

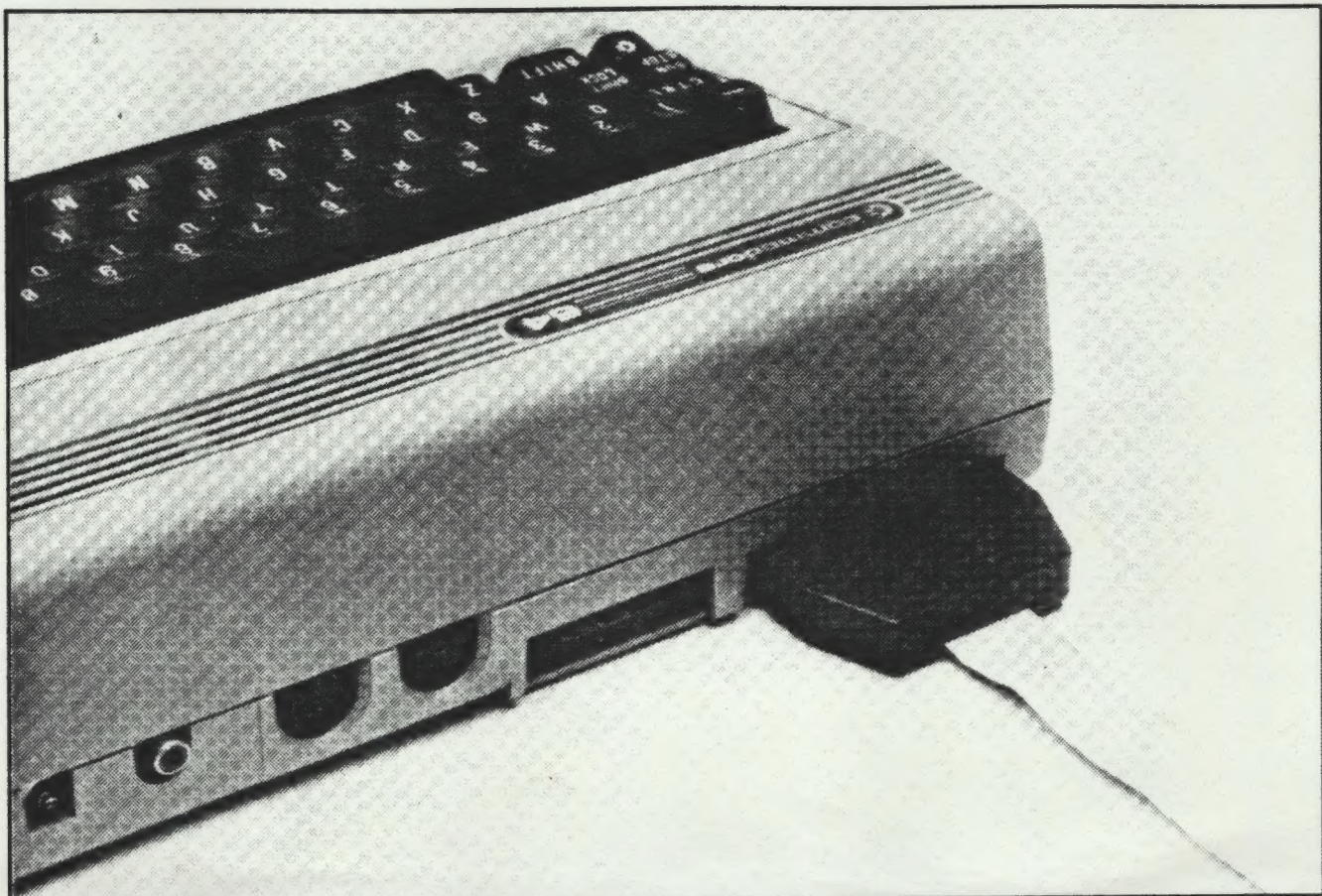
Gebrauchshinweise

Art. 9765

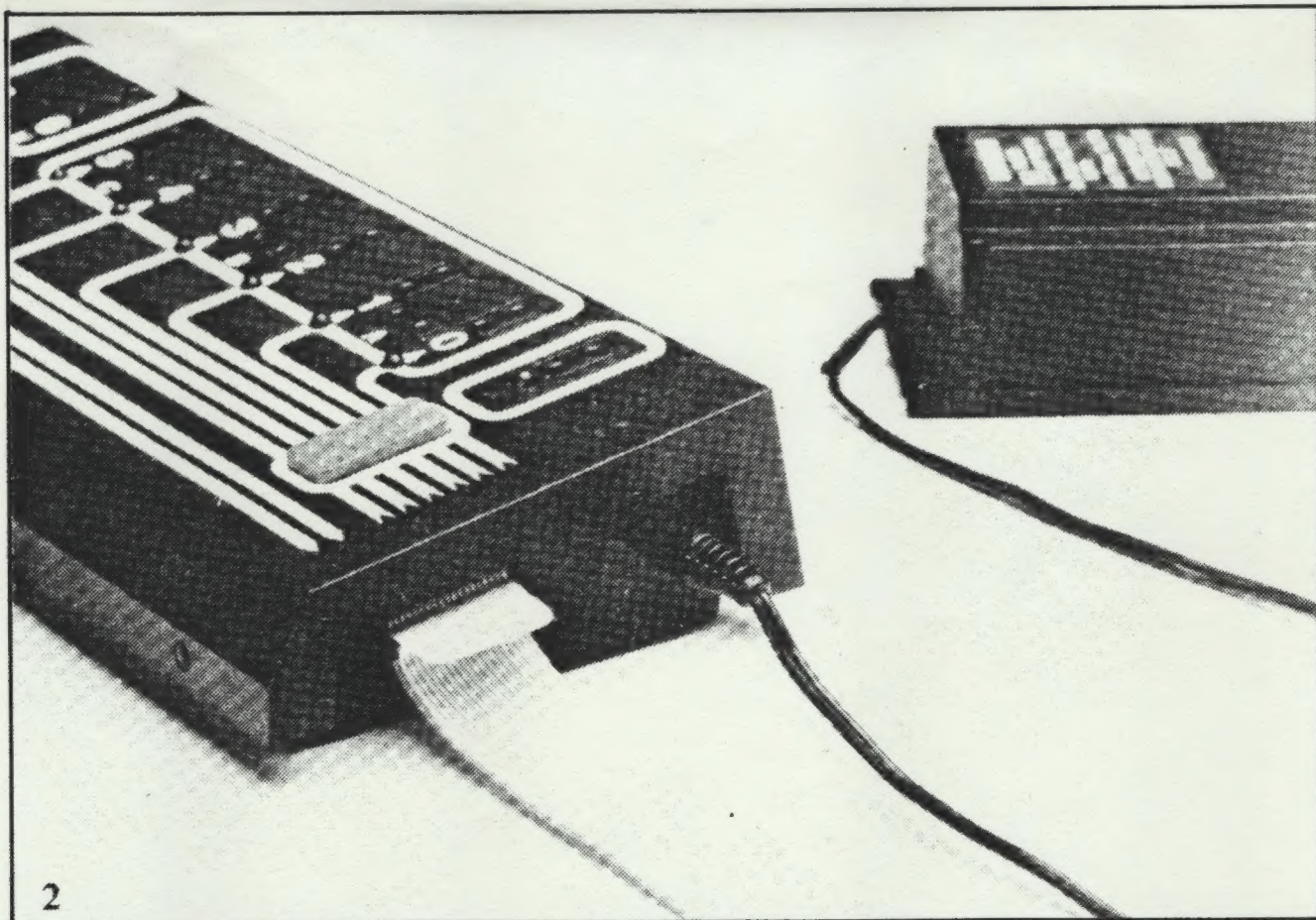
**LEGO® Interfacekabel zu den
Computern Commodore 64 und 128**

Technic





1



2

Dieser LEGO^R Technic Artikel enthält ein LEGO Interfacekabel zu den Computern Commodore 64 und 128 und Gebrauchshinweise.

