

Workshop

Information & Kommunikation

Eingangssicherung mit Computer

Peter Grimm

Heft 2/99

Herausgeber:

Technik - Unterricht: Forum eV

Verband der Techniklehrer für die gymnasiale Oberstufe in NRW

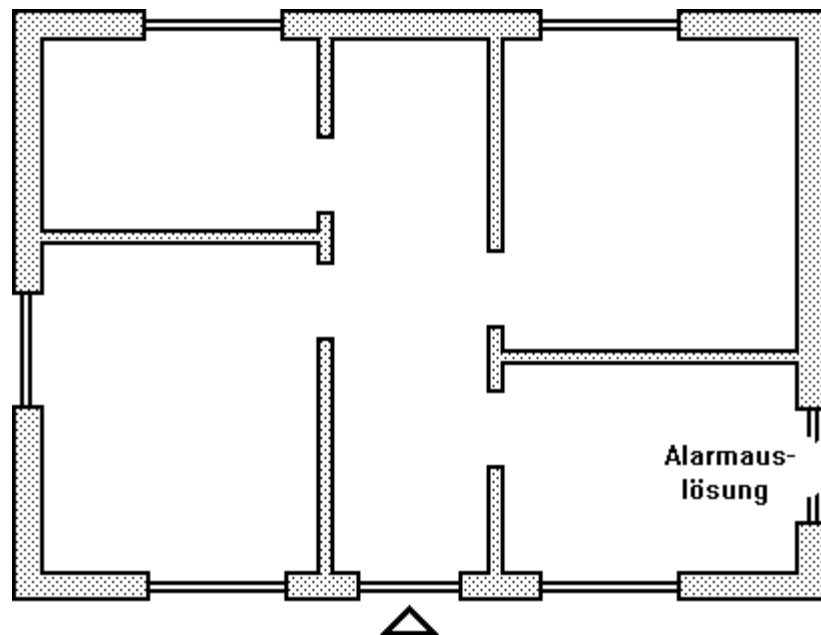
Memelstraße 75 · 47057 Duisburg · Tel/Fax 0203 - 354992

http://www.technik-und-unterricht.de/ email: tuf@technik-und-unterricht.de

TECHNIK

Haussicherungstechnik

Teil 2, Computergesteuerte Alarmzentrale



Allgemeine Hinweise

Der Computer erlaubt, verschiedene Teilsysteme (Sensoren) miteinander zu verbinden und zu einer komplexen Gesamtlösung des Problems Zugangssicherheit und Alarmüberwachung zu vernetzen.

Im Folgenden sollen erste Hinweise für eine unterrichtliche Behandlung des Themas gegeben werden. Wegen des Themenumfanges werden bewußt Beschränkungen vorgenommen.

Alternativ zur hier vorgestellten Centronics-Schnittstelle und dem I2C-Bus und den Programmierhinweisen in Turbo-Pascal 4.0 können alle Programmiersprachen (z.B. Basic, Logo), die ein Ansprechen der Schnittstellen erlauben, benutzt werden. Natürlich sind auch noch weitere Systeme z.B. Oktobus, Fischertechnik, Modulbus u.a. einsetzbar.

Wichtig:

Die bisher erzielten Teillösungen müssen hinsichtlich der Bereitstellung von 0-1 Pegeln untersucht und wenn noch nicht geschehen, überarbeitet werden.

Da die DE1..3 Eingänge der Centronics-Schnittstelle im unbeschalteten Zustand 1-Signal aufweisen und das "0-Signal" durch Verbinden mit GND erzielt wird, läßt sich eine potentialfreie Lösung durch Einbau von Relais erzielen (siehe Skizzen rechts).

Kurzdarstellung der Centronicsschnittstelle

Die hier benutzten Bezeichnungen und Strom: auf das im Dümmlerverlag herausgegebene B Weber, Messen und Experimentieren, IBM u.l 427-42362-X). In Kurzform die wichtigsten Hin

[A] DATA 1..8 Datenaus- und -eingänge

Im allgemeinen nur als Datenausgang t
blastbarkeit liegt bei $I < 100 \text{ mA}$ je Ausg
sind kurzschlußfest. Softwaremäßig wi
durch OUT 888,0..255 bzw. PRINT INP
PORT[888]:=0..255 bzw. WRITE(POR
angesprochen.

