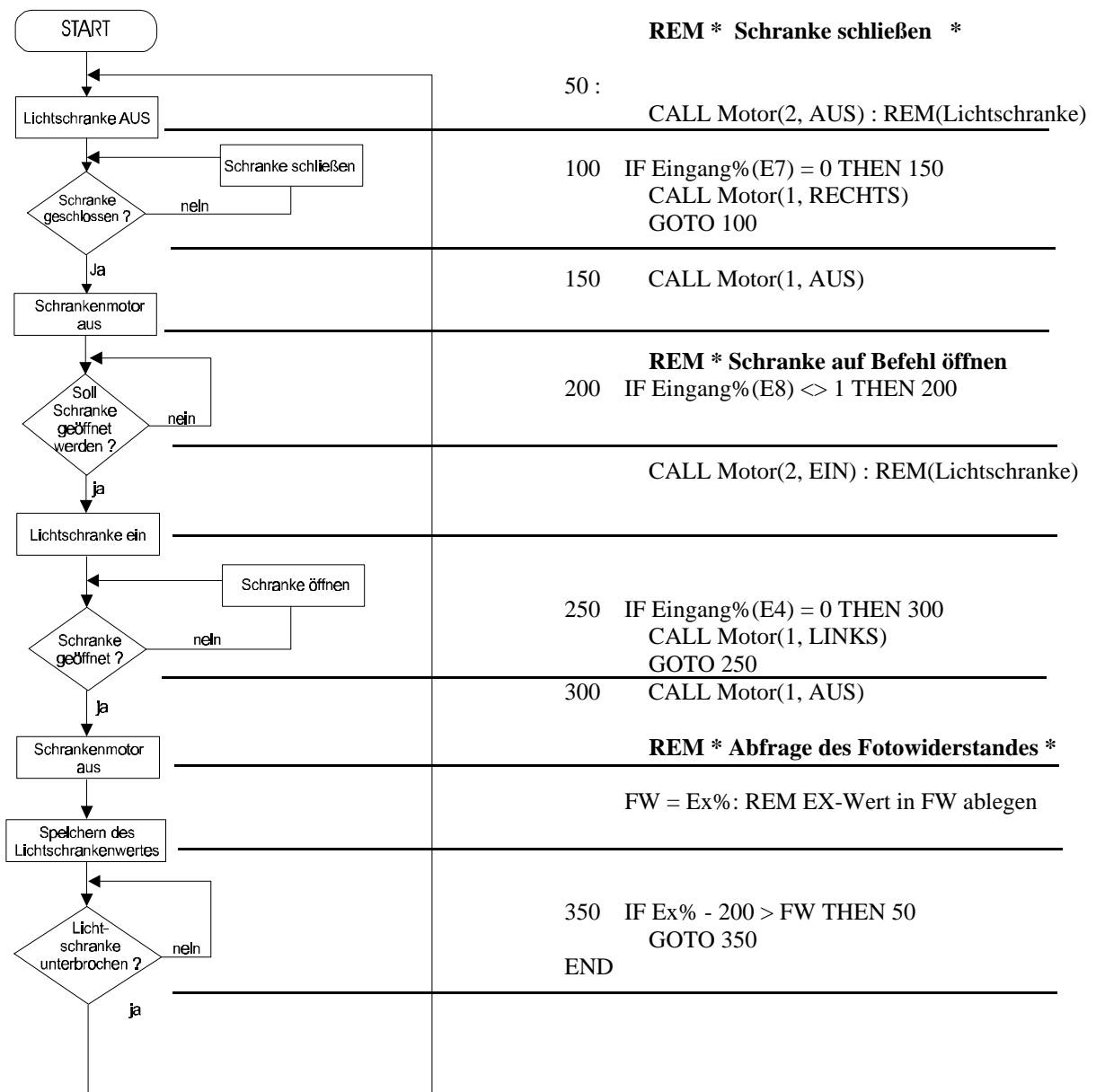
Ablaufdiagramm Zustellschranke


Unterrichtseinheit: Steuern und Regeln mit dem Computer
Thema: Nutzung von Interface und Programmiersprache

DEINE AUFGABE :

Du sollst ein Programm entwickeln, daß den Computer anweist, die Schranke nur dann zu öffnen, wenn die Lichtschranke unterbrochen und die Schranke geschlossen wurde. Die einzelnen Ablaufschritte sind hier noch einmal zeichnerisch dargestellt. Schreibe neben den einzelnen Arbeitsbefehlen die entsprechenden Computerbefehle.

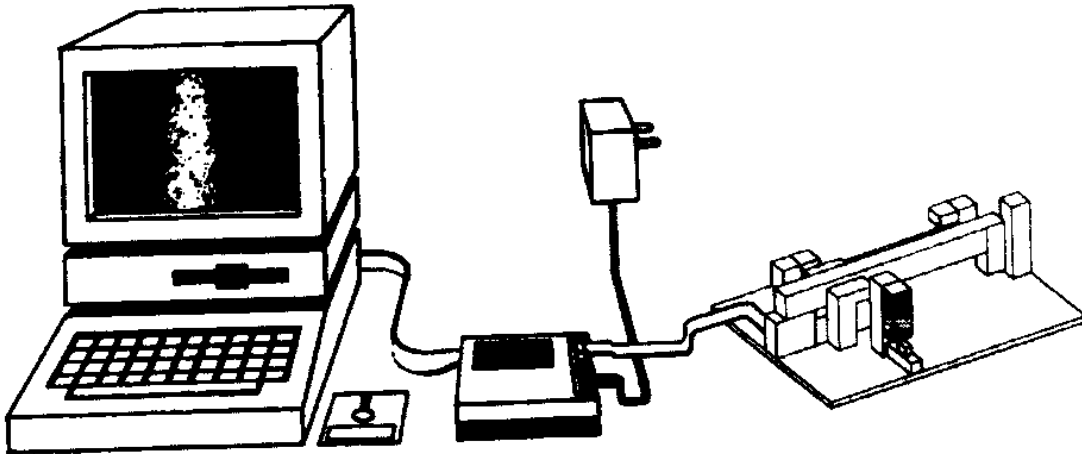
Ablaufdiagramm Zustellschranke



Zu den QBasic-Programmen

Einige Hinweise dazu:

Zunächst muß das Fischertechnik-Interface mit dem Programm „INSTALL.BAS“ auf die parallele Drucker-Schnittstelle des Computers eingestellt werden.