Gebrauchshinweise

Um eine korrekte Verwendung sicherzustellen, wird folgendes erläutert:

- A Anschließen des LEGO Interface A.
- B Hinweise zum direkten Ansprechen des LEGO Interface A in C64-BASIC und in COMAL 80.
- C Fehlersuche: bei Problemen mit dem Interface und den Modellen kann hier nachgesehen werden.

A Anschließen des LEGO Interface A

Das Interface wird folgendermaßen angeschlossen:

1. Der Computer wird ausgeschaltet, um mögliche Beschädigung zu vermeiden.
2. Das Interfacekabel wird an den user-port des Computers angeschlossen (vgl. Abb. 1).
3. Das Interfacekabel wird an das Interface angeschlossen (vgl. Abb. 2).
4. Das Netzteil wird an das Interface angeschlossen (vgl. Abb. 2).
5. Das Netzteil wird an das Netz angeschlossen.
6. Der Computer wird eingeschaltet.

Der rote Stop-Schalter auf dem Interface soll nicht gedrückt sein!

**B Hinweise zum direkten Ansprechen des LEGO Interface A
in C64-BASIC und in COMAL 80**

Vorgehensschritte:

1. Das Herstellen der Datenverbindung vom Computer zum LEGO Interface A
2. Die Programmierung der Ausgänge 0-5
3. Das Abfragen der Eingänge 6 und 7

**1. Das Herstellen der Datenverbindung vom Computer zum
LEGO Interface A:**

Zunächst müssen die Programmzeilen des Hilfsprogramms, s. Anhang A, in den Computer eingegeben werden.

Für den weiteren Gebrauch ist es sinnvoll, das Hilfsprogramm auf einer Diskette oder einer Kassette zu sichern. Ein Ausdruck mit einem Drucker erleichtert die Suche nach eventuellen Eingabefehlern.

Das Hilfsprogramm besteht aus den folgenden fünf Unterprogrammen:

- init
- biton
- bitoff
- getbit
- wait

init:

Das Unterprogramm init stellt die Datenverbindung zum Interface her. Hier sind keine Parameter erforderlich. Dieses Unterprogramm muß in der ersten Zeile aller Programme zur Steuerung von Modellen aufgerufen werden.

biton:

Das Unterprogramm biton schaltet einen Ausgang auf dem Interface ein. Die Nummer des Ausgangs wird als Parameter übergeben. Im BASIC Programm heißt die Variable zur Übergabe des Parameters NUM%. Der Parameter muß zwischen 0 und 5 liegen.

bitoff:

Das Unterprogramm bitoff schaltet einen Ausgang auf dem Interface aus.
Für die Nummer des Ausgangs gilt das gleiche wie bei biton.

getbit:

Das Unterprogramm getbit liest den Zustand eines Eingangs ab. Die Nummer des Eingangs wird als Parameter übergeben. Sie muß 6 oder 7 sein. Im BASIC Programm heißt die Variable zur Übergabe dieses Parameters NUM%. Der Zustand des Eingangs wird durch die Variable Y% wiedergegeben.

wait:

Das Unterprogramm wait wartet einige Sekunden. Die Zahl - die Wartezeit in Sekunden - wird als Parameter übergeben. Im BASIC Programm heißt die Variable zur Übergabe des Parameters TIM%. Der Parameter muß positiv und kleiner als 32768 sein.

Für ein einwandfreies Funktionieren der C64-BASIC Programme muß sorgfältig zwischen den Programmvariablen und den Variablen zur Interfacesteuerung wie NUM%, Y% und TIM% unterschieden werden.