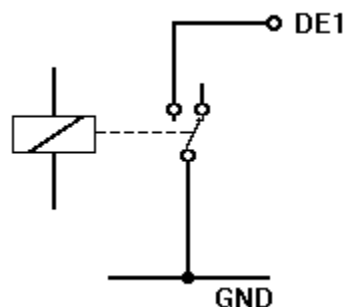
[B] DE 1..3 Dateneingänge

DE1..3 Verbinden der Eingänge mit GND bzw. $R < \text{ca. } 6\text{k}\Omega$. Die Abfrage erfolgt über PRINT INP(888+1) in Basic bzw. WRITE (PORT[888+1]); in Pascal

[C] DA 1, 2 Datenausgänge

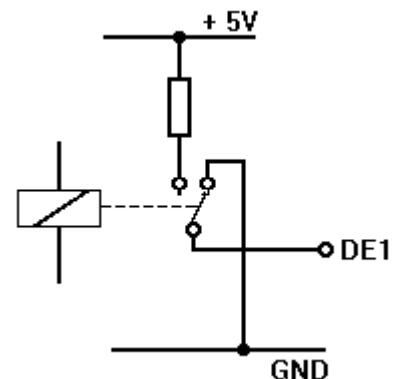
DA1,DA2 gehören zum Kontrollregister der Centronics-Schnittstelle. Die Ausgänge sind kurzschlußfest, der Strom wird auf ca. 0,4-0,8mA begrenzt. Angesprochen werden die Ausgänge über OUT 888+2,X bzw. PORT[888+2]:=X über OUT 888+2,X bzw. PORT[888+2]:=X.

Die Basis-Adresse des LPT1-Ausgangs ist 888, weitere Centronics-Schnittstellen haben die Basis-Adressen 632 bzw. 956.



Version-1

Version-2

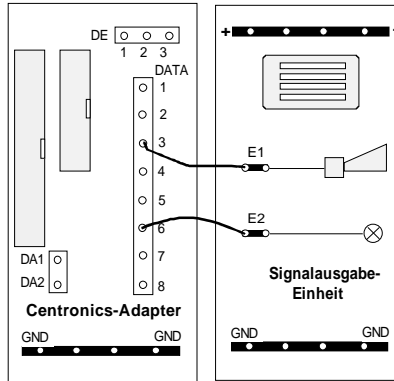


Handhabung der Schnittstelle bezüglich der Alarmsensor-Erfassung

Achtung: Bei den folgenden Experimenten wird davon ausgegangen, daß die benutzten Platinen mit einer Stromquelle von 5V versorgt werden. Die entsprechenden Verbindungen sind nicht eingezeichnet. In den Skizzen sind nur die Signalleitungen eingezeichnet.

Aufgabe-1

Schließe die Signalausgabeeinheit an die DATA-Ausgänge DATA 3 und 6 an und schalte die Aktoren per Computer ein und aus.



Ergebnisse-1:

.....

.....

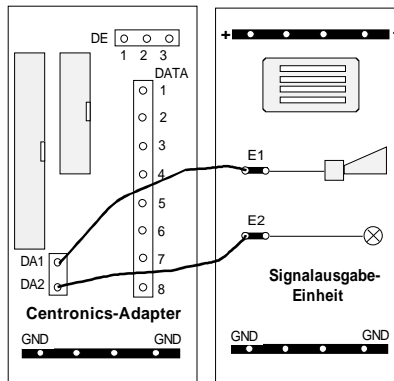
.....

.....

.....

Aufgabe-2

Schließe die Signalausgabeeinheit an die Daten-Ausgänge DA 1 und 2 an und schalte die Aktoren per Computer ein und aus.



