

Universal interface:

Robox710

Dette interface, kaldet ELLKIT ROBOX 710, er beregnet til at sætte mellem en computer, - i første omgang Commodore 64/128, og en opstilling man gerne vil have styret.

Den er specielt velegnet til brug sammen med LEGO-Technic, men kan iøvrigt anvendes til meget andet, f.eks. sammen med Fischer-teknik.



Diagrammet

Interfacet har 2 indgange for tilslutning af en sensor, kontakt eller lignende, og 6 udgange som hver kan tænde og slukke en motor, lampe etc. To udgange kan anvendes til retningsstyring af en motor så denne kan køre to veje.

Ind- og udgangstrin er adskilt fra hverandre, bortset fra en delvis fælles spændingsforsyning.

Diagrammet til interfacet kan derfor inddeles i tre dele:

To indgangstrin.

Seks udgangstrin.

Spændingsforsyning
som leverer 9V og 4V til henholdsvis ind- og udgangstrin.

Indgangstrinnet

De to indgangstrin er helt identiske og denne gennemgang vil derfor kun dreje sig om det ene (nr.6), hvis diagram er vist i figur 1. Diagrammet til indgang 7 er præcis det samme bortset fra andre komponentbetegnelser, - se iøvrigt komponentlisten.

De to indgange har numrene 6 og 7, fordi de er forbundet til bit 6 og 7 i computerens port (i C64/128 er det userporten), men herom senere.

Indgangstrinnet er bygget

op omkring operationsforstærkeren IC1 (LM358), som her dog anvendes som comparator. IC1 er den eneste komponent som anvendes i både indgang 6 og 7, idet LM 358 (altså IC1) indeholder to operationsforstærkere i samme hus. Bennumrene i parentes i fig. 1 henviser til de anvendte ben i indgang 7.

Den ikke-inverterende indgang på IC1 er forbundet til en spændingsdeler, R3/R5, der danner en spænding på den halvforsynings-spænding (9V). P1 sørger for en positiv tilbagekobling, således at IC1 reagerer som en schmitt-trigger med en hysteresis, der kan justeres på P1.

Den inverterende indgang på IC1 er gennem C1 forbundet til sensoren, som kan være en fotomodstand, fototransistor eller blot en kontakt (en original LEGO-sensor kan også anvendes). Når en fotomodstand belyses falder dens modstand, og spændingen på den inverterende indgang af IC1 vil derfor stige.

Når spændingen her overstiger spændingen på den ikke-inverterende indgang, vil udgangen på IC1 gå mod 0 volt. Dette vil få T1 til at åbne, hvorved lysdioden i

optokobleren (IC2) vil belyse fototransistoren (i IC2), som derved ligeledes vil åbne. Hele resultatet af dette bliver, at udgang D6 (som går til bit 6 i computeren) vil gå HØJ når fotomodstanden belyses. Det samme vil ske, hvis man anvender en kontakt eller foto-transistor, mens en original LEGO-sensor vil fungere lige modsat (den modstand **stiger** ved belysning (udgangen fra interfacet går LAV). Optokobleren IC2 sørger for en galvanisk adskillelse mellem computer og interface/detektor. Herved kan man aldrig komme til at ødelægge computeren, selvom noget elektronik i interfacet skulle bryde sammen.

Udgangstrinnet

De seks udgangstrin er ligesom indgangstrinnene helt identiske, og kun et enkelt udgangstrin (nummer 0) vil derfor blive beskrevet. Diagrammet vist i figur 2, gælder således for alle udgange, bortset fra komponentbetegnelserne, - se iøvrigt komponentlisten.

Udgangene er nummererede fra 0 til 5, da de har forbindelse til henholdsvis bit 0,1,2,3,4 og 5 i computerens port.

Når udgangen fra computer-

porten (her bit 0) går HØJ, vil den trække basis på T8 mod 5V, og denne vil derfor åbne, under forudsætning af, at der er spænding på punktet A. Når T8 åbner, belyser lysdioden i optokobleren IC4 transistoren, som derved åbner.

Herved trækkes basis på T7 mod 0, som derved åbner og tvinger T5 til at åbne. Er der tilsluttet en motor til udgangen (i diagrammet betegnet "OUT0"), vil denne køre. Dette indikeres også ved at lysdioden D3 lyser op. Dioderne D4 og D5 skal 'uskadeliggøre' eventuelle induktionsstrømme fra DC-motorer.

Når udgangen fra computeren går LAV lukker T8, hvorved transistoren i optokobleren (IC4) også lukker. I stedet for T5/T7 åbner nu transistorparret T4/T6, idet modstanden R25 trækker basis på T4 mod plus (4V). T6 vil nu tvinge udgangen mod nul, og hvis en motor er sluttet til "OUT0", vil den ikke køre. Dette indikeres ved at lysdioden D3 er slukket.

Udgangene på interfacet kan anvendes på to måder: Enten enkelvis til start og stop af f.eks. en motor, eller parvis til retningsstyring af en motor. I sidst nævnte

tilfælde forbindes motoren mellem to udgange, f.eks. til henholdsvis udgang 0 (OUT0) og udgang 1 (OUT1). Sættes udgang 0 (OUT0) til ON (logisk 1 på portbit 0 $\Leftrightarrow$ lysdioden lyser), mens udgang 1 (OUT1) sættes til OFF (logisk 0 på portbit 1 $\Leftrightarrow$ lysdiode slukket), vil motoren køre den ene vej. Byttes nu rundt på udgangens niveau (OUT0 = 0, OUT1 = 1), vil motoren køre den anden vej, mens motoren kan standses ved at sætte logisk 0 (LAV) på begge port-bits.

Spændingen fra udgangstrinnene ligger på cirka 4 V, og der kan fra hvert trin trækkes en strøm på op til 250 mA, dog ialt max. 1 A fra alle 6 trin på en gang.

Spændingsforsyningen

Vekselspændingen fra transformatoren ensrettes af dioderne D21-D24 og udglattes af C4. IC10 (LM317) er variabel spændingsregulator, hvis udgangsspænding er bestemt af forholdet mellem R85 og P3. I dette tilfælde skal P3 justeres så IC10 leverer en spænding på cirka 4 volt til udgangstrinnene.

IC10 kan levere en strøm på op til 1,5 ampere, hvilket er rigeligt til at forsyne f.eks. 6 små elektromotorer sluttet til udgangene. Man skal dog være opmærksom på at IC10 kan blive noget varm med den her valgte køleplade hvis interfacet belastes fuldt ud. D29 er en ekstra sikkerhed mod induktionsstrømme fra eventuelle tilsluttede DC-motorer, mens C5 skal stabilisere de 4 volt. Lysdioden D28 er placeret på kassens front og lyser (gult) når der er strøm fra transformatoren.

Dioderne D25/D26 og kondensatorerne C3/C6 fungerer som spændingsdoblere og ensretter, således at de 6 volt fra transformatoren bliver til cirka 12 volt DC. Dette er nødvendigt da spændingsregulatoren IC11 kræ-

ver en spænding på mindst 11 volt for at fungere. IC11, som er en 78L08, skal levere spænding til indgangstrinnene, hvilket klares ved at hæve dens udgangsspænding (som egentlig er på 8 volt) til knap 9 volt via spændingsfaldet over dioden D27. Kondensatoren C7 skal stabilisere spændingen. Indgangstrinnet bruger omkring 20 mA, mens regulatoren IC11 kan levere op til 100 mA.

Sammenhæng

I figur 4 er vist hvordan ind- og udgangstrin, samt spændingsforsyningen er forbundet med hinanden. Kasserne "OUT" er udgangstrin mens "IN" er indgangstrin. Bemærk at de 5V/0 fra computeren ingen forbindelse må have med selve spændingsforsyningen.