Programmbeispiele zum Herstellen der Datenverbindung vom Computer zum LEGO Interface A:

Die Datenverbindung zum LEGO Interface A wird durch das Eingeben des Programmbeispiels in C64-BASIC bzw. in COMAL 80 vorbereitet.

Beispiel C64-BASIC:	Kommentar/Erklärung
10 GOSUB 10000: REM init	Herstellung der Datenverbindung
20 END	
Beispiel COMAL 80:	Kommentar/Erklärung
0010 init	Herstellung der Datenverbindung
0020 END	

Mit der Tastenfolge "RUN" und der Taste <RETURN> werden die Programmbeispiele gestartet. Die Kontrolllampen an den Ausgängen 0-5 erlöschen. Jetzt ist der Computer für die Datenübertragung zum und vom Interface startbereit. Bei etwaigen Fehlern muß das eingegebene Programmbeispiel überprüft und eventuell die eingegebenen Unterprogramme mit dem Anhang A verglichen werden.

2. Die Programmierung der Ausgänge 0-5

Programmbeispiele in C64-BASIC bzw. in COMAL 80:

Beispiel C64-BASIC:	Kommentar/Erklärung
10 GOSUB 10000: REM INIT	Herstellung der Datenverbindung
20 NUM%=3	Schaltet den Ausgang 3 ein
30 GOSUB 11000: REM BITON	
40 TIM%=4	Wartet 4 Sekunden
50 GOSUB 14000: REM WAIT	
60 NUM%=4	Schaltet den Ausgang 4 ein
70 GOSUB 11000: REM BITON	
80 TIM%=2	Wartet 2 Sekunden
90 GOSUB 14000: REM WAIT	
100 NUM%=3	Schaltet den Ausgang 3 aus
110 GOSUB 12000: REM BITOFF	
120 TIM%=2	Wartet 2 Sekunden
130 GOSUB 14000: REM WAIT	
140 NUM%=4	Schaltet den Ausgang 4 aus
150 GOSUB 12000: REM BITOFF	
160 END	

Beispiel COMAL 80:	Kommentar/Erklärung
0010 init	Herstellung der Datenverbindung
0020 biton(3)	Schaltet den Ausgang 3 ein
0030 wait(4)	Wartet 4 Sekunden
0040 biton(4)	Schaltet den Ausgang 4 ein
0050 wait(2)	Wartet 2 Sekunden
0060 bitoff(3)	Schaltet den Ausgang 3 aus
0070 wait(2)	Wartet 2 Sekunden
0080 bitoff(4)	Schaltet den Ausgang 4 aus
0090 END	

Bei etwaigen Fehlern muß das eingegebene Programmbeispiel überprüft und eventuell die eingegebenen Unterprogramme mit dem Anhang A verglichen werden.

Wichtiger Hinweis: Die Unterprogramme "biton"/"bitoff" können Ausgänge einzeln ein-/ausschalten, ohne den Zustand der anderen Ausgänge zu beeinflussen.

Beispiel einer Anweisungsfolge:

Einen Ausgang einschalten, warten, einen anderen Ausgang einschalten, warten, den ersten Ausgang ausschalten, warten und den zweiten Ausgang ausschalten.

3. Das Abfragen der Eingänge 6 und 7

Mit den Eingängen 6 und 7 können Signale vom LEGO Optosensor aufgenommen werden.

Der Optosensor wird an Eingang 6 angeschlossen. Wenn ein gelber LEGO Stein dicht vor der ovalen Öffnung des Optosensors hin und herbewegt wird, blinkt die grüne Kontrollampe am Eingang gleichzeitig mit der Bewegung. Jedesmal, wenn die grüne Kontrollampe aufleuchtet, zeigt der Computer bei einem geeigneten Programm den Wert 1 an, wenn die grüne Lampe erlischt, wird der Wert 0 angezeigt.

