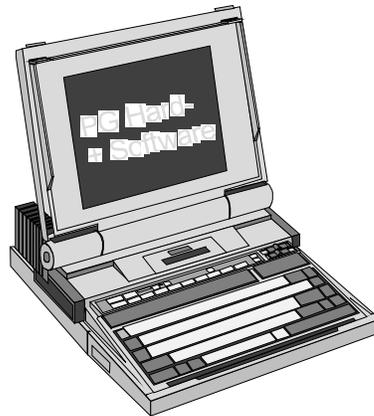


TECHNIK

Die Benutzung der Druckerschnittstelle
für einfache Steuerungsaufgaben

IBM und Kompatible



Vorwort

Zur Optimierung technischer Verfahren ist der Computer nicht mehr wegzudenken. Anpassungen auf veränderte Funktionsabläufe lassen sich softwaremäßig schnell realisieren, die Hardware kann meist zu einem großen Teil übernommen werden. Die vorliegende Schrift möchte den Kollegen Mut machen, sich mit diesem interessanten Thema auseinanderzusetzen. Der Vorteil der vorgestellten Hardware liegt weiterhin in der kostengünstigen Realisierung (45 DM für die Grundausstattung) der Experimente. Bei vielen Anbietern müssen erst 300 - 400 DM investiert werden, um mit einem PC Steuerungsaufgaben durchzuführen. Die Platinen sind kompatibel zum ESTU-System und können mit diesen kombiniert werden.

Zur Programmierung der Software wurde Turbo Pascal 3.0 herangezogen, da diese Sprache von der Schule für den Informatikunterricht angeschafft wurde. Wie an anderer Stelle vielfach beschrieben, erlaubt Pascal ein komfortables Editieren und strukturiertes Programmieren. Hiermit will ich mich aber nicht in einen Streit begeben, welches die sinnvollste Programmiersprache für den Technikunterricht sei. Die neuen Turbo Versionen haben aus didaktischer Sicht meines Erachtens den Nachteil, daß sie schon für grundlegende Befehle (z.B. Bildschirm löschen) Dateien einbinden müssen.

Aus den bisherigen Erfahrungen kann ich nur sagen, daß sich die Beschäftigung mit diesem Thema voll für den Technikunterricht auszahlt. Die Schüler/innen sind sehr motiviert sich den Anforderungen zu stellen. Lauflicht-, Ampel- und 7-Segment-Anzeigensteuerung werden von Schülern des 8. Jahrgangs bereits mit Leichtigkeit gelöst. Im 11. Jahrgang kann der Positioniertisch einfach über den Computer gesteuert und geregelt werden. Ein einfach aufzubauender DA/AD-Wandler (ca. 25 DM) ermöglicht den Einstieg in die Analogtechnik

Inhalt

- 1 Hard- und softwaremäßige Voraussetzungen**
- 2 Hardware**
 - 2.1 Adapter für die Druckerschnittstelle
 - 2.2 Verbindungskabel Computer - Adapter
 - 2.3 LED - Anzeige
 - 2.4 Verbindungskabel für LED - Anzeige
 - 2.5 Ansteuerung eines Gleichstrommotors
 - 2.6 Ansteuerung eines Schrittmotors
 - 2.7 Ansteuerung eines Schrittmotors mit Decoder
- 3 Software**
 - 3.1 Ermittlung der Portadresse der Druckerschnittstelle
 - 3.2 Programmierung der 8-Bit Aus/Eingabe
 - 3.3 Programmierung der zusätzlichen 3-Bit Eingabe
 - 3.4 Programmierung der zusätzlichen 2-Bit Ausgabe
- 4 Anwendungs-Software**
 - 4.1 Lauflicht
 - 4.2 Ampel
 - 4.3 Gleichstrommotor
 - 4.4 Schrittmotor
 - 4.5 Programmierung des Schrittmotor-Decoders
 - 4.6 Styroporschneidemaschine (CAD-CAM System)
- 5 Platinenlayouts**
- 6 Beschaffungshilfe**
- 7 Literaturliste**

Anhang

Arbeitsblätter - Projekte mit der Centronicsschnittstelle

Fußgängerampel
Ampel-Kreuzung
Positioniertisch
Schrittmotor
CAD-CAM

1 Hard- und softwaremäßige Voraussetzungen

Die gesamten Ausführungen beziehen sich in erster Linie auf das Buch "Messen und Experimentieren" aus dem Dümmler Verlag. Wer sich ausführlich informieren will, sollte unbedingt das Buch (ca. 40 DM) erwerben, da viele weiterführende Versuche erläutert werden.

Hardware:

- IBM und Kompatible PC's, XT's oder AT's
- min. 256 kB Speicherkapazität für die vorgestellte Software
- Centronics (Drucker) Schnittstelle
- Grafikkarte + Monitor im Prinzip gleich, da die Beispielprogramme nur im Textmodus ablaufen. Erprobt wurden die Programme auf einem AT mit Hercules-Karte.

Software:

Im Prinzip jede Software, die es erlaubt, auf Schnittstellen zuzugreifen. Die in dieser Schrift verwendete Sprache ist Turbo Pascal 3.0.

2. Hardware

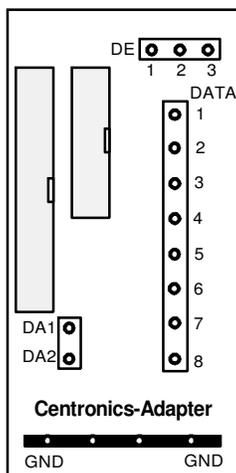
2.1 Adapter für die Druckerschnittstelle

Die im vorab genannten Buch vorgestellte Experimentierplatine (75mm * 130mm) benutzt einen Centronics-Stecker und 4mm Buchsen für Versuche. Um kompatibel zu dem 1,3mm Lötstift- bzw. weit verbreiteten ESTU-System zu sein, habe ich ein Layout mit den Abmessungen 50mm * 100mm entworfen.

Die an der Druckerschnittstelle üblichen Bezeichnungen wie INIT, STROBE, ACKNLG usw. werden durch die durchgängig im Buch benutzten Bezeichnungen

DATA 1..8	8 Ein- Ausgabeleitungen
DE1..DE3	3-Bit zusätzliche Eingabe
DA1,DA2	2-Bit zusätzliche Ausgabe

ersetzt und auch hier wegen der leichteren Übertragbarkeit benutzt.