Ergebnisse-2:

.....

.....

.....

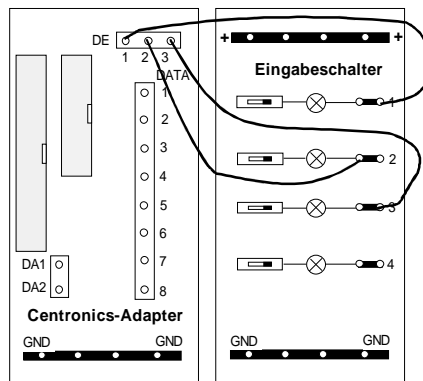
.....

.....

Aufgabe-3

Schließe die Eingabeschalter an die Daten-Eingänge DE 1..3 an und ermittle die Registerinhalte.

(Bei fehlender Eingabeeinheit können die Werte auch durch direktes Verbinden der Eingänge mit GND ermittelt werden.)



Ergebnisse-3:

.....

.....

.....

.....

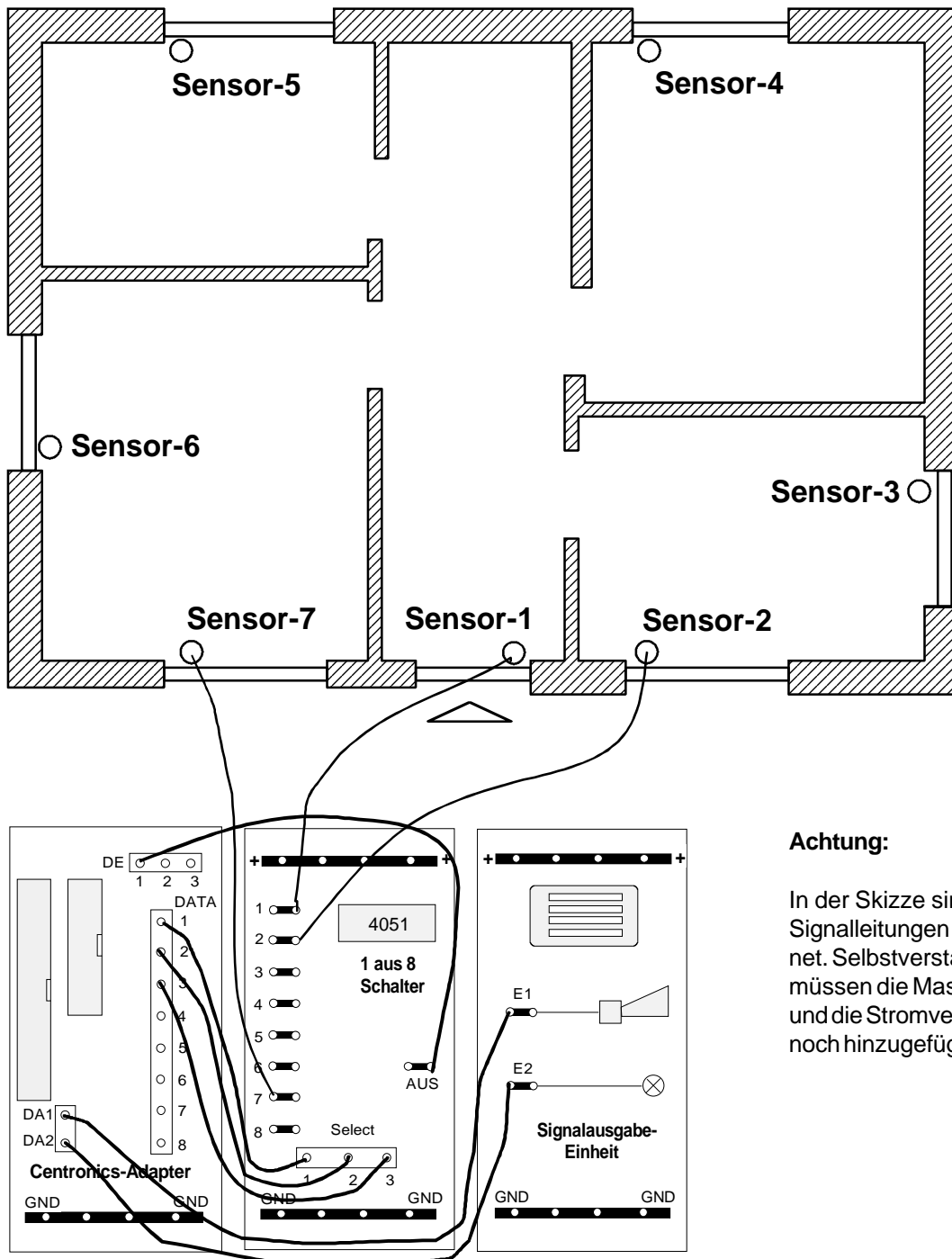
.....