Transistoren T3 styrer strømmen til punkterne A på udgangstrinnene. Hvis omskifteren S1 er OFF, vil R34 sørge for at åbne transistoren således, at der kan løbe strøm til punktet A, som derved bliver lagt på ca. 5 volt. Nu kan computeren styre udgangstrinnene ON og OFF som tidligere beskrevet. Når omskifteren S1 er ON, vil T3 være lukket, og der kan ikke løbe strøm til punktet A, og udgangstrinnene vil derfor alle være OFF (slukket), uanset hvilke styreniveauer computeren udsender. På denne måde kan alle udgange slukkes på en gang, hvis noget går i skoven og hurtigt skal standses.

Byggevejledning

Start med at lodde de 18 nitter fast på forpladen. De skal sidde med kraven udvendig på forpladen. Afstanden mellem nitterne er afpasset så LEGO's dobbelt stik kan anvendes, og hvis man ønsker at benytte sig heraf, skal man huske at kontrollere afstanden mellem nitterne både vandret og lodret (de tre udgange parvist).

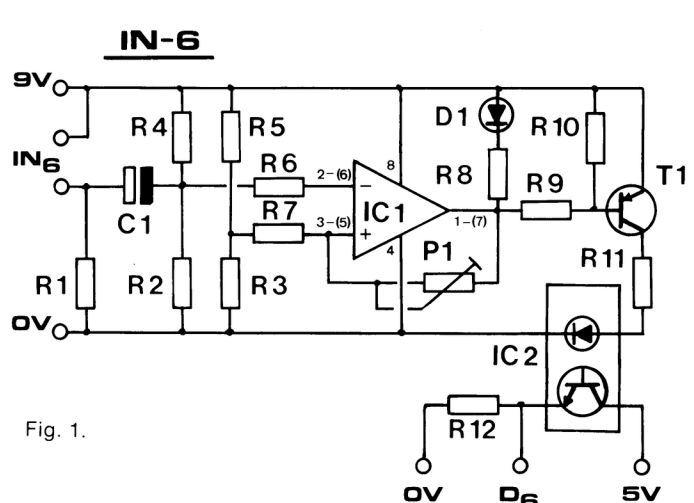


Fig. 1.

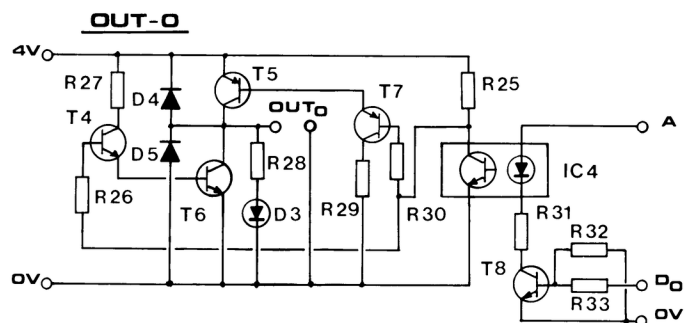


Fig. 2.

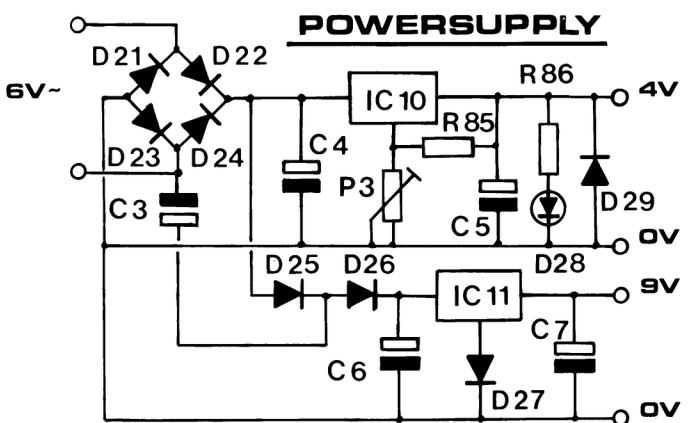


Fig. 3.

Monter derefter komponenter på printet, **-men vent med lysdioderne!**

Begynd med de 10 lus. Klip eventuelt et stykke af benene på modstandene og brug det. Derefter monteres modstande, trimmepotentiometre, kondensatorer, dioder, transistorer og IC'er/op-tokoblere. Husk at vende de fire sidstnævnte rigtigt! Printspydene (4 stk.) skal placeres i forbindelserne til S1 og AC. Spændingsregu-

latoren IC10 skal spændes fast (liggende) på køleplade/printplade med skrue og møtrik før dens ben loddes! Inden det 20 benede hanstik loddes i printet skal de 3 styreriller på undersiden af stikket fjernes med en fil før, at det passer præcist med forpladen.

Lysdioderne skal nu loddes fast på **loddessiden** af printet! Start med at klippe et stykke af benene, så de er ca. 5 mm lange. Pas på: Når

benene er klippet lige lange, kan man ikke se hvad der er anode og katode, - det lange ben er anoden og skal vende mod + på printet. Skub dioderne på plads i hakkerne i printet (de skal helt ind til printet) og lod benene fast - MEN kontroller hele tiden at dioderne passer i hullerne i forpladen.

DC-bøsningen limes på forpladen og trykomskifteren skrues fast i forpladen, men vent med at forbinde dem til printet, før forplade og printet er loddet sammen. Printet skal loddet fast på forpladen, så de danner en ret vinkel mod hverandre. Vær opmærksom på, at forpladen skal passe i kassens styreriller og at printet samtidig skal passe med skruehullerne i bunden af kassen. Dette kræver både tålmodighed og omhyggelighed! Til sidst loddet monteringsledninger fra DC-bøsningen og trykomskifteren til henholdsvis loddesspydene 'S1' og 'AC'.

Printet skrues fast i bunden af kassen med fire selvskærende skruer, men vent med at sætte låget på kassen, da trimmepotentiometrene skal justeres.

Kabelforbindelse

Til kabelforbindelsen mellem interface og Commodore 64/128 skal bruges et 20 leder fladkabel, et 20 pollet fladkabel stik (female header) samt et stik til Commodorens user-port. Det 20-polede hunstik skal ikke loddet, men klemmes på fladkablet, hvilket nemmest gøres i en skruestik. På stikket er en lille pil ved ben 1, og den yderste leder i fladkablet er forbundet hertil. Ben 2 i stikket er benet overfor ben 1 (ikke ved siden af!) - og lederen ved siden af den yderste leder, går derfor til ben 2. Næste leder går til ben 3 osv. De 20 ledere skal forbindes som følger til user-portstikket:

Det skal bemærkes at benforbindelserne til stikket i interfaceet direkte kan aflæses på forpladen af kassen! Interfaceet kan selvfølgelig styres fra enhver anden computer end C64/128 med en 8 bit port. Der skal blot være 5 volt DC til rådighed, enten fra computeren eller en ekstern strømforsyning.

Sensorer m.v.

De 18 nitter/bøsninger på forpladen passer præcis i størrelse til almindelige 2 mm bananbøsninger, som kan købes hos enhver løsdelshandel. Hvis man anvender LEGO's dobbelt stik, så passer de også i forpladen.

Udgangene kan som nævnt anvendes på to måder: Enten til on/off styring af motor/lampe, eller til retningsstyring af motor. I førstnævnte tilfælde, skal motoren/lampen forbindes til de to bøsninger/nitter hørende til en given udgang, - altså to bøsninger **lodret**. Ved retningsstyring af motor, skal stikkene sættes imellem to udgange **vandret**, hvilket vil sige de to øverste bøsninger ved f.eks. udgang 0/1, 2/3 eller 4/5.