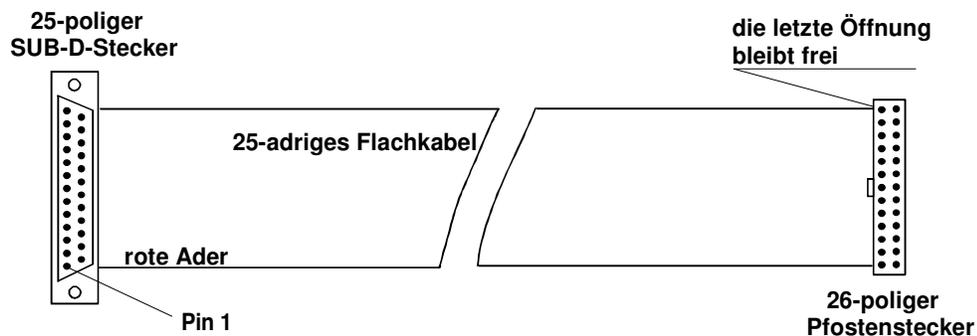


Belastbarkeit:

- DATA 1..8 Ausgang: $I < 100\text{mA}$ je Leitung
Eingang: Die Leitungen können scheinbar bedenkenlos mit GND verbunden werden (Kurzschluß), $I_{\text{max}} < 100\text{mA}$. Bei meinem Rechner lag I_{max} bei 47mA . Soll das Low-Signal über einen Widerstand erzeugt werden, darf der Widerstandswert $10\ \Omega$ nicht übersteigen.
- DE1..DE3 Eingang: Für ein definiertes Low-Signal, Verbindung mit GND, sollte ein einzubauender Widerstand zwischen 0 und $6\ \text{k}\Omega$ liegen.
- DA1, DA2 Ausgang: kurzschlußfest, der Strom wird auf $0,4\text{mA}$ begrenzt

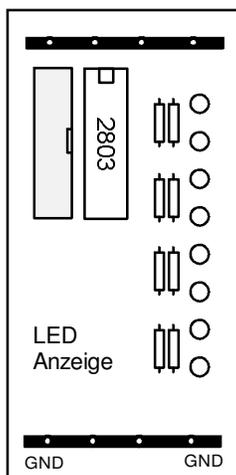
2.2 Verbindungskabel Computer-Adapter

Darstellung ohne Zugentlastung



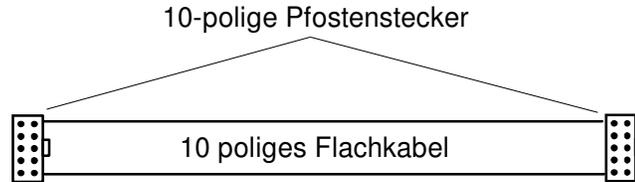
2.3 LED - Anzeige

Sollte bereits eine 8-Bit-Anzeige vorliegen, kann diese an den Lötstiften DATA 1..8 angeschlossen werden. Die hier vorgestellte Anzeige hat den Vorteil, daß die Verbindung über den 10-poligen Pfostenstecker schneller hergestellt wird.



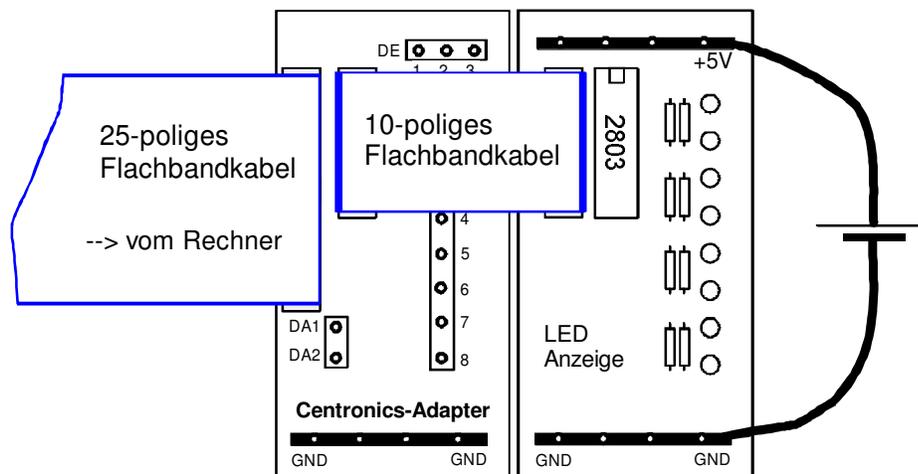
Die Anzeige wird benutzt, um Aktivierungszustände auf den Datenleitungen anzuzeigen. Verschiedene Lauflichtprogramme können getestet und zur Not kann die Anzeige auch als Ampelersatz erhalten. Die Anzeige kann bei anderen Applikationsschaltungen, die die Lötstifte 1..8 benutzen, zugeschaltet bleiben, um die Zustandswechsel (0-1) beobachten zu können. Das Layout der Anzeige folgt später.

2.4 Verbindungskabel für LED - Anzeige



2.5 Schaltungsvorschläge

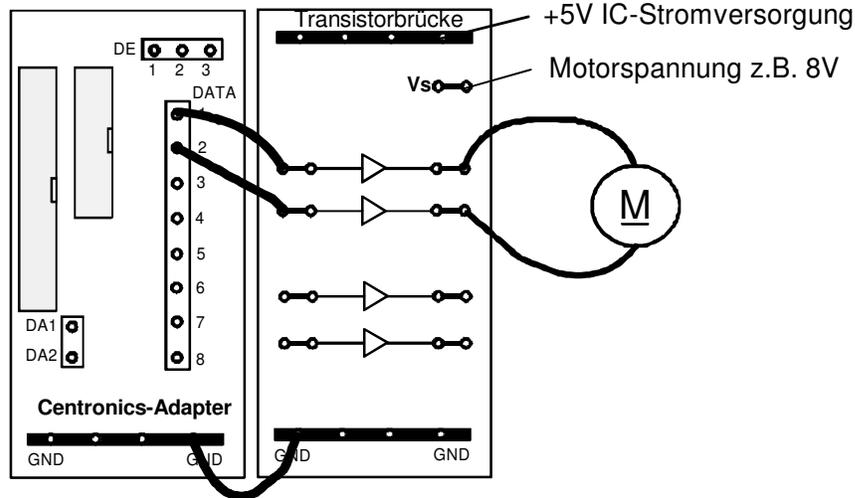
2.5.1 Grundschaltung der LED - Anzeige



Eine 4,5V Batterie reicht zum Austesten. Die GND-Verbindung erfolgt über das 10-polige Flachkabel.

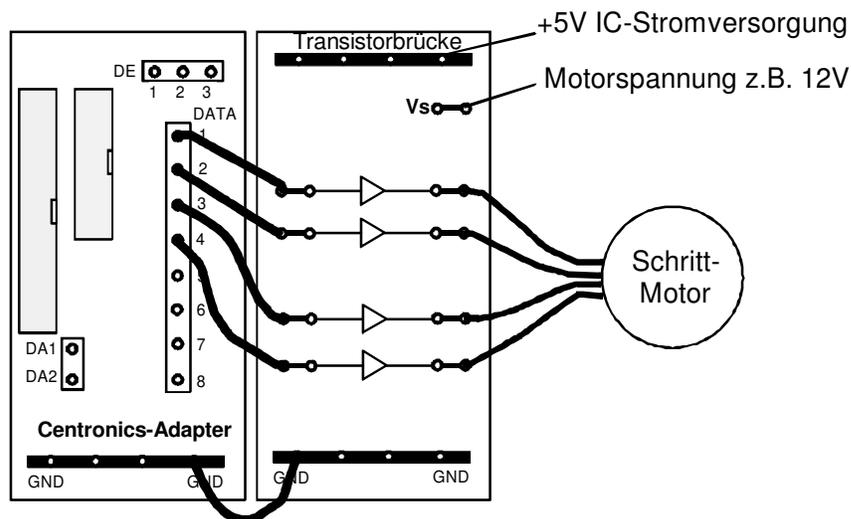
2.5.2 Ansteuerung eines Gleichstrommotors

Eine einfache Wendepolschaltung lässt sich mittels Relais realisieren, entsprechende Platinen sind meist vorhanden, die Ansteuerung bekannt. Einfacher im Aufbau und für viele Anwendungszwecke (1A Belastbarkeit pro Brücke) ausreichend ist die elektronische Realisierung. Das auf der Platine verwendete IC (L293D) und seine Ansteuerung ist ausführlich in der unter 2 genannten Literatur (S.137) erläutert. Eine fertig aufgebaute Platine ist aus dem ESTU-System erhältlich. Wegen der Verbreitung dieser Platine und um die Ansteuerung zu verdeutlichen, habe ich das Bestückungslayout der ESTU-Platine verwendet. Die Verbindung zum Rechner ist aus Vereinfachungsgründen hier und in den nachfolgenden Zeichnungen nicht eingezeichnet worden.