Aufgabe-4

Verbinde Aufg. 1 u. 3 so miteinander, daß ein Alarm über die Schalter ausgelöst wird und der akustische Melder den Alarm für eine voreingestellte Zeit ausgibt. Die Meldelampe soll den Alarmfall durch Blinken anzeigen, das Zurücksetzen erfolgt durch die Taste Q.

2.6.1.1 Centronics

Die Skizze zeigt ein mögliches Schaltungsbeispiel für die Abfrage von 8-Sensoren.



Achtung:

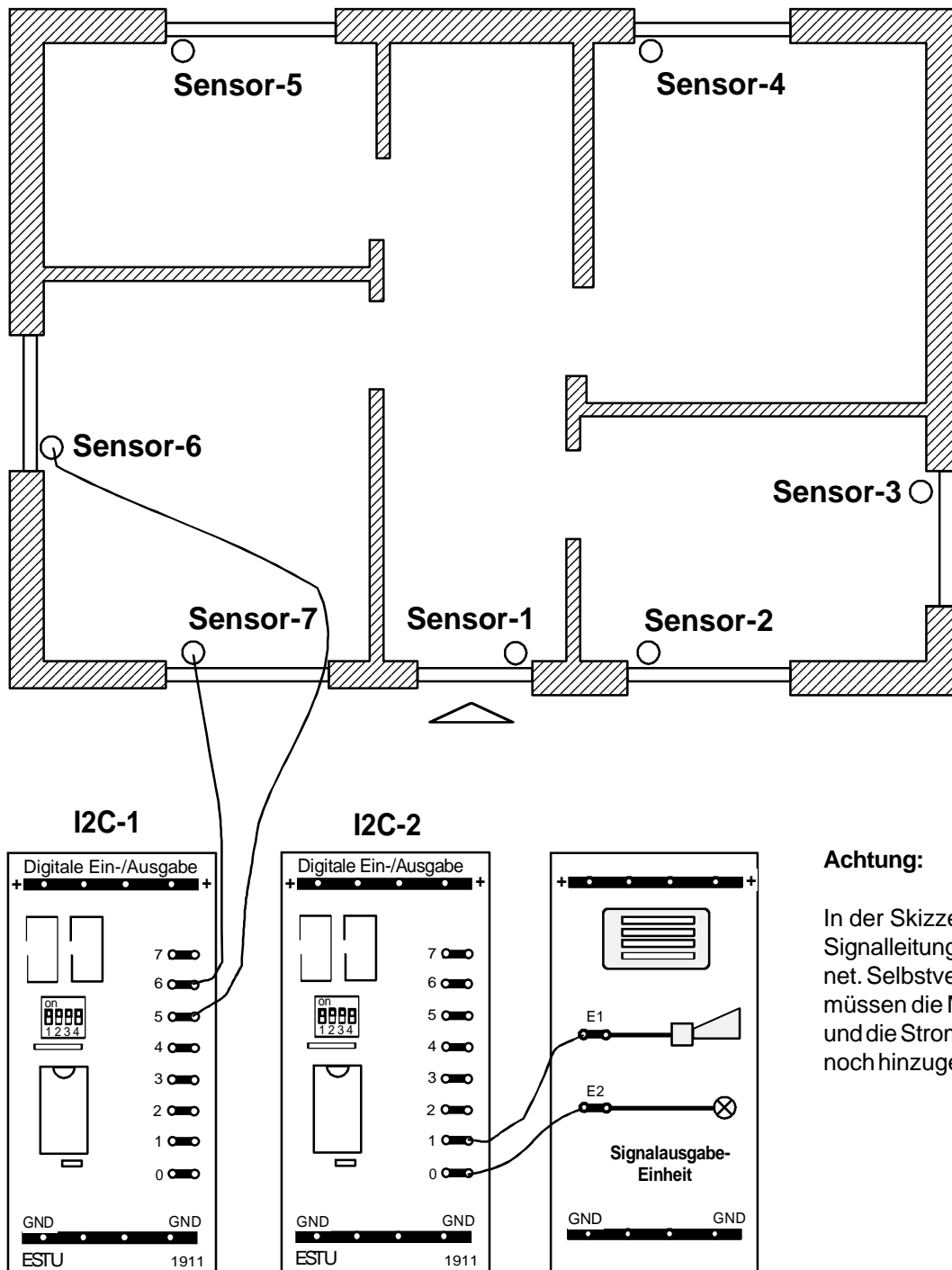
In der Skizze sind nur die Signalleitungen eingezeichnet. Selbstverständlich müssen die Masseleitungen und die Stromversorgung noch hinzugefügt werden.