Ein Beispielprogramm in C64-BASIC bzw. COMAL 80, das 100 mal den Zustand des Eingangs 6 anzeigt:

Beispiel C64-BASIC:	Kommentar/Erklärung
10 COSUB 10000: REM INIT	Herstellung der Datenverbindung
20 NUM%=6	Nummer des Eingangs
30 NO%=100	Zahl der Durchläufe
40 FOR X=1 TO NO%	
50 COSUB 13000: REM GETBIT	Ermittelt den Zustand des Eingangs 6
60 PRINT Y%	Zeigt den Zustand an
70 NEXT X	
80 END	

Beispiel COMAL 80:	Kommentar/Erklärung
0010 init	Herstellung der Datenverbindung
0020 Zahl:=100	Zahl der Durchläufe
0030 FOR X:=1 TO Zahl DO	
0040 PRINT getbit(6)	Ermittelt und zeigt den Zustand
0050 ENDFOR X	des Eingangs 6 an
0060 END	

Bei etwaigen Fehlern muß das eingegebene Programmbeispiel überprüft und eventuell die eingegebenen Unterprogramme mit dem Anhang A verglichen werden.

Während des Programmablaufs muß der Sensor 100 mal z. B. in der oben beschriebenen Weise aktiviert werden.

Ein Beispiel zu einem LEGO Modell:

Vorlage: Ein Modell mit einem Motor, Optosensor und einer Zählscheibe, z. B. ein Riesenrad. Der Motor wird an den Ausgang 1, und der Optosensor an den Eingang 6 angeschlossen.

Die Grobeinteilung der Zählscheibe soll zur ovalen Öffnung des Sensors hin gerichtet sein.

Bei dem folgenden Beispiel wird der Optosensor zum Zählen verwendet. Das Programm zählt 24 Wechsel der Hell-Dunkel-Segmente der Zählscheibe, also 24 Wechsel des Zustandes am Eingang 6.

In den folgenden Programmbeispielen in C64-BASIC bzw. COMAL 80 kann das BASIC Programm nur bis zu 3 Wechsel pro Sekunde zählen. Falls schneller gezählt werden soll, muß ein Unterprogramm benutzt werden, das im Maschinenkode geschrieben ist.

Beispiel C64-BASIC:

Kommentar/Erklärung

10	GOSUB 10000: REM INIT	Herstellung der Datenverbindung
20	NO%=24	Anzahl der Wechsel
30	NUM%=6	Ermittelt und speichert den Start-
40	GOSUB 13000: REM GETBIT	zustand des Eingangs 6
50	QST%=Y%	
60	NUM%=1	Schaltet den Ausgang 1 ein
70	GOSUB 11000: REM BITON	
80	NUM%=6	Nummer des Eingangs
90	FOR QI=1 TO NO%	
100	GOSUB 13000: REM GETBIT	Wartet, bis der Eingang 6 umgeschaltet
110	IF Y%=QST% THEN GOTO 100	wird
120	QST%=Y%	Zwischenspeicherung des Zustandes
130	NEXT QI	
140	NUM%=1	Schaltet den Ausgang 1 aus
150	GOSUB 12000: REM BITOFF	
160	END	

Beispiel COMAL 80

Kommentar/Erklärung

0010	init	Herstellung der Datenverbindung
0020	biton(1)	Schaltet den Ausgang 1 ein
0030	no:=24	Anzahl der Wechsel
0040	FOR qi:=1 TO no DO	
0050	qst:=getbit(6)	Zwischenspeicherung des Zustandes
0060	REPEAT	Wartet, bis Eingang 6 umschaltet
0070	UNTIL getbit(6) <> qst	
0080	ENDFOR qi	
0090	bitoff(1)	Schaltet den Ausgang 1 aus
0100	END	

Bei etwaigen Fehlern muß das eingegebene Programmbeispiel überprüft und eventuell die eingegebenen Unterprogramme mit dem Anhang A verglichen werden.