Wenn das Modell (hier Zustellschranke) aufgebaut ist, wird dessen Funktionstüchtigkeit mit dem Programm „DIAGNOSE.BAS“ getestet.

Nach erfolgreichem Test wird das Programm „GRUNDPR.BAS“ geladen (initialisiert den Treiber „TREIBER:BIN“ (muß im gleichen Verzeichnis liegen). Jetzt stehen folgende Befehle zur Ansteuerung des Fischer-Interface zur Verfügung:

Init

Initialisiert das Interface, d.h. alle Motoren werden ausgeschaltet.

Motor (Nummer%, Richtung%)

Befehl zur Ansteuerung der vier Motorausgänge des Interfaces

Nummer% = 1 bis 4

Richtung% = aus, links, rechts, ein(=rechts)

Beispiel:

```
Init
DO
  CALL Motor(1,links)
LOOP UNTIL INKEY$<>" "
CALL Motor(1,aus)
END
```

Eingang% (eNummer%)

Übergibt die Zustände der acht Digitaleingänge als "0" oder "1".

Beispiel:

```
Init
DO
  LOCATE 10,10: PRINT Eingang%(E1)
LOOP UNTIL INKEY$<>" "
END
```

Ex%

Übergibt den Wert des Analogeingangs EX.

Beispiel:

```
Init
DO
  LOCATE 10,10: PRINT USING "####";Ex%
LOOP UNTIL INKEY$<>" "
END
```

Ey%

Übergibt den Wert des Analogeingangs EY.

Beispiel:

```
Init
DO
  LOCATE 10,10: PRINT USING "####";Ey%
LOOP UNTIL INKEY$<>" "
END
```

Wichtig!

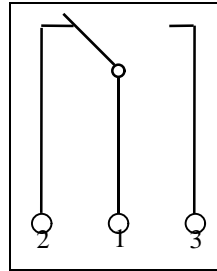
Wenn Sie eigene Programme schreiben möchten, speichern Sie das Programm GRUNDPR.BAS sofort nach dem Öffnen unter einem neuen Namen ab, damit es nicht verändert wird.

```
600 CLS
610 REM *****
620 REM *
630 REM *
640 REM *
650 REM * ZUSTELL-SCHRANKE *
660 REM * (Dü) *
670 REM *
680 REM *
690 REM *****
700 :
```

```

710 REM *****
720 REM * Bildschirmanzeige      *
730 REM *****
740 :
745 CLS
750 LOCATE 3
760 PRINT TAB(11); "*****"
770 PRINT TAB(11); "*               *"
780 PRINT TAB(11); "*   ZUSTELLSCHRANKE   *"
790 PRINT TAB(11); "* (ABBRUCH mit STRG + BREAK) *"
800 PRINT TAB(11); "*****"
810 :
820 REM *****
830 REM * ABFRAGEBELEGUNG      *
840 REM *****
850 :
860 REM E4 = Schranke offen
870 REM E7 = Schranke geschlossen
880 REM E8 = Befehl zum Öffnen
890 REM Ex = Fotowiderstand
900 REM M1 = Motor
910 REM M2 = Lampe  (Lichtschranke)
912 REM M3 = Ampel grün
914 REM M4 = Ampel rot
920 :
930 REM *****
940 REM * Schranke schließen    *
950 REM *****
960 :
970 CALL Motor(2, AUS)
972 CALL Motor(3, AUS)
974 CALL Motor(4, EIN)
980 IF Eingang%(E7) = 0 THEN 1010
990 CALL Motor(1, RECHTS)
1000 GOTO 980
1010 CALL Motor(1, AUS)
1020 :
1030 REM *****
1040 REM * Schranke auf Befehl "öffnen" *
1050 REM *****
1060 :
1070 LOCATE 10, 5
1080 PRINT " TASTE << EINFAHRT >> betätigen "
1090 IF Eingang%(E8) <> 1 THEN 1090
1100 CALL Motor(2, EIN)
1110 IF Eingang%(E4) = 0 THEN 1160
1120 CALL Motor(1, LINKS)
1130 GOTO 1110
1140 :
1150 REM *****
1160 REM * Abfrage des Fotowiderstandes *
1170 REM *****
1180 :
1190 CALL Motor(1, AUS)
1192 CALL Motor(4, AUS)
1194 CALL Motor(3, EIN)
1200 LOCATE 10, 5
1210 PRINT " << LICHTSCHRANKE >> unterbrechen "
1220 FW = Ex%: REM EX-Wert in FW ablegen
1230 IF Ex% - 15 > FW THEN 740
1240 GOTO 1230
1250 END

```



Taster für „Schranke auf , Schranke zu“ sind auf 1 mit +5V und 2 mit E4 und E7 belegt worden.

Taster E8 is mit +5V auf 1 und E8 auf 3 belegt worden

hier muß die Polung am Baustein (Fototransistor!) beachtet werden; rote Markierung am Baustein = EX