Til indgangene kan, som tidligere nævnt, forbindes en

Funktion:	Ben nr. interface/leder nr.:	Userport, ben nr.:
+ 5 volt	1	2
Ikke anvendt	2	
+ 5 volt	3	2
Ikke anvendt	4	
0 volt	5	N
Udgang 0 (bit 0)	6	C
0 volt	7	N
Udgang 1 (bit 1)	8	D
0 volt	9	N
Udgang 2 (bit 2)	10	E
0 volt	11	N
Udgang 3 (bit 3)	12	F
0 volt	13	N
Udgang 4 (bit 4)	14	H
0 volt	15	N
Udgang 5 (bit 5)	16	J
0 volt	17	N
Indgang 6 (bit 6)	18	K
0 volt	19	N
Indgang 7 (bit 7)	20	L

række forskellige sensorer, f.eks. fotomodstande, fototransistorer, kontakter eller LEGO's lys-censorklods. Vil man gerne styre LEGO-opstillinger, men finder LEGO's lysklods for dyr (den koster ca. 150 kr.) kan man lave sin egen på følgende vis:

I en 2-knop (ca. 7,5x15x10 mm) LEGO-klods med midterhul, placeres en fototransistor, f.eks. BPX95B. Hullet i LEGO-klodsens er 5 mm

i diameter, og da BPX95B er i et standard 5 mm lysdiodehus, passer den lige præcis i hullet. Benene på transistoren kan klippes af og der kan loddet ledninger på benene.

Efter samme princip kan man lave en lysgiver som passer til LEGO, idet man istedet for fototransistoren anvender en 5 mm lysdiode. Man skal blot huske at sætte en 220 ohms modstand i serie med dioden, så strøm-

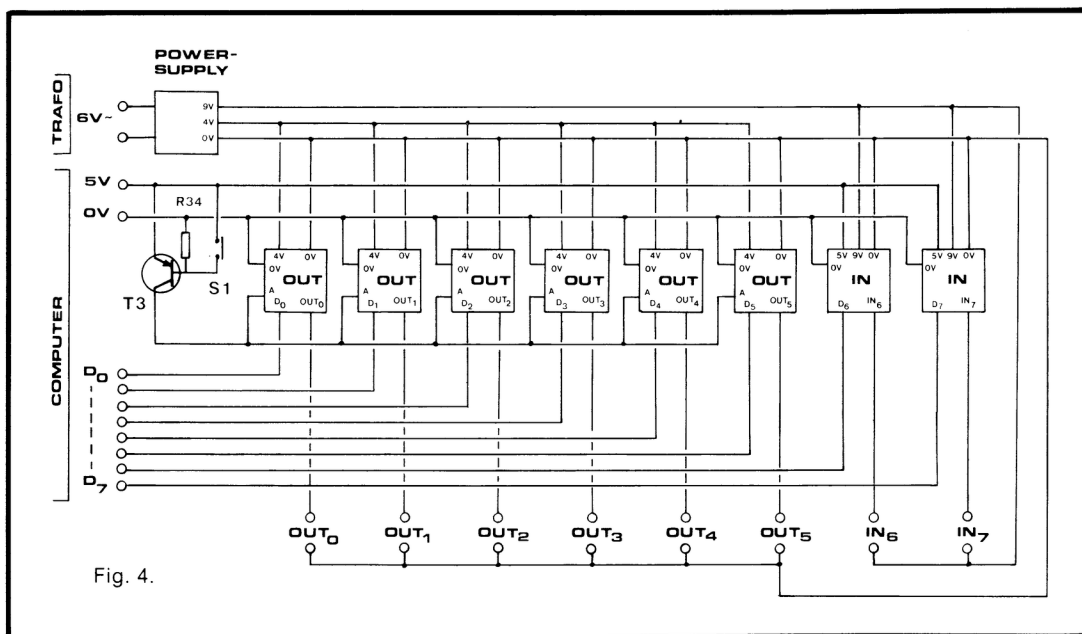


Fig. 4.

men gennem den begræn-
ses. Lysgiveren kan så en-
ten styres fra en udgang,
eller sluttes til bøsninger
ved den gule lysdiode, som
der hele tiden er ca. 4 volt på.
Efter denne beskrivelse af
interfacet, følger en artikel
af Børge Christensen, op-
finderen af programmer-
ings sproget COMAL, som
giver eksempler på hvordan
interfacet anvendes til sty-
ring af LEGO-modeller med
'hjemmelavede' censorer
og lysgivere.

Afprøvning og justering

Først skal potentiometret
P3 justeres, hvilket kræver
adgang til et voltmeter.
Start med at dreje P3 helt i
bund, hvilket vil sige mod
R85. Forbind voltmeteret o-
ver C5, sæt strøm på kon-
struktionen ved at slutte 6
volt AC (fra transformator)
til DC-stikket, og juster på
P3 indtil voltmeteret viser
4,0 volt.

Når dette er i orden kan man
begynde selve afprøvningen:
Forbind interfacet til com-
puteren og indtast følgen-
de lille testprogram i BASIC:

```
0010 POKE 56579,63
0020 POKE 56577,63
0030 INPUT A$
0040 POKE 56577,0
0050 PRINT CHR$(147)
0060 PRINT PEEK(56577)
0070 GOTO 50
```

Linie 10:

Portene initialiseres så bit
6 og 7 er indgange, og bit 5
til 0 er udgange.

Linie 20:

Udgangene tændes (de rø-
de lysdioder 0 til 5 lyser).
hvis en motor eller lampe er
tilsluttet en udgang skal
den køre/lyse.

Linie 30:

Maskinen afventer at der
trykkes RETURN.

Linie 40:

Udgangene slukkes (de rø-
de lysdioder 0 til 5 slukker).
Hvis en motor eller lampe
er tilsluttet skal den stoppe/
ikke lyse.

Linie 50:

Skærmen slettes.

Linie 60:

Der skrives et tal i øverste
venstre hjørne af skærmen.

Linie 70:

Programmet vender tilbage
til linie 50.

Linie 80:

Skal forårsage at følgende
skrives på skærmen:

192 - hvis lysdiode for ind-
gang 6 og 7 lyser.

128 - hvis lysdiode for ind-
gang 7 lyser.

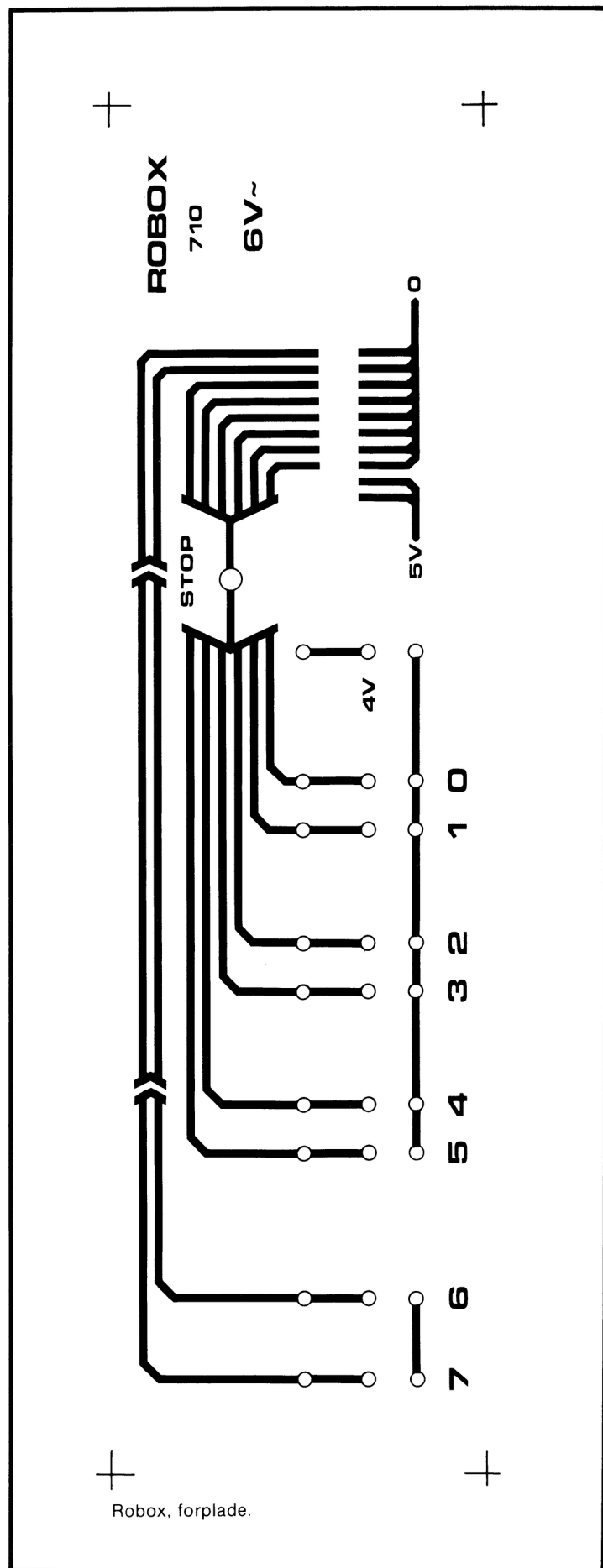
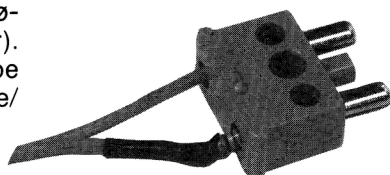
64 - hvis lysdiode for ind-
gang 6 lyser.