2.5.3 Ansteuerung eines Schrittmotors

Die Ansteuerung soll am Beispiel eines bipolaren Schrittmotors mit Hilfe der oben vorgestellten Platine erfolgen. Die Ansteuerung eines unipolaren Schrittmotors ist ebenfalls ohne Probleme mit der im ESTU-System als Transistorgruppe benannten Platine realisierbar. Die Programmierung unterscheidet sich nicht und wird unter 4.4 vorgestellt.



2.5.4 Ansteuerung eines Schrittmotors mit Decoder ³⁾

Bereits bei 2 angeschlossenen Schrittmotoren sind die Datenleitungen erschöpft. Für weitere Steuerungszwecke stehen nur noch die Ausgabeleitungen DA1 und Da2 zur Verfügung. Benötigt man weitere Ausgabeleitungen, dann ist ein Hardware-Decoder erforderlich. Bei der Schrittmotoransteuerung ist dies relativ einfach, da zwei Leitungen invertiert angesteuert werden. Der vorgestellte Decoder vereinfacht die Ansteuerung noch einmal, d.h. zur Ansteuerung ist nur noch ein Taktimpuls erforderlich, der jeweils auf eine der Steuerleitungen gelegt wird. Die Ansteuerung kann dadurch recht einfach über eine Taktgeberplatine (ohne Computer) überprüft werden.

Ansteuerungsprinzip:

Takt	DATA1	DATA2	DATA3	DATA4	Dezimalwert
1	0	1	0	1	10
2	1	0	0	1	9
3	1	0	1	0	5
4	0	1	1	0	6

Da $\overline{\text{DATA2}} = \text{DATA1}$ und $\overline{\text{DATA4}} = \text{DATA3}$ benötigen wir nur DATA1 und DATA3 zur Darstellung des Bitmusters.

Takt	DATA1	DATA3
1	0	0
2	1	0
3	1	1
4	0	1

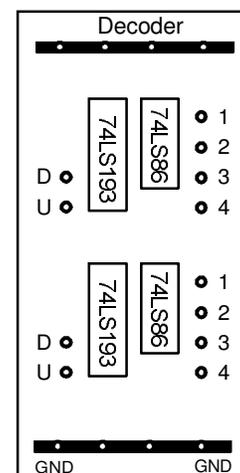
Da es sich nur um vier Takte handelt, kann die Ansteuerung durch einen 2-Bit Binärzähler erfolgen, d.h.:

Takt	Q _B	Q _A	DATA1	DATA3
1	0	0	0	0
2	0	1	1	0
3	1	0	1	1
4	1	1	0	1

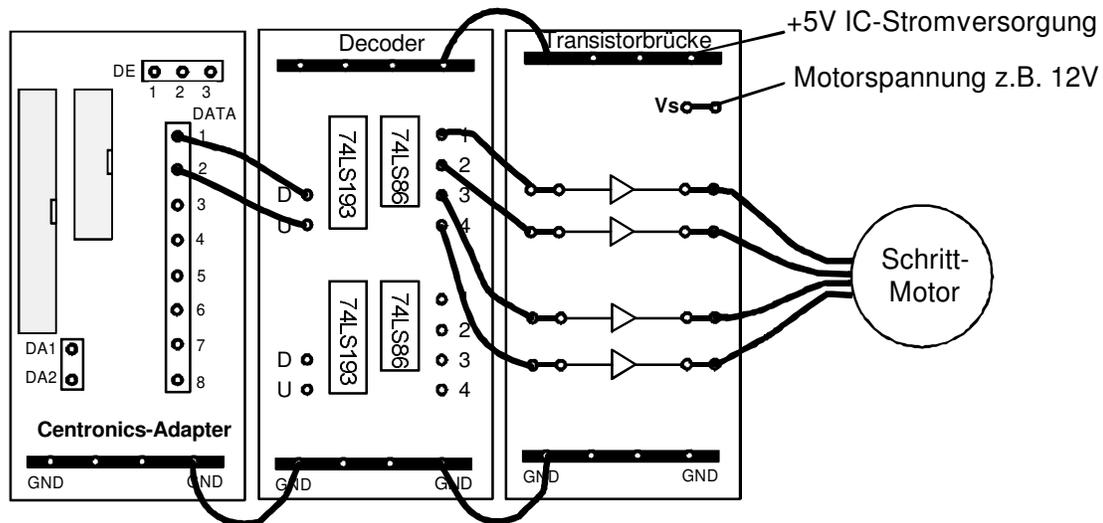
DATA3 kann direkt durch die höherwertige Leitung Q_B des 2-Bit Binärzählers angesteuert werden. DATA1 wird durch eine Exklusiv-ODER Verknüpfung der vier Taktzustände dargestellt. Realisiert wird die Schaltung durch den vorwärts/rückwärts Zähler 74193 und das Exklusiv-Oder Gatter 7486.

Auf dem nachfolgend vorgestellten Layout sind zwei Decoder installiert, d.h. 2 Schrittmotoren können mit 4-Bit angesteuert werden. 4-Bit stehen zusätzlich zu Steuerungszwecken zur Verfügung.

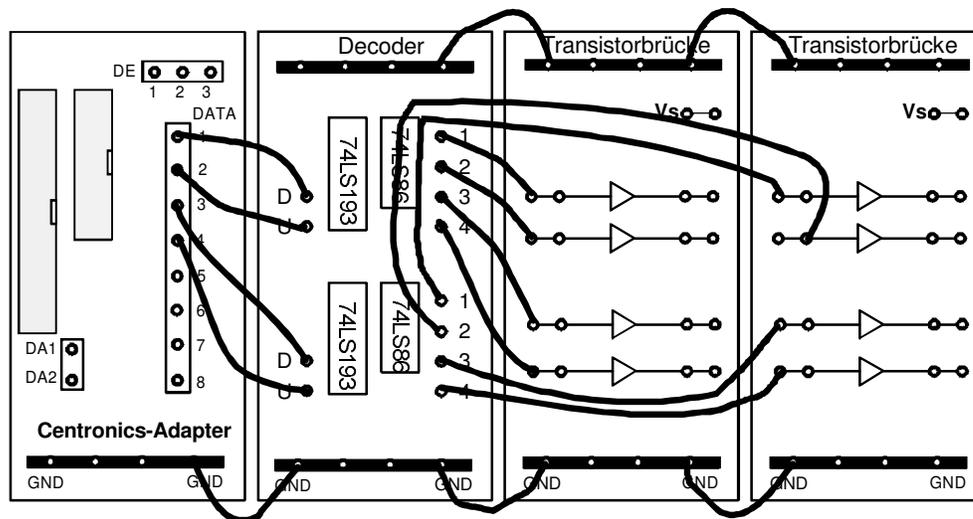
Decoder-Platine



Nachfolgend die Anschlußbelegung eines Schrittmotors an die Decoderplatine, D und U stehen für auf- und abwärts zählen:



2.5.5 Ansteuerung einer Styroporschneidemaschine (2 Schrittmotoren)



Da die Decoderplatine zwei Schrittmotoren steuern kann, ist eine zweite Transistorbrücken-Platine erforderlich. Sollen Relais- oder andere Platinen gesteuert werden, stehen die DATA Leitungen 5..8 und DA1, DA2 zur Verfügung. Aus Vereinfachungsgründen sind die Verbindungsleitungen zu den Schrittmotoren (siehe oben), das Verbindungskabel zum Rechner und die Stromzuführungen fortgelassen. Die hier vorgestellte Schaltung wird bei dem unter 4.6 vorgestellten CAD-CAM System benutzt. Zusätzlich werden die DATA Leitungen 5 und 6 zur Betätigung zweier Relais zwecks Zuschaltung der Versorgungsspannungen der Schrittmotoren und des Hitzedrahtes von der Software unterstützt. In einer ersten Ausbaustufe kann auf die Relaischaltung verzichtet werden, wenn die Stromversorgung fest geschaltet wird.

3. Software

3.1 Ermittlung der Portadresse der Druckerschnittstelle

Die Schnittstellen LPT1..3 haben die Basis-Adressen 632, 888 und 956. Wenn kein Drucker angeschlossen ist, können die Adressen m.E. bedenkenlos ausgetestet werden, da diese Adressbereiche für die Druckersteuerung reserviert sind und es zu keinem Konflikt mit einem Programm kommen kann. Benutzer einer Hercules-Karte mit zusätzlicher Centronics-Schnittstelle können eine Schnittstelle für den Drucker belassen und die andere für Steuerungszwecke nutzen. Bei Fehlen einer 2. Drucker-Schnittstelle kann diese preiswert (Fa. Conrad 39,-DM) nachgerüstet werden. Wer es genau wissen will, sollte es mit dem nachfolgenden Programm versuchen. Falls vorhanden, werden zwei Basis-Adressen auf dem Bildschirm ausgegeben.

PROGRAM Druckeradresse_suchen;

```
VAR LPT1, LPT2 : integer;
```

```
FUNCTION Adresse_existiert:boolean;
```

```
BEGIN
```

```
    LPT1:= Mem[$0040:$0008]+256*Mem[$0040:$0009];
```

```
    LPT2:= Mem[$0040:$000A]+256*Mem[$0040:$000B];
```

```
    Adresse_existiert := LPT1<>0;
```

```
END;
```

```
PROCEDURE Adresse_suchen;
```

```
BEGIN
```

```
    ClrScr;
```

```
    IF Adresse_existiert then
```

```
        BEGIN
```

```
            writeln ('Ausgabe der Adresse/n der Druckerschnittstelle/n');
```

```
            writeln;
```

```
            writeln ('LPT1 = ',LPT1);
```

```
            writeln ('LPT2 = ',LPT2);
```

```
        END
```

```
        ELSE
```

```
            BEGIN
```

```
                writeln ('Keine Centronics-Schnittstelle vorhanden !');
```

```
            END;
```

```
END;
```

```
{ Hauptprogramm }
```

```
BEGIN
```

```
    LPT1:=0;
```

```
    LPT2:=0;
```

```
    Adresse_suchen;
```

```
    repeat until keypressed;
```

```
END.
```

3.2 Programmierung der 8-Bit Aus/Eingabe

Stellvertretend für die anderen Adressen wird hier die Basisadresse 888 benutzt, die bei vielen Rechnern für LPT1 steht. Die nachfolgend vorgestellten Programme sind absichtlich sehr einfach gehalten, um das Wesentliche besser herausstellen zu können.

Program Ausgabe_1;

```
begin
  Port[888] := 1;      { DATA1 wird aktiviert (+5V) }
  Delay (500);        { willkürlich festgelegte Verzögerung }
  Port[888] := 17;    { DATA1 und DATA5 aktiviert }
  Delay (200);        { kürzere Verzögerungszeit }
  Port[888] := 0;     { ganz wichtig, alle Leitungen werden deaktiviert }
end.
```

Beispiel eines Eingabeprogramms über DATA1..8.

Program Eingabe_1;

var Wert : integer;

```
begin
  Port [888] := 255;      { Leitungen vorab auf 1 setzen }
  clrscr;                 { Bildschirm löschen }
  Wert := Port [888];    { Zuweisung an die Variable }

  writeln ('Das angelegte Bitmuster hat den Dezimalwert: ', Wert);

  {Bildschirmausgabe }

end.
```

Nicht verbundene (offene) Leitungen haben den Binärwert "1", mit GND verbundene Leitungen den Binärwert "0".

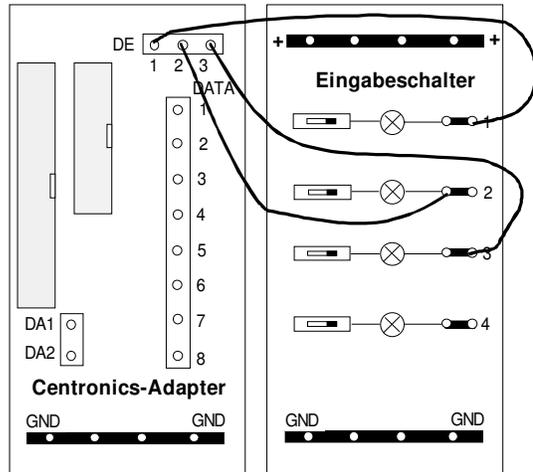
Beispiel: Die Leitungen 1,3,7,8 sind offen, die Leitungen 2,4,5,6 sind mit GND verbunden, das entspricht der Binärzahl:

Wertigkeit	128	64	32	16	8	4	2	1	
Binärzahl	1	1	0	0	0	1	0	1	, die entsprechende Dezimalzahl lautet 197.

Achtung: Dieses Eingabeverfahren ist zwar nach Schmidt/Weber (siehe Literaturverzeichnis) erlaubt, zog aber bei dem hier benutzten Rechner 50mA pro Bit, d.h. im Extremfall 400mA aus dem Ein- Ausgabechip. Da für die meisten Technikexperimente die 3 hochohmigen Eingänge DE1 - DE3 ausreichen verzichte ich im allgemeinen auf diese Art der Eingabe.

3.3 Programmierung der zusätzlichen 3-Bit Eingabe

Wie oben beschrieben, kann der Eingabewert durch Verbinden mit GND bzw. wie aus der Schaltskizze ersichtlich geändert werden. Die Basisadresse wird jeweils um 1 erhöht, d.h. 888+1 (632+1 oder 956+1).