Das BASIC Programm kann etwas beschleunigt werden, indem die Parameterkontrolle übersprungen wird. Die Zeilennummer bei den GOSUB-Anweisungen müssen dann jeweils folgendermaßen geändert werden:

BITON:	GOSUB 11020
BITOFF:	GOSUB 12020
GETBIT:	GOSUB 13020
WAIT:	GOSUB 14020

C Fehlersuche

Anleitung zur schnellen Fehlersuche

Bei Problemen mit dem Interface und den Modellen wird die folgende kurze Checkliste zum Erkennen von Fehlern in den meisten Fällen ausreichen:

1. Der rote Stop-Schalter auf dem Interface soll nicht gedrückt sein.
2. Das Interfacekabel muß korrekt am Interface und am Computer angebracht sein, vgl. "Anschließen des Interface".
3. Der Stecker des Netzteils muß korrekt am Interface angebracht sein.
4. Das Netzteil muß am Netz angeschlossen sein (die rote Kontrollampe über dem Stop-Schalter muß leuchten).
5. Funktionieren die Ausgänge 0-5? (Überprüfung mit einem LEGO Leuchtstein)
6. Ist das Modell korrekt angeschlossen?
7. Sind alle Anschlußkabel in Ordnung?

Erweiterte Anleitung zur Fehlersuche:

Nachfolgend eine ausführliche Checkliste. Sie umfaßt folgende Teile:

- 1** **LEGO Interface A und Netzanschluß**
- 1.1** **Test der Spannungsversorgung vom Netzteil**
- 1.2** **Test des Permanentausganges des Interface**
- 1.3** **Test der Eingänge des Interface**
- 2** **LEGO Interface A, Datenübertragung**
- 2.1** **Start**
- 2.2** **Schalten der Ausgänge**
- 2.3** **Abfragen der Eingänge**

CHECKVERFAHREN:

1.0 Untersuchung auf visuelle Defekte

1.1 Test der Spannungsversorgung vom Netzteil

ACHTUNG: Bei diesem Test soll der Permanentausgang des Interface (über dem Stop-Schalter) nicht belegt sein.

AKTION	TEST	WENN JA	WENN NEIN
1. Anschluß des Netzteils an das Interface A. 2. Anschluß des Netzteils an das Netz.	Leuchtet die rote Kontrolllampe über dem Stop-Schalter	Das Netzteil ist in Ordnung.	Ist das Netzteil mit dem Netz verbunden? Wenn möglich: Test des Netzteils mit einem anderen Interface. Test des Interface mit einem anderen Netzteil.

1.2 Test des Permanentausganges des Interface

AKTION	TEST	WENN JA	WENN NEIN
Ein LEGO Leuchtstein wird an den Permanentausgang angeschlossen.	Leuchtet der Leuchtstein?	Der Permanentausgang ist in Ordnung.	Test des Permanentausganges mit einem anderen Leuchtstein. Test mit einem anderen LEGO Anschlußkabel.

2.1 Start

AKTION	TEST	WENN JA	WENN NEIN
<p>1. Der Computer wird gemäß den Anleitungen des Herstellers aufgestellt und angeschlossen.</p> <p>2. Der Computer muß abgeschaltet sein.</p> <p>3. Das LEGO Interface A wird an den Computer angeschlossen. Das Netzteil wird an das Interface angeschlossen.</p> <p>Der Stop-Schalter auf dem Interface darf nicht gedrückt sein.</p> <p>4. Das Netzteil wird an das Netz angeschlossen.</p> <p>5. Der Computer wird eingeschaltet.</p>	<p>Leuchten sämtliche roten Kontrolllampen an den Ausgängen?</p>	<p>Die Kontrolllampen an den Ausgängen sind in Ordnung.</p>	<p>Ist das Netzteil mit dem Netz verbunden?</p> <p>Test, ob der Stop-Schalter am Interface gedrückt ist. Er soll nicht gedrückt sein.</p> <p>Test, ob alle Anschlüsse korrekt sind.</p> <p>Wenn vorhanden:</p> <p>Test mit einem anderen Interface.</p> <p>Test mit einem anderen Interfacekabel.</p> <p>Test mit einem anderen Computer.</p>
<p>6. Eingabe und Ausführung der Anweisungen: POKE 56579,63 POKE 56577,0</p>	<p>Werden alle Ausgänge ausgeschaltet?</p>	<p>Weiter: Abschnitt 2.2 "Schalten der Ausgänge".</p>	<p>Wenn vorhanden:</p> <p>Test mit einem anderen Interfacekabel.</p> <p>Test mit einem anderen Interface.</p> <p>Test mit einem anderen Computer.</p>