0 - hvis ingen af de grøn-
ne lysdioder lyser (for
indgang 6 og 7).

Prøv at kortslutte den ene
indgang med en prøveled-
ning. Når det sker, skal lys-
dioden for pågældende ind-
gang tænde, samtidig med
at tallet på skærmen skif-
ter værdi. Det samme gøres
med den anden indgang.
Hvis dette er i orden, og lys-
dioderne for udgangene
tænder som beskrevet ved
kørsel af programmet, skul-
le der ikke være fejl i elek-
tronikken.

Til sidst skal indgangene
justeres så de passer til cen-
sorerne. Forbind først den
valgte censor til indgang 6,
og juster på P1 indtil ind-
gangen (den grønne lysdio-
de) reagerer når censoren
udsættes for lys og mørke.

Indgang 7 justeres på sam-
me måde ved at bruge P2.
Hvis man bruger forskelli-
ge sensorer, skal man være
opmærksom på, at det kan
være nødvendigt at foreta-
ge en justering hver gang
sensoren skiftes.



Komponenter mv.

Indgangstrinnene:

Indg. 6	Indg. 7	Værdi:
R1	R13	390 ohm
R2	R14	22 kohm
R3	R15	4,7 kohm
R4	R16	22 kohm
R5	R17	4,7 kohm
R6	R18	2,2 kohm
R7	R19	2,2 kohm
R8	R20	680 ohm
R9	R21	10 kohm
R10	R22	1,5 kohm
R11	R23	680 ohm
R12	R24	820 ohm
P1	P2	220 kohm lille vandret trimpot.
IC1	IC1	LM358
IC2	IC3	PC817 optokob.
C1	C2	10 μ F/16V e.lyt
D1	D2	LED 3 mm grøn
T1	T2	BC557

Udgangstrinnene:

Udgang 0 Udgang 1 Udgang 2 Udgang 3 Udgang 4 Udgang 5 Værdi:

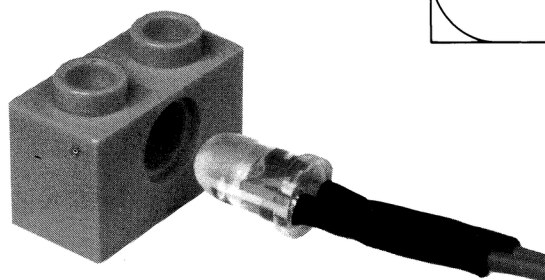
R25	R35	R45	R55	R65	R75 ... 680 ohm
R26	R36	R46	R56	R66	R76 .. 2,7 kohm
R27	R37	R47	R57	R67	R77 68 ohm
R28	R38	R48	R58	R68	R78 ... 270 ohm
R29	R39	R49	R59	R69	R79 68 ohm
R30	R40	R50	R60	R70	R80 .. 3,3 kohm
R31	R41	R51	R61	R71	R81 ... 270 ohm
R32	R42	R52	R62	R72	R82 ... 10 kohm
R33	R43	R53	R63	R73	R83 .. 2,2 kohm
R34					10 kohm
T3					BC557
T4	T9	T14	T19	T24	T29 BC547
T5	T10	T15	T20	T25	T30 BD136
T6	T11	T16	T21	T26	T31 BD135
T7	T12	T17	T22	T27	T32 BC557
T8	T13	T18	T23	T28	T33 BC547
D3	D6	D9	D12	D15	D18 . LED 3mm
D4	D7	D10	D13	D16	D19 1N4003
D5	D8	D11	D14	D17	D20 1N4003
IC4	IC5	IC6	IC7	IC8	IC9 PC817 optok

Spændingsforsyning:

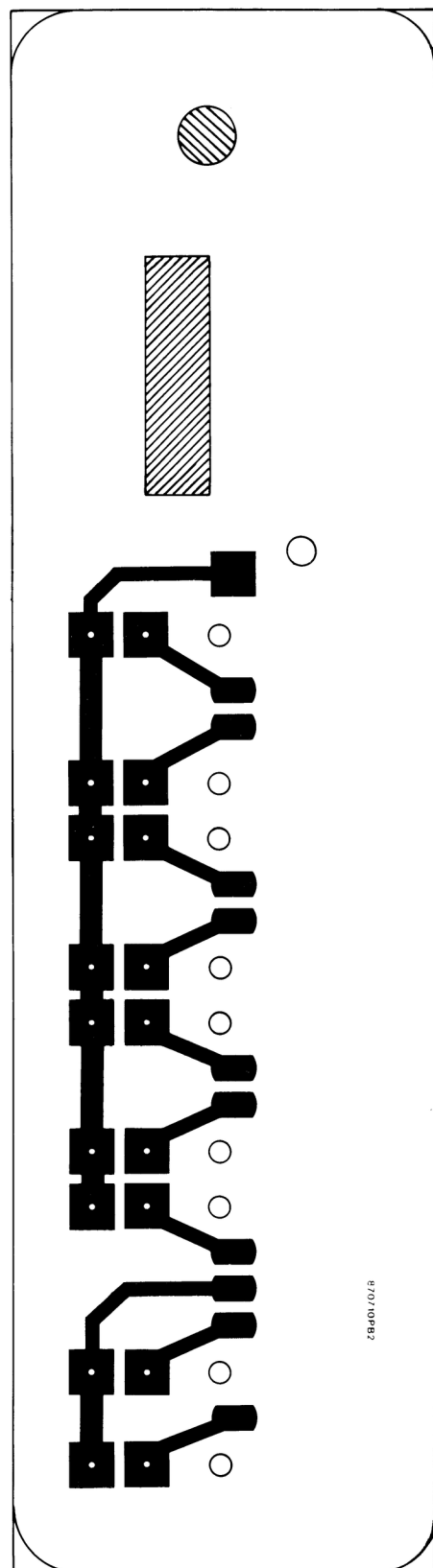
IC10	LM317 og køleplade
IC11	78L08
D21-26	1N4003
D27	1N4148
D28	LED, 3 mm, gul
D29	1N4003
P3	4,7 kohm, lille, vandret trimp.
C3	1000 μ F/25V elektrolyt
C4	2200 μ F/25V elektrolyt
C5	1 μ F/16V elektrolyt
C6	1000 μ F/25V elektrolyt
C7	1 μ F/16V elektrolyt
R85	220 ohm
R86	270 ohm

Yderligere anvendes:

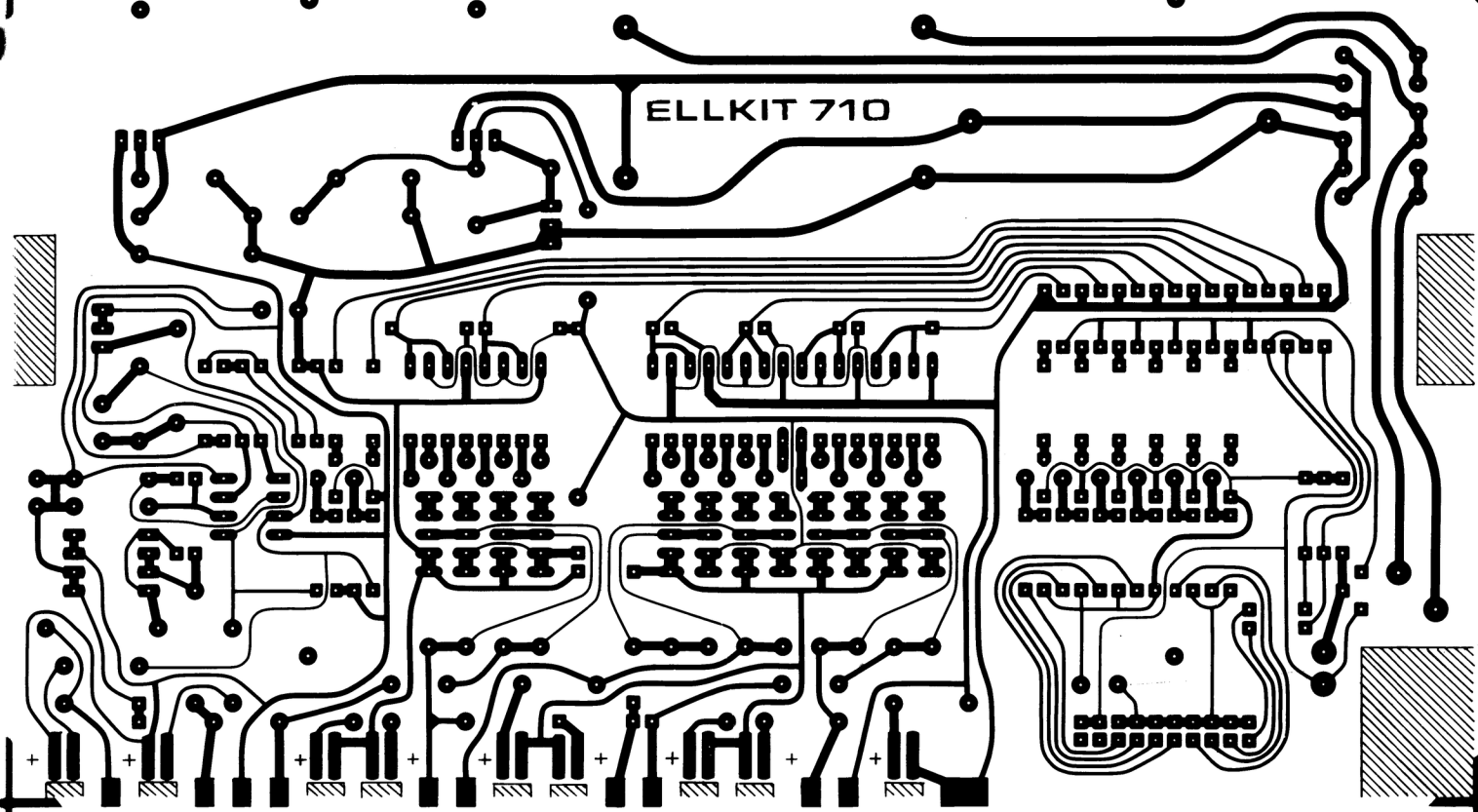
- 1 indbygningskasse, med eller uden håndtag.
 - 1 transformator 6V/7,5VA (1,5A) + indbygningskasse
 - 18 Bøsninger/nitter (EL-FI).
 - 4 loddespyd.
 - 1 hanstik, 20 polet i vinkel (shrouded male-header).
 - 1 trykomskifter.
 - 1 DC bøsning.
 - 1 Maskinskrue 3x8 mm med møtrik.
 - 4 selvsikrørende 3x6 mm skruer.
- Monteringsledning.



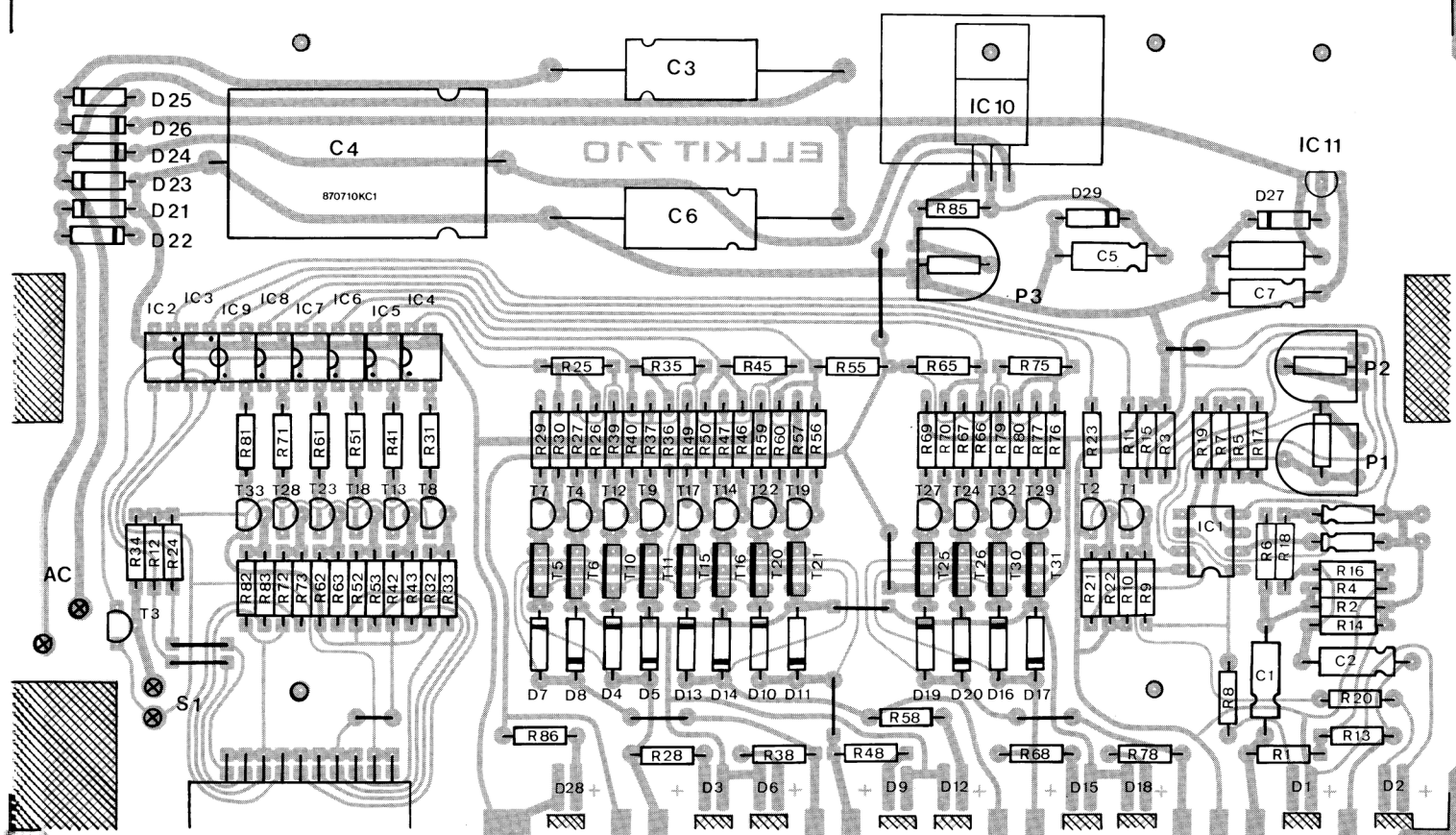
ELKIT 710, Interfacebox set fra loddessiden.