Program Eingabe_2;

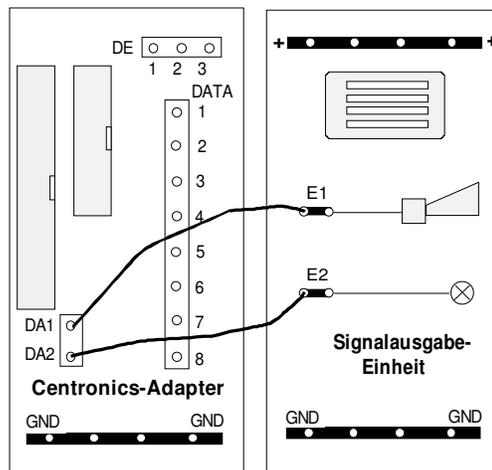
```
begin
    clrscr;
    writeln (Port[889]);
end.
```

Die ermittelten Werte sind rechnerabhängig und sollten durch Test ermittelt werden. Für den hier benutzten Rechner ergeben sich die weiter unten angeführten Werte, die sich natürlich auch ändern, wenn mehrere Eingänge gleichzeitig mit GND verbunden werden.

Rechnerspezifische Werte für Port[889]:

Eingänge	offen	DE1-GND	DE2-GND	DE3-GND
	127	111	95	63

3.4 Programmierung der zusätzlichen 2-Bit Ausgabe (Schaltungsbeispiel)



Die Basisadresse wird jeweils um 2 (888+2 = 890) erhöht.

Program Ausgabe_2;

```
begin
    clrscr;
    Port[890] := 4;
end.
```

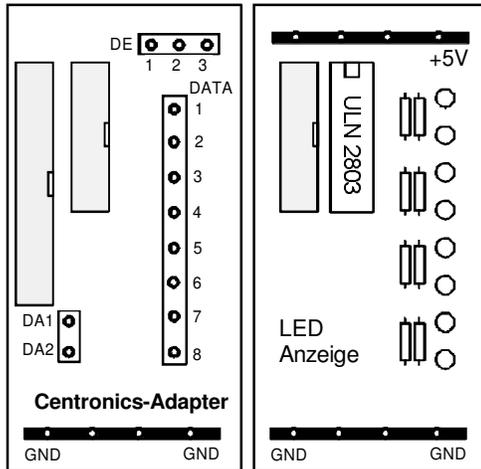
Leider ist der Zusammenhang zwischen der auszugebenden Dezimalzahl und der Aktivierung von DA1 und DA2 nicht einfach zu durchschauen, deshalb die folgende Tabelle:

Ausgabe	DA2	DA1
Port[890]:=1;	0	0
Port[890]:=0;	0	1
Port[890]:=5;	1	0
Port[890]:=4;	1	1

Achtung, soll an die Schnittstelle wieder ein Drucker angeschlossen werden, dann muß abschließend der Befehl Port[890] := 4 ausgeführt bzw. durch ein Reset die Schnittstelle wieder initialisiert werden.

4. Anwendungs-Software

4.1 Lauflicht



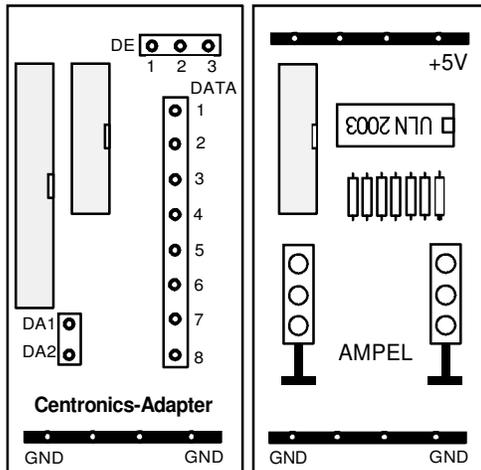
PROGRAM Lauflicht;

```

const a=200; ioaus=888;
begin
  clrscr;
  writeln('Lauflicht');
  repeat
    port [ioaus] := 1; delay (a);
    port [ioaus] := 2; delay (a);
    port [ioaus] := 4; delay (a);
    port [ioaus] := 8; delay (a);
    port [ioaus] := 16; delay (a);
    port [ioaus] := 32; delay (a);
    port [ioaus] := 64; delay (a);
    port [ioaus] :=128; delay (a);
  until keypressed;
  port[ioaus]:=0;
end.

```

4.2 Ampel



PROGRAM Ampel;

```

const ioaus=888;
begin
  clrscr;
  writeln('Ampelsteuerung');
  repeat
    port [ioaus] := 1; delay (3000); {rot}
    port [ioaus] := 3; delay (500); {rot-gelb}
    port [ioaus] := 4; delay (2000); {grün}
    port [ioaus] := 2; delay (500); {gelb}
  until keypressed;
  port[ioaus]:=0;
end.

```

4.3 Gleichstrommotor

Das Programm bezieht sich auf die Schaltung 2.5.2 bzw. steuert zwei Relais (je ein Wechsler), mit deren Hilfe eine Wendepolschaltung realisiert ist.

Program Gleichstrommotor;

```
var taste : char;

begin
  clrscr;
  writeln ('Taste r      betätigen für Rechtsdrehung');
  writeln ('Taste l      betätigen für Linksdrehung ');
  writeln ('Taste Space  betätigen für Stop      ');
  writeln ('Taste ESC    betätigen für ENDE      ');
  repeat
    read (kbd,taste);
    case taste of
      'r' : Port[888] := 1;
      'l' : Port[888] := 2;
      #32 : Port[888] := 0;
    end;
  until Taste = #27;
end.
```

4.4 Schrittmotor

Program Schrittmotor;

```
const ioaus = 888; a = 100;
var taste : char;

procedure links_drehen;
begin
  gotoxy(35,18); write ('Linksdrehung ');
  repeat
    port[ioaus]:= 6; delay(a);
    port[ioaus]:= 5; delay(a);
    port[ioaus]:= 9; delay(a);
    port[ioaus]:=10; delay(a);
  until keypressed;
  port[ioaus]:=0;
end;

procedure rechts_drehen;
begin
  gotoxy(35,18); write ('Rechtsdrehung ');
  repeat
    port[ioaus]:=10; delay(a);
    port[ioaus]:= 9; delay(a);
    port[ioaus]:= 5; delay(a);
    port[ioaus]:= 6; delay(a);
  until keypressed;
  port[ioaus]:=0;
end;
```

{ Hauptprogramm }

```
begin
  clrscr;
  gotoxy(10,11); write (' <r>  rechts-drehen');
  gotoxy(10,12); write (' <l>  links-drehen ');
  gotoxy(10,13); write ('<TASTE> Stop ');
  gotoxy(10,14); write (' <ESC> ENDE ');
  repeat
    read(kbd,taste);
    case taste of
      'r' : rechts_drehen;
      'l' : links_drehen ;
    else begin
      gotoxy(35,18);
      write (' Stop ');
      end;
    end; {case}
  until taste = #27;
end.
```

4.5 Programmierung des Schrittmotor-Decoders

Program Schrittmotor_mit_Decoder;

```
const ioaus = 888; a = 100;
```

```
var taste : char;
```

```
procedure links_drehen;
```

```
begin
  gotoxy(35,18); write (' links-drehen ');
  repeat
    port[ioaus]:=1; delay(a);
    port[ioaus]:=3; delay(a); { Takt auf DATA2 }
  until keypressed;
  port[ioaus]:=0;
  gotoxy(35,18); write (' ');
end;
```

```
procedure rechts_drehen;
```

```
begin
  gotoxy(35,18); write (' rechts-drehen ');
  repeat
    port[ioaus]:=2; delay(a);
    port[ioaus]:=3; delay(a); { Takt auf DATA1 }
  until keypressed;
  port[ioaus]:=0;
  gotoxy(35,18); write (' ');
end;
```