2.2 Schalten der Ausgänge

AKTION	TEST	WENN JA	WENN NEIN
<p>1. Startverfahren wie unter Abschnitt 2.1 "Start" beschrieben.</p> <p>2. Eingabe und Ausführung der Anweisung: POKE 56577,41</p> <p>Eingabe und Ausführung der Anweisung: POKE 56577,22</p> <p>3. Die Ausgänge A, B, C werden jeder für sich mit einem LEGO Motor getestet.</p> <p>4. Eingabe und Ausführung der Anweisung: POKE 56577,42</p> <p>5. Eingabe und Ausführung der Anweisung: POKE 56577,21</p> <p>6. Eingabe und Ausführung der Anweisung: POKE 56577,0 Der Motor wird ausgeschaltet.</p>	<p>Leuchten die Kontrolllampen an den Ausgängen 0, 3 und 5 auf?</p> <p>Leuchten die Kontrolllampen an den Ausgängen 1, 2 und 4 auf?</p> <p>Startet der Motor, wenn er mit den Ausgängen A, B und C verbunden wird?</p> <p>Ändert der Motor die Drehrichtung?</p>	<p>Datenübertragung zu den Ausgängen.</p> <p>Datenübertragung zu den Ausgängen.</p> <p>Weiter: Aktion 5.</p> <p>Die Ausgänge sind in Ordnung.</p>	<p>Wenn vorhanden: Test mit einem anderen Interfacekabel.</p> <p>Wenn vorhanden: Test mit einem anderen Interface. Test mit einem anderen Computer.</p> <p>Test mit einem anderen LEGO Anschlußkabel. Test mit einem anderen LEGO Motor. Ausgang defekt.</p> <p>Ist ein Anschlußstecker in seiner Buchse am Motor oder am Interface gegenüber der Aktion 4 "umgedreht" eingesteckt? Die Ausgänge sind defekt.</p>

2.3 Abfrage der Eingänge

AKTION	TEST	WENN JA	WENN NEIN
<p>1. Startverfahren wie unter Abschnitt 2.1 "Start" beschrieben.</p> <p>2. Test der Eingänge wie unter Abschnitt 1.3 "Test der Eingänge des Interface" beschrieben.</p> <p>3. Ein LEGO Leuchtstein wird an den Eingang 6 und ein LEGO Leuchtstein an den Eingang 7 angeschlossen.</p> <p>Eingabe und Ausführung der Anweisung: PRINT PEEK(56577) AND 192</p>			
<p>4. Die Leuchtsteine werden von den Eingängen 6 und 7 entfernt.</p> <p>Eingabe und Ausführung der Anweisung: PRINT PEEK (56577) AND 192</p>	<p>Erscheint die Zahl 192 auf dem Bildschirm?</p> <p>Erscheint die Zahl 0 auf dem Bildschirm?</p>	<p>Weiter: Aktion 4.</p> <p>Die Anschlußkabel und die Eingänge sind in Ordnung.</p>	<p>Test mit anderen LEGO Anschlußkabeln.</p> <p>Wenn vorhanden: Test mit einem anderen Interfacekabel. Test mit einem anderen Interface. Test mit einem anderen Computer.</p> <p>Wenn vorhanden: Test mit einem anderen Interfacekabel. Test mit einem anderen Interface. Test mit einem anderen Computer.</p>