870/10P82



Hovedprintet til Universal Interface, Robox 710. Printet er set fra kobbersiden i forhold 1:1.



Priser:

Byggesæt, ELLKIT ROBOX 710:	kr. 327,20
Print til ROBOX 710 (to stk.):	kr. 190,35
Indbygningsskase uden hank:	kr. 49,95
Indbygningsskase med hank:	kr. 64,55
Løs sensor, BPX95B:	kr. 8,05

Robox 710 samlet komplet incl. strømforsyning: kr. 998,00
Alle priser er incl. 22% moms.

Forhandler:

Interfacet kan købes som byggesæt eller samlet hos:
 EL-FI A/S - postbox 17 - 7000 Fredericia. Tlf.: 05 93 32 00.

2. del af artiklen Universal interface:

Robox710

Vi gør klar til at køre.

En datamat kan som bekendt ikke selv bore huller, grave grøfter eller male byen rød. Men den kan **styre** en boremaskine, en rende-graver eller en malersprøjte. En maskine, som styres af en datamat, kaldes en **robot** eller en **automat**. I de fleste hjem findes én eller flere robotter. En vaskemaskine er et eksempel på en husholdningsrobot. I denne artikel og de følgende vil vi mest beskæftige os med industrirobotter, som spiller en stigende rolle i produktion og transport og derfor også i mange menneskers hverdag. Det er naturligvis ikke meningen, at læseren nu skal styre ud i byen og købe f.eks. en svejserobot hos den nærmeste maskinhandler. Det ville helt sikkert få både husholdningsbudgettet og stuegulvet til at revne med et brag. I stedet vil vi i al stilfærdighed vise, hvordan man kan bruge nogle små robot-modeller, bygget af LEGO-klodser, til at skaffe sig indsigt i de grundlæggende principper for robot-styring.

Man skal som sædvanligt bruge noget materiel og noget programmel. Materiellet består i dette tilfælde af en Commodore 64/128 med diskteststation (eller båndoptager), en **styrebox** (system controller) og nogle LEGO byggeelementer. Styreboxen kan man enten bygge selv eller købe. I forrige nummer af P.E. findes en vejledning i, hvordan man kan bygge den, og oplysninger om, hvor man får fat på delene til den.

I en senere artikel vil jeg fortælle om andre styreboxe fra ind- og udland.

Programmet består af en lille COMAM-fortolker og lidt maskinkode, som skal bruges til at drive selve styreboxen. Som det kan ses af byggevejledningen, har styreboxen 6 **styrbare udgange**, én **fast udgang** og 2 **indgange**. De styrbare udgange, som kan bruges enkeltvis eller i par, har numrene 0-5, og indgangene har numrene 6-7. Vi skal altså bruge et styreprogram, som kan tænde eller slukke udgangene, og som kan 'lytte på indgangene'.

En liste over et **styreprogram** er vist længere fremme i denne artikel. Før det testes ind, skal man hente Comal-fortolkeren ind fra disketten (eller kassetten). Den fortolker, vi vil bruge til at begynde med, hedder 0.14 og er skrevet af UniComal. Den er ikke nær så stor og slagkraftig som version 2.01, der fås på kapsel. Hvis man ikke i forvejen kender til COMAL er den imidlertid særdeles interessant, og til vort formål er den langt bedre end f.eks. Simons Basic. I en senere artikel vil jeg skrive om, hvordan de store Comal-versioner til Commodore, Amstrad og PC kan bruges til styring af robotter. Lige nu gælder det imidlertid om at komme igang med at bruge COMAL.

Altså hent Comal ind fra disketten med kommandoen:

```
LOAD "*",8
```

og derpå:

```
RUN
```

Den første kommando henter en turbo lader ind fra disketten. Den anden kommando får turbo laderen til at hente COMAL ind (Det varer ca. 17 sek.).

COMAL starter med at skrive noget om sig selv på skærmen. Det er hverken vittigt eller uartigt, men derfor kan man naturligvis godt læse det. Skift til små bogstaver ved som sædvanligt at trykke <SHIFT>+<C=> (altså skiftentøglen sammen med Commodore-

re-tasten). Danskere, som har gået i skole i mere end 2 år, kan udmærket bruge små bogstaver. Skriv derpå kommandoen:

```
auto
```

og begynd at indtaste styre-programmet efter listen. Når programmet er tastet ind, trykker man en ekstra gang på <RETURN> tasten. Så holder COMAL op med at 'lave numre'.

Det er klart, at man skal være meget omhyggelig med talkonstanterne i programmet. Kontroller hellere en gang for meget end en gang for lidt. Gem programmet på diskette med kommandoen:

```
save "robox"
```

eller på kassette med kommandoen:

```
save "robox",1
```

Slut den færdige styrebox til 64'eren som beskrevet i byggevejledningen og skriv **kommandoerne** (husk <RETURN> efter hver):

```
run
```

```
robox
```

Hvis programmet virker, som det skal, afbryder det alle de styrbare udgange. Man kan se, at de røde lysdioder slukkes.

Tænd udgang nr. 0 med kommandoen:

```
on (0)
```

Sluk den igen med kommandoen:

```
off (0)
```

Prøv at tænde alle udgange med kommandoen:

```
for i=0 to 5 do on(i)
```

Sluk dem alle med kommandoen:

```
cut
```

Afprøv indgangene ved at skrive kommandoen:

```
print bit(6),bit(7)
```

Så skal programmet skrive to nuller på skærmen. Læg mærke til, at de to grønne lysdioder er slukket. Kortslet indgang nr. 6 med en stump ledning og bemærk, at den grønne lysdiode tændes. Skriv kommandoen:

```
print bit(6)
```

Nu skal programmet skrive et 1-tal på skærmen. Prøv på tilsvarende måde med indgang 7. Af denne test ser man, at **bit** er en Boolsk funktion, der giver værdien 0 (svarende til 'false'), når lysdioden er slukket, og værdien 1 (svarende til 'true'), når lysdioden er tændt. Hvis noget i prøvekursen går galt, skal man først undersøge, om

styreboxen er i orden ved at gentage den test, som er beskrevet i slutningen af byggevejledningen. Hvis der ingen fejl er i styreboxen, må man gå styreprogrammet omhyggeligt igennem. Man kommer let til at lave fejl, når man skal taste mange tal ind. Går prøvekørslen, som den skal, udvider vi først styreprogrammet med en meget nyttig procedure. Hvis styreprogrammet er skrevet nøjagtigt efter forskriften, slutter det med linjen:

```
Ø63Ø //
```

(' svarer til 'rem' i BASIC). Programmet skal altså fortsætte med linje 640. Derfor skal man skrive kommandoen:

```
auto 64Ø
```

og derpå indtaste følgende linjer:

```
Ø64Ø func watch closed
Ø65Ø   t1:=peek(16Ø)
Ø66Ø   t2:=peek(161)
Ø67Ø   t3:=peek(162)
Ø68Ø   return (t1*256+t2)*256+t3
Ø69Ø endfunc watch
Ø70Ø //
Ø71Ø proc wait(t) closed
Ø72Ø   final:=watch+t*6Ø
Ø73Ø   while watch<final do null
Ø74Ø endproc wait
Ø75Ø //
```

Stands den automatiske nummerering ved to tryk på <RETURN> og afprøv det udvidede styreprogram med kommandoerne:

```
run
wait(3)
```

Så skal markøren (kursoren) forsvinde i 3 sekunder. Gem programmet på diskette med kommandoen:

```
save "robox/2"
```

og på bånd med kommandoen:

```
save "robox/2",1
```

og så kører vi!