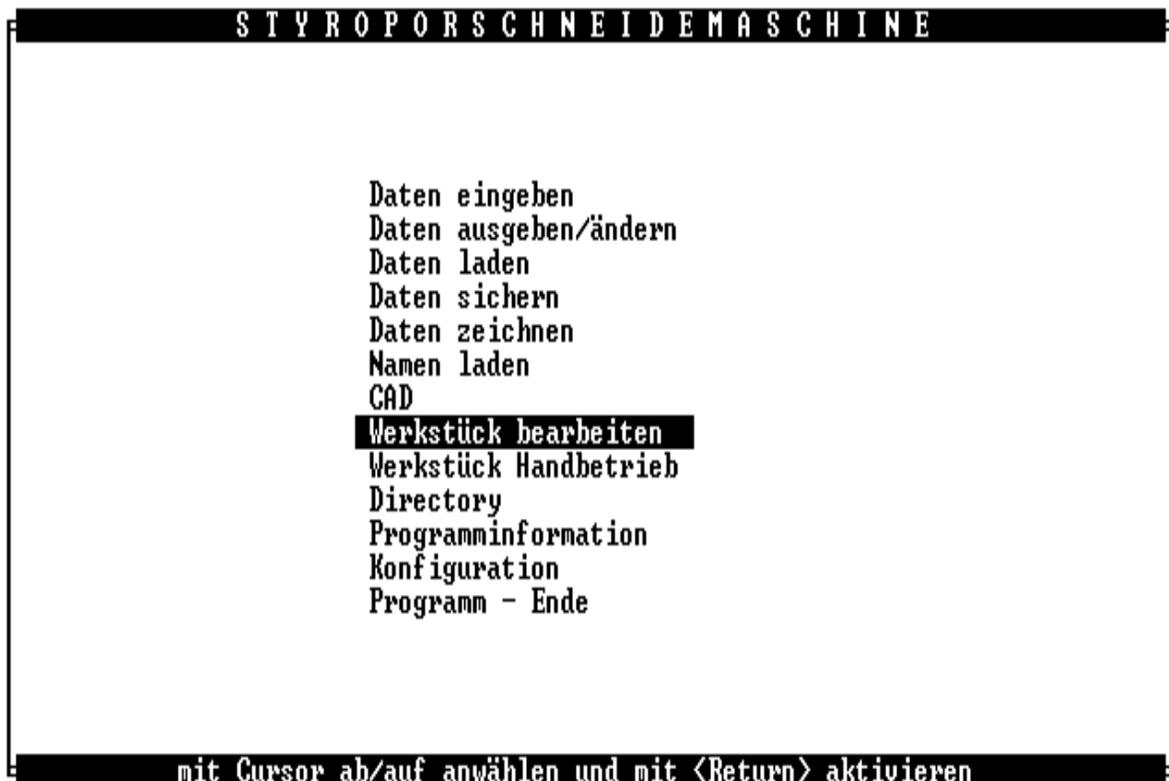
{ Hauptprogramm }

```
begin
  clrscr;
  gotoxy(10,11); write (' < r >      rechts-drehen');
  gotoxy(10,12); write (' < l >      links-drehen');
  gotoxy(10,13); write (' < TASTE > Stop ');
  gotoxy(10,14); write (' < ESC >   ENDE ');
  repeat
    read(kbd,taste);
    case taste of
      'r' : rechts_drehen;
      'l' : links_drehen ;
    end; {case}
  until taste = #27;
end.
```

4.6 Styroporschneidemaschine (CAD-CAM System)

Das mit der Druckerschnittstelle auch relativ komfortable Steuerungsaufgaben zu bewältigen sind, soll das nachfolgend beschriebene Programm verdeutlichen. Das Programm ist für Schulzwecke geschrieben und steuert eine Styroporschneidemaschine über die Druckerschnittstelle. Die elektronische Ansteuerung ist unter 2.5.5 beschrieben, die von mir benutzte Maschine ist ein Nachbau der von der Fa. Learnware Computer-Lehrmittel vertriebenen Maschine (siehe Pkt. 6). Das unten abgebildete Hauptmenü des Programms läßt die Optionen erkennen, einige seien kurz erläutert.

Bildschirmausdruck des Menüs:



Daten eingeben: Befehlsfolge zur Steuerung der Maschine z.B.

1. r - 12.5 rechts 12.5mm
2. o - 30 oben 30mm
3. r - 50 rechts 50mm
4. # Ende der Dateneingabe

Namen laden:

Unter diesem Menüpunkt können aus einem vorprogrammierten Alphabet Buchstaben zu einem Namen zusammenkopiert und gefertigt werden (siehe Bildschirmausdruck).

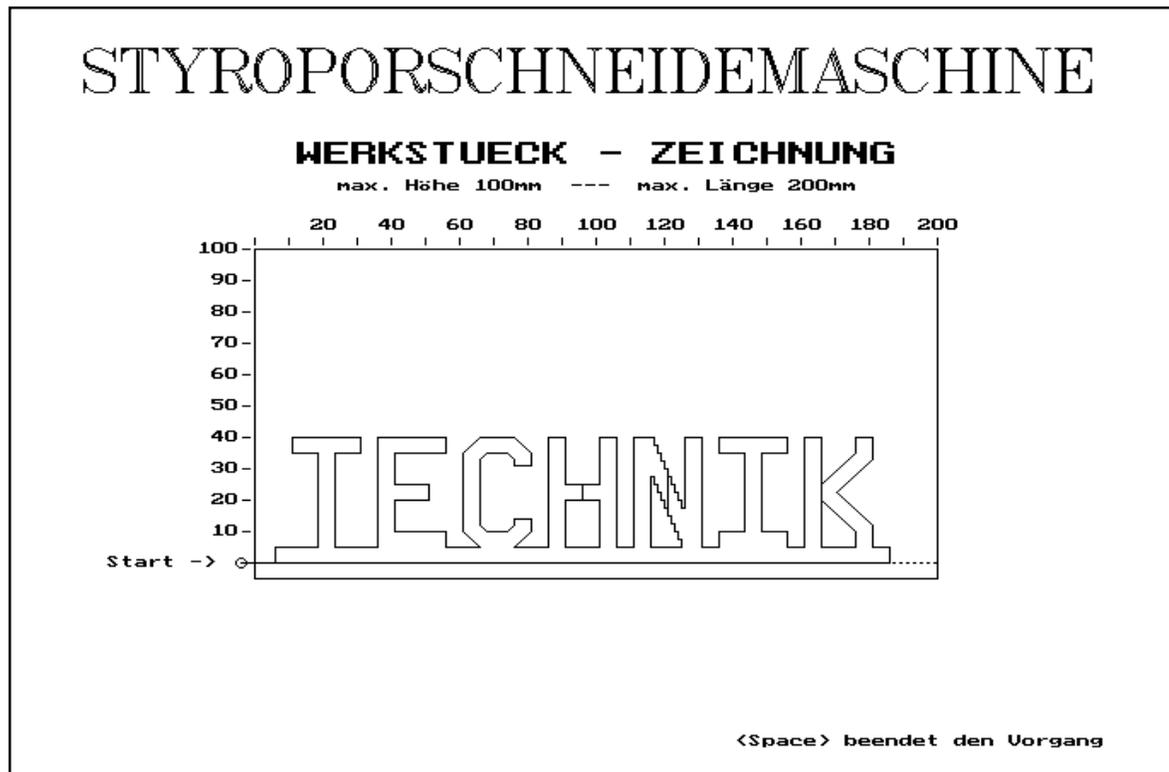
CAD :

Einfaches Zeichen-Programm per Tastatur, gleichzeitig wird die Befehlsfolge wie unter Daten eingeben festgehalten.