BILAG A: COMAL 80

```
8000
8010 // end of user program space
8020 END
8030
8040 PROC init
8050     portb:=56577
8060     POKE 56579,63
8070     POKE portb,0
8080 ENDPROC init
8090
8100 PROC biton(num)
8110     IF num<0 OR num>5 THEN errhand(1)
8120     POKE portb,2↑num BITOR PEEK(portb)
8130 ENDPROC biton
8140
8150 PROC bitoff(num)
8160     IF num<0 OR num>5 THEN errhand(1)
8170     POKE portb,PEEK(portb) BITAND (255-2↑num)
8180 ENDPROC bitoff
8190
8200 FUNC getbit(num)
8210     IF num<>6 AND num<>7 THEN errhand(2)
8220     RETURN (PEEK(portb) BITAND 2↑num)=2↑num
8230 ENDFUNC getbit
8240
8250 PROC wait(tim)
8260     IF tim<0 THEN errhand(3)
8270     qt:=TIME+tim*60
8280     REPEAT
8290     UNTIL TIME>qt
8300 ENDPROC wait
8310
8320 PROC errhand(erc)
8330     IF erc=1 THEN PRINT "outputbit must be between 0 and 5"
8340     IF erc=2 THEN PRINT "inputbits must be 6 or 7"
8350     IF erc=3 THEN PRINT "waittime must be positive"
8360     END
8370 ENDPROC errhand
```




BILAG A, CBM 64 BASIC

```
1 REM INIT                LINE 10000
2 REM BITON               LINE 11000
3 REM BITOFF              LINE 12000
4 REM GETBIT              LINE 13000
5 REM WAIT                LINE 14000
6 REM ERRORHANDLING       LINE 20000
9940 END
9950 REM
9960 REM ----
9970 REM INIT
9980 REM ----
9990 REM
10000 P=56577
10010 POKE 56579.63
10020 POKE P.0
10030 RETURN
10950 REM
10960 REM -----
10970 REM BITON          PAR:NUM%
10980 REM -----
10990 REM
11000 IF NUM%>=0 AND NUM%<6 THEN GOTO 11020
11010 ERC-1:GOTO 20000
11020 POKE P.(2*NUM% OR PEEK(P))
11030 RETURN
11950 REM
11960 REM -----
11970 REM BITOFF        PAR:NUM%
11980 REM -----
11990 REM
12000 IF NUM%>=0 AND NUM%<6 THEN GOTO 12020
12010 ERC-1:GOTO 20000
12020 POKE P,PEEK(P) AND (255-2*NUM%)
12030 RETURN
12950 REM
12960 REM -----
12970 REM GETBIT        PAR:NUM%
12980 REM -----
12990 REM
13000 IF NUM%=6 OR NUM%=7 THEN GOTO 13020
13010 ERC-2:GOTO 20000
13020 Y%=ABS((PEEK(P) AND 2*NUM%)-2*NUM%)
13030 RETURN
13950 REM
13960 REM -----
13970 REM WAIT          PAR:TIM%
13980 REM -----
13990 REM
14000 IF TIM%>=0 THEN GOTO 14020
14010 ERC-3:GOTO 20000
14020 QT=TIME+(TIM%*60)
14030 IF QT>TIME THEN GOTO 14030
14040 RETURN
19950 REM
19960 REM -----
19970 REM ERRORHANDLING
19980 REM -----
19990 REM
20000 PRINT:PRINT"PARAMETER ERROR"
20010 IF ERC=1 THEN PRINT "OUTPUTBITS MUST BE BETWEEN 0 AND 5"
20020 IF ERC=2 THEN PRINT "INPUTBITS MUST BE 6 OR 7"
20030 IF ERC=3 THEN PRINT "WAITTIME MUST BE POSITIVE"
20040 END
```