Slut en motor til udgang nr. 0. Stikket skal sidde lodret over 0'et. Giv kommandoen:

```
on (Ø)
```

Lad motoren køre lidt og skriv derpå kommandoen:

```
off (Ø)
```

Udfør en tilsvarende prøvekørsel med de øvrige udgange. Slut igen motoren til udgang nr. 0 og skriv følgende **kommandolinje**:

```
on(Ø); wait(5); off(Ø)
```

(husk at trykke på <RETURN>, når linjen er skrevet). Det skal få motoren til at køre i 5 sekunder. Styr markøren op over kommandolinjen (CRSR UP), ret tallet i parentes efter **wait** til 8 og tryk igen <RETURN>. Nu skal motoren naturligvis køre i 8 sekunder. Tag stikket ud af udgang 0 og sæt det på tværs, så det ene ben sidder i den øverste bøsning ved udgang 0 og det andet i den øverste bøsning ved udgang 1. For fremtiden vil jeg betegne denne tilslutning som 0-1 og tilsvarende for de øvrige udgange. Brug atter kommandolinjen:

```
on(Ø); wait(5); off(Ø)
```

Når motoren har kørt i 5 sekunder, så ret linjen til:

```
on(1); wait(5); off(1)
```

og læg mærke til, at motoren atter kører i 5 sekunder, men med modsat omløbsretning. Sådan!

Træk stikket til motoren ud og sæt i stedet stikket til en lysdiode (se byggevejledning) i udgang 0. Skriv derpå denne procedure (brug auto-kommandoen):

```
Ø76Ø proc blink
Ø77Ø   on(Ø)
Ø78Ø   wait(.3)
Ø79Ø   off(Ø)
Ø80Ø   wait(.3)
Ø81Ø endproc blink
Ø82Ø //
```

(husk at slutte med to gange tryk på <RETURN>. Skriv så kommandoen:

```
run
```

for at få den nye procedure gjort klar til kørsel. Man skal huske, at COMAL er en semicompiler. Derfor skal man altid skrive **run**, når et program er blevet ændret, før procedurerne kan udføres. Afprøv med kommandoen:

```
blink
```

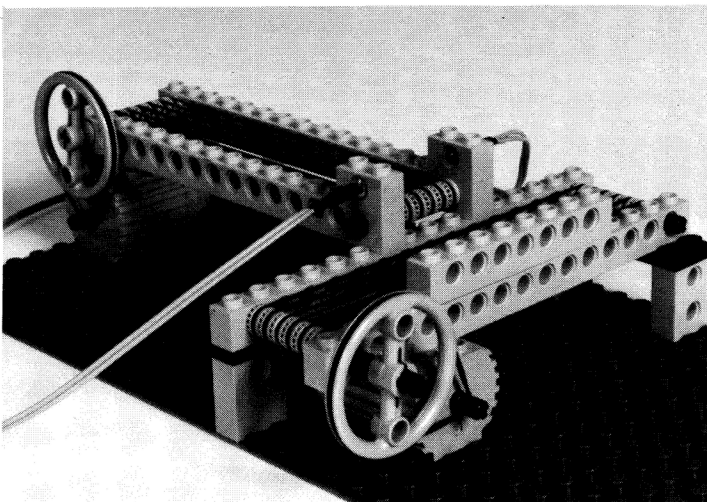
og derpå med kommandolinjen:

```
for i=1 to 1Ø do blink
```

Dioden skal blinke 10 gange efter den sidste kommando.

Sæt lysdioden fast på en byggeplade og sæt en fotosensor (se byggevejledning) fast overfor den i en afstand, så man kan tælle tre knopper mellem lyset og sensoren. Slut fotosensoren til indgang nr. 6. Brug igen kommandolinjen ovenfor og læg mærke til, hvordan fotosensoren giver en impuls, hvor gang lysdioden blinker. Det skal se ud, som om den grønne lampe ved indgangen 'tæller med'.

Projektforslag 1. Byg to trafikfyr med grønne, gule og røde lysdioder. Slut dioderne i det ene trafikfyr til udgangene 0, 1 og 2, og i det andet til 3, 4 og 5. Lad de to trafikfyr regulere hver sin trafikretning og skriv en procedure, som får de to fyr til at arbejde korrekt sammen. Hvis man gerne vil have alle fire retninger i krydset dækket, skal man blot dublere de to lysfyr. Udgangene kan med lethed trække to lysdioder hver. Proceduren kan f.eks. begynde sådan:




```
on(0) //grøn mod nord
on(5) //rød mod vest
wait(5) //hold i 5 sek.
off(0) //sluk grøn nord
on(1) //gul mod nord
```

Når proceduren er skrevet færdig og man har tastet **run**, kan krydset afprøves med kommandolinjen:

```
for i=1 to 1000 do lyskryds
```

Denne linje bevirker, at proceduren lyskryds kaldes 1000 gange. Den kan standses med et tryk på <STOP>-tasten.

Man kan også lave et 'lysshow' med forskelligt farvede lysdioder. Man behøver kun bruge de tre procedurer **on**, **off** og **wait**.

Varer til Jensen og Jørgensen

Vort næste projekt skal være bygning og styring af et dobbelt transportbånd. Fig. 1 viser den samlede opstilling, og på fig. 2 kan man se, hvordan de elastikker, som udgør selve båndet, er spændt op over 6 små remhjul, samlet på en aksel. I slutningen af artiklen findes en liste over de elementer, der skal bruges til modellen.

Slut den motor, der driver det langsgående bånd, til udgang nr. 0, og motoren, der driver det tværgående bånd, til udgangene 2-3. På hver sin side af det langsgående bånd er monteret en lysdiode og en fotosensor (fig. 1). Slut lysdioden til den faste udgang og fotosensoren til indgang nr. 6.

Sørg for, at programmet 'robex/2' er i datamatens arbejdslager og føj de to følgende procedurer til. Bemærk, at procedurerne er angivet uden linjenumre. Brug **auto**-kommandoen og lad nummerringen starte med et nummer, som er 10 højere end det sidste linjenummer i 'robex/2':

```
proc awaitbit(p)
  while bit(p) do null
endproc
//
proc awaitkey
  while key$=chr$(0) do null
endproc
//
```

Giv **run**-kommandoen og afprøv de to procedurer med følgende kommandolinje:

```
awaitkey; on(0); awaitbit(6); off(0)
```

(husk <RETURN> til slut). Læg en almindelig 2x2 klods på det langsgående bånd ved den ende, hvor motoren sidder. Tryk så på en taste og iagttag, hvad der sker. Linjen skulle gerne virke således: Først venter systemet på, at der trykkes på en taste. Så starter motoren, der er sluttet til udgang 0, og systemet venter på, at der kommer signal fra indgang 6. Dette signal kommer, så snart klodsen kaster skygge over fotosensoren, og derpå afbrydes strømmen til motoren omgående. Resultatet er, at klodsen bliver hængende i lysbarrieren.

Disse erfaringer vil vi bruge til at skrive en procedure, der kan drive det langsgående bånd. Den kan f.eks. se sådan ud:

```
proc frem
  awaitkey
  on(0)
  awaitbit(6)
  wait(.5)
  off(0)
endproc
//
```

Giv **run**-kommandoen og afprøv proceduren med følgende kommando:

frem

Læg klodsen på båndet og tryk på en taste. Hvorfor står der **wait (.5)** efter **awaitbit (6)**?

Læg mærke til, at proceduren **frem** startes ved, at man blot skriver dens navn. Man siger også, at proceduren **kaldes**, og selve ordren **frem** kaldes et **procedurekald**. Ordre, som f.eks. **on (0)**, **awaitkey** og **awaitbit (6)**, er andre eksempler på procedurekald. Tallene i parenteser efter kaldet kaldes **parametre**. I moderne programmering anses procedurer med parametre for en selvfølge. Alene af den grund burde BASIC forlængst have været henvist til nærmeste museum for fortidsuhyrer.