Werkstück bearbeiten:

Fertigungsvorgang, die eingegebene Befehlsfolge wird nacheinander abgearbeitet. Der Fertigungsverlauf kann parallel am Bildschirm verfolgt werden. Auch ohne Maschine (Simulation) wird der Fertigungsverlauf auf dem Bildschirm angezeigt, der zeitliche Verlauf entspricht dem des Modells. Das Werkstück kann bei gleichem Datensatz in vier verschiedenen Größen gefertigt werden.

Bildschirmausdruck "Werkstück bearbeiten" mit dem Beispiel einer Namensfertigung.



5. Platinenlayouts

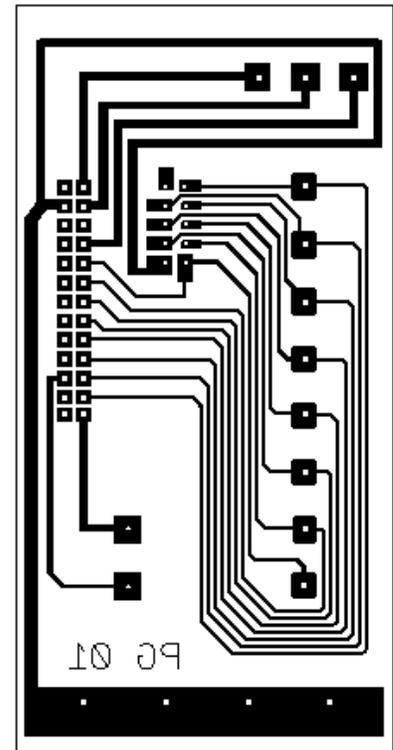
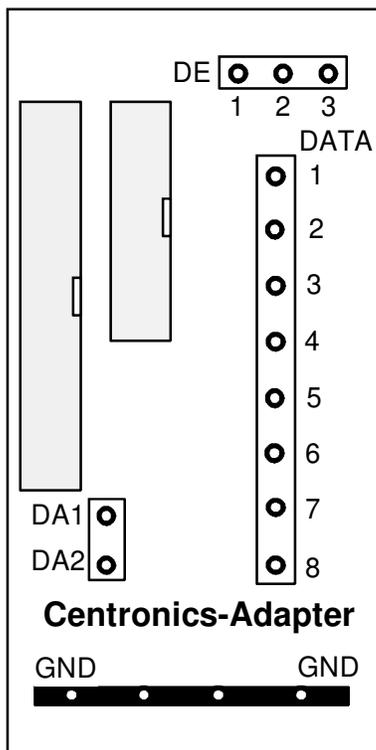
Die Layouts werden jeweils so übertragen, daß die Schrift auf der Lötseite lesbar bleibt.

5.1 Adapter für die Druckerschnittstelle

Die Platine hat die Abmaße 50*100mm².

Neben der Platine sind noch die folgenden Bauteile erforderlich:

1. 26-polige Stiftleiste
2. 10-polige Stiftleiste
3. 17 Lötnägel 1,3mm

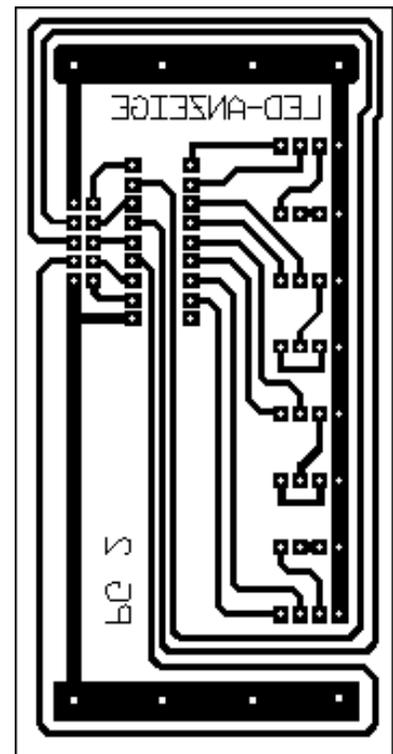
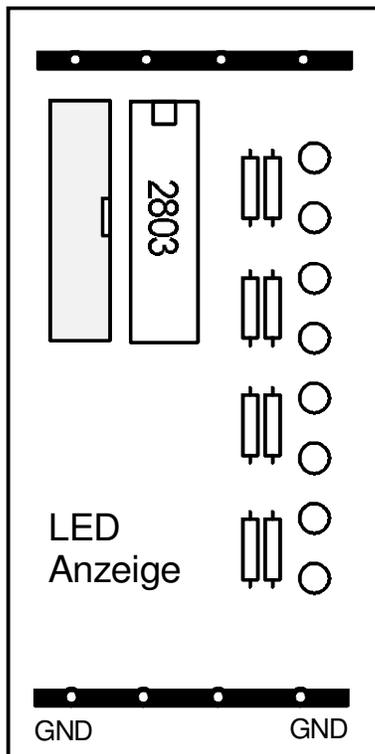


5.2 8 Bit Led-Anzeige

Die Platine hat die Abmaße 50*100mm².

Neben der Platine sind noch die folgenden Bauteile erforderlich:

1. 10-polige Stiftleiste
2. 8 Löt Nägel 1,3mm
3. 8 Widerstände 220Ω
4. 8 Leuchtdioden
5. IC-Fassung 18PINs
5. 8-fach Treiber IC ULN2803



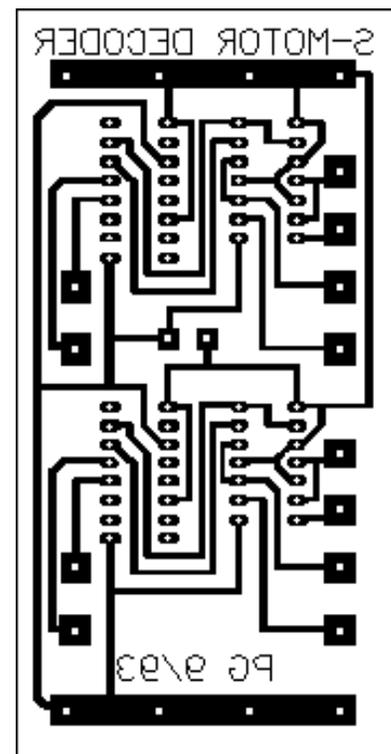
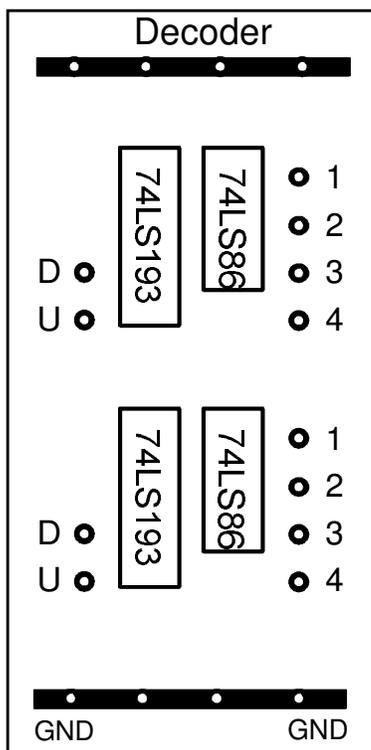
5.3 Decoder für Schrittmotore

Die Platine ermöglicht ein sehr einfaches Ansteuern von 2 Schrittmotoren. An den Eingängen D oder U wird ein Taktsignal angelegt, das Bitmuster für die Schrittmotoren wird auf der Platine gebildet.