Så kan vi fortsætte med at få styr på det tværgående transportbånd. Hertil kan vi benytte følgende to procedurer:

```
proc hoejre
  on(2)
  wait(1.5)
  off(2)
endproc
//
proc venstre
  on(3)
  wait(1.5)
  off(3)
endproc
//
```

Tast **run** og afprøv proceduren **hoejre** på følgende måde: Læg 2x2 klodsen midt på det tværgående bånd og giv kommandoen:

hoejre

Så skal klodsen blive ført til højre på båndet og kastet ud af det. Hvis den ikke når at blive kastet ud, skal parameteren i kaldet af **wait** gøres større, så båndet kører i længere tid.

Afprøv proceduren **venstre** på tilsvarende måde. Hvis de to procedurer virker, som de skal, så prøv med følgende kommandolinjer:

```
frem; hoejre
```

Læg klodsen på det langsgående bånd og tryk på en taste. Klodsen skal nu køre frem, passere lysbarrieren og blive kastet ud til højre.

Nu har vi skrevet alle de grundlæggende procedurer til styring af det dobbelte transportbånd. Indtil nu har vi afprøvet disse procedurer ved at **kalde dem direkte** i kommandoer eller kommandolinjer. Man kan naturligvis også kalde dem i programlinjer og derved opbygge et totalt styreprogram. Jeg vil foreslå læseren at benytte følgende udvidelse af det program, som er skrevet indtil nu:

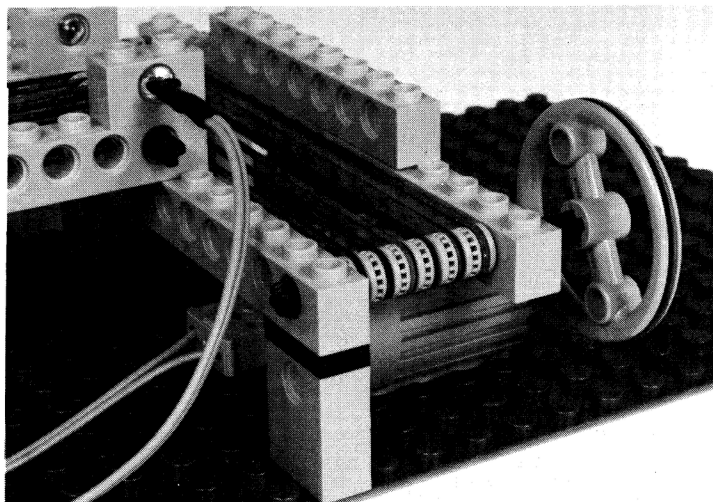


Fig. 2

```
proc bestil
  print "jensen skal have",
  input hj
  print "jørgensen skal have",
  input vn
endproc
//
proc fordel(x,y)
  for i=1 to x do frem; hoejre
  for i=1 to y do frem; venstre
endproc
//
proc go
  bestil
  fordel(hj,vn)
  print "tak for ordren"
endproc
//
```

Tast **run** og afprøv hele systemet ved at skrive:
go

Afgiv derpå bestilling til d'herrer Jensen og Jørgensen, læg klodser på det langsgående bånd og tryk på en taste, hver gang en klods er klar til levering. Bare deruda'.

I næste artikel vil jeg vise, hvordan man kan gøre udskrifter m.v. lidt pænere. I mellemtiden kan læseren forsøge sig med forskellige projekter.

Projektforslag 2. Skriv en procedure, som udskriver følgesedler til Jensen og Jørgensen, når 'varerne' er leveret. COMAL bruger print-sætninger på samme måde som BASIC. Man kan f.eks. kalde proceduren for **følgesedler**. Kaldet til den skal stå i hovedproceduren **go** i stedet for linjen med 'tak for ordren'. Husk nu at skrive pænt 'tak for ordren' på følgesedlerne. Man kan udbygge proceduren **bestil**, så den beder om både navn, adresse og styktal.

Disse oplysninger kan så bruges i proceduren **følgesedler**, og man kan skrive endnu en procedure med navnet **faktura**, som udskriver 'rigtige' fakturaer med navn, adresse, styktal, enhedspris og samlet pris. Var der nogen, der sagde CIM?

Projektforslag 3. Sæt endnu en lysdiode og en fotosensor på det langsgående bånd, men sådan at de sidder ovenpå hver sin 1x2 sten. Slut fotosensoren til indgang nr. 7 og stikket til lysdioden ind i siden på det stik, der sidder i den faste udgang, så begge lysdioder får strøm fra denne.

Denne nye lysbarriere kommer altså 'en etage højere op'. Den skal sidde 3 knopper før den, vi hidtil har arbejdet med. Nu vil en almindelig 2x2 sten passere uantastet igennem den nye barriere og først blive 'set' af den sidste.

Hvis man laver en 'pakke' bestående af 2 stk. 2x2 sten, vil denne blive registreret af begge sensorer. Benyt dette til at lave et sorteringsanlæg, der virker sådan, at alle enkeltsten bliver lagt ud til højre, og alle dobbeltsten bliver lagt ud til venstre. Udstyr evt. programmet med to tællere, så det bagefter kan fortælle, hvor mange enkelt- og hvor mange dobbeltsten, der er gået igennem systemet.

Næste artikel: Man må tjene på karrusellen..., hvor man kan læse om motorstyring ved hjælp af tællere samt flere lækre COMAL procedurer til bl.a. skærmstyring.

Tekst: Børge R.Cristensen.

Styreprogram

```
0010 proc on(x)
0020   poke 177,x
0030   sys 52992
0040 endproc on
0050 //
```

```
0060 proc off(x)
0070   poke 177,x
0080   sys 53006
0090 endproc off
0100 //
0110 func bit(x)
0120   poke 177,x
0130   sys 53020
0140   return peek(177)
0150 endfunc bit
0160 //
0170 proc cut
0180   poke 56577,0
0190 endproc cut
0200 //
0210 proc robox
0220   for i:=0 to 63 do
0230     read a
0240     poke 52992+i,a
0250   endfor i
0260   //
0270   poke 56579,63
0280   poke 56577,0
0290   a:=peek(56578)
0300   poke 56578,4*(a div 4)+(a mod 2)
0310 endproc robox
0320 //
0330 data 164,177
0340 data 185,58,207
0350 data 13,1,221
0360 data 41,63
0370 data 141,1,221
0380 data 96
0390 //
0400 data 164,177
0410 data 185,58,207
0420 data 73,63
0430 data 45,1,221
0440 data 141,1,221
0450 data 96
0460 //
0470 data 165,177
0480 data 201,7
0490 data 240,9
0500 data 173,1,221
0510 data 41,64
0520 data 240,9
0530 data 208,10
0540 data 173,1,221
0550 data 16,2
0560 data 48,3
0570 data 169,0
0580 data 44,169,1
0590 data 133,177
0600 data 96
0610 //
0620 data 1,2,4,8,16,32
0630 //
```

Stykliste til det dobbelte transportbånd. Fra LEGO TECHNIC:

- 1 stk. byggeplade mindst 32x16
- 24 stk. lille remhjul
- 2 stk. rat (stort remhjul)
- 3 stk. aksel, 6-knops længde
- 1 stk. aksel, 8-knops længde
- 12 stk. elastik, stor
- 2 stk. elastik, mellem
- 4 stk. bjælke, 1x16
- 1 stk. bjælke, 1x8
- 1 stk. bjælke, 1x6
- 8 stk. bjælke, 1x2
- 1 stk. plade, 2x6
- 1 stk. plade, 1x2
- 2 stk. TECHNIC motor - 4.5 V4
- 6 stk. forbindelsesledning, 75 cm
- Samt et antal alm. 2x2 byggesten.
- Endvidere lysdiode og fotosensor som beskrevet i byggevejledningen til styreboxen.

Fortsættes