Die Platine hat die Abmaße 50*100mm².

Neben der Platine sind noch die folgenden Bauteile erforderlich:

1. 20 Lötnägel 1,3mm
2. 1 Elko 100 uF
3. 14 PIN IC-Fassung 2mal
4. 16 PIN IC-Fassung 2mal
5. 2 ICs 74LS86
6. 2 ICs 74LS193

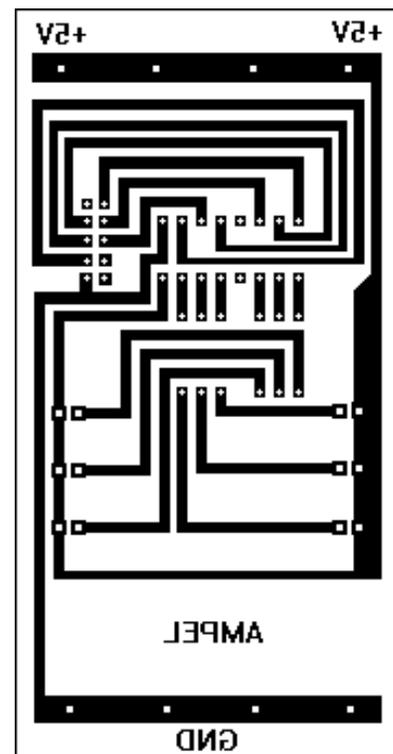
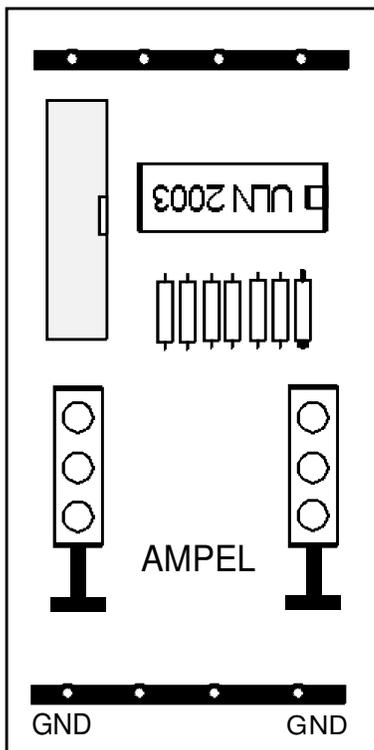


5.4 2 LED Ampeln

Die Platine hat die Abmaße 50*100mm².

Neben der Platine sind noch die folgenden Bauteile erforderlich:

1. 10-polige Stiftleiste
2. 8 Lötnägel 1,3mm
3. 6 Widerstände 220Ω
4. 2 rote, 2 gelbe und 2 grüne LEDs
5. 16PIN IC-Fassung
6. 7-fach Treiber IC ULN2003

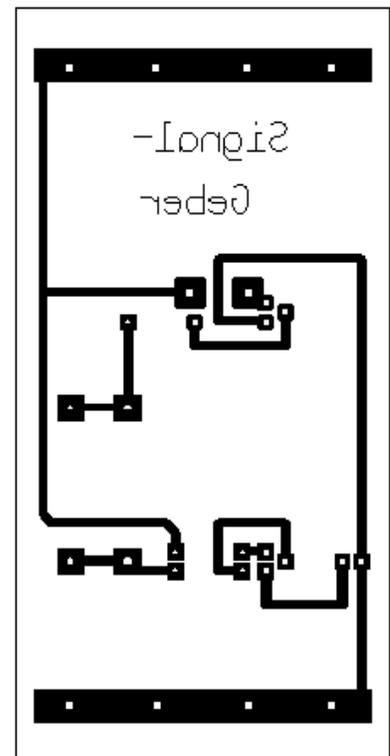
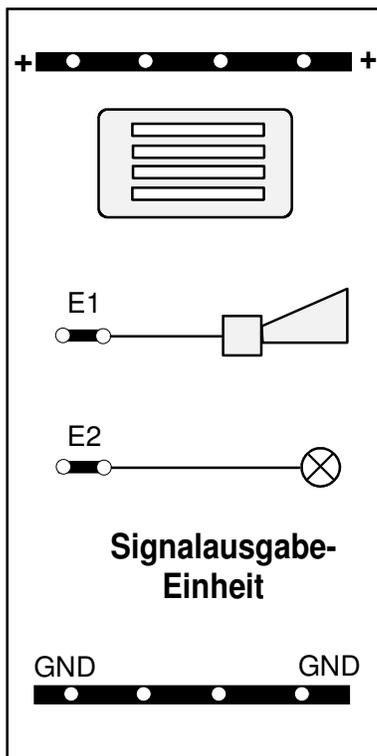


5.5 Signalausgabe

Die Platine hat die Abmaße 50*100mm².

Neben der Platine sind noch die folgenden Bauteile erforderlich:

1. 12 Lötnägel 1,3mm
2. Summer
3. 2 Transistoren z.B. BC238
4. 2 Widerstände 3,3 k Ω
5. 1 Widerstand 220 Ω
5. 1 LED



6. Beschaffungshilfe Hardware

Die Platinen können evtl. nach Absprache bei der

Technik-AG P.Grimm
Willy-Brandt-Gesamtschule Castrop-Rauxel
Bahnhofstr. 160
44575 Castrop-Rauxel
Tel.:02305-106-2607

bestellt werden.

Für 25 DM kann bei der Technik-AG eine Diskette bestellt werden, auf der erprobte Programme für den Technikunterricht enthalten sind.

(Die Diskette enthält u.a. ein umfangreiches Testprogramm für die Schnittstelle und ein CAD-CAM Programm zur Ansteuerung einer Styroporschneidemaschine siehe unter 4.6)

Eine Styroporschneidemaschine kann bei der

Learnware Computer-Lehrmittel GmbH
Liebigstr. 5
44139 Dortmund
Tel.:0231-104163

bestellt werden.

7. Literaturliste, Quellenverzeichnis:

1. Schmidt/Weber Messen und Experimentieren, IBM und Kompatible
Ferd. Dümmler^s Verlag
Bonn 1989
Tel.:0228 / 22 30 31
2. Pütz, Jan (Hrsg.) Einführung in die Mikroprozessor Anwendung
vgs Verlagsgesellschaft
Köln 1987. Seite 134 u. folgende
3. Kindt, Volker Funkschau 2187, Seite 61 162
4. Texas Instruments (Hrsg.) Pocket Guide,
Lineare Integrierte Schaltungen Bd 2
Freising 1985
5. IWF (Hrsg.) TTL Taschenbuch, Teil 1
Vaterstetten bei München 1987
6. ESTU (Hrsg.) Niebur Gesellschaft mbH
Postfach 200421
45633 Recklinghausen
Tel.:(02361) 24346