Wieviele Sensoren lassen sich bei Verwendung mehrerer 1-aus-8 Schalter-Platinen mit der vorgestellten Technik abfragen?

Lösung: _____

2.6.1.2 I2C-Bus

Die Skizze zeigt ein mögliches Schaltungsbeispiel für die Abfrage von 8-Sensoren.



Achtung:

In der Skizze sind nur die Signalleitungen eingezeichnet. Selbstverständlich müssen die Masseleitungen und die Stromversorgung noch hinzugefügt werden.

Wieviele I2C Platinen des Typs digitale Ein-/Ausgabe lassen sich max. an den seriellen Bus anschließen?

Lösung: _____

Projektarbeit (Centronics oder I2C)

Erstelle ein benutzerfreundliches Programm, das mehrere Sensoren erfaßt und eine optisch ansprechende Bildschirmausgabe enthält.

Mögliche Aufgabenverteilungen könnten sein:

- a) Produktbeschreibung
Benutzungshinweise, technische Beschreibung
- b) Einkauf, Verkauf
Ermittlung der Anlagekosten, Verkaufspreis, Werbematerial
- c) Bildschirmausgabemaske (ASCII bzw. grafisch)
- d) Menüstruktur, Vergabe von Prozedur-, Funktionsnamen
- e) Erfassen des Alarmfalls
Zeit-, Datumserfassung und Abspeichern der Ereignisse
Erfassen der Einschalt- und Quitierungsvorgänge,
Erfassen der Aktivierung der Alarmschaltung

Weitere Aufgaben könnten sein:

Zeitgesteuerte Außenlichtschaltung;

Rolladensteuerung;

Heizungssteuerung, Temperaturüberwachung;

Allgemeines

Die verwendete Programmiersprache ist Turbo-Pascal 4.0. Die Befehle der Unit Security werden dann korrekt ausgeführt, wenn die Hardware entsprechend Seite-4 bzw. 5 ausgeführt wird. Mit leichten Änderungen läßt sich die Software auch auf andere Problemstellungen übertragen.

Handhabung

Die Security.TPU kann unterrichtlich auf 2 Arten gehandhabt werden:

1. Die TPU wird als Erweiterung des Pascalbefehlssatzes ähnlich CRT genannt, die Befehle sukzessive bei Bedarf erläutert.
2. Die TPU wird offen behandelt, die Schüler können die TPU verändern bzw. benötigte Prozeduren in ihr Programm einbinden. Dies ist sicherlich der Weg für fortgeschrittene Programmierer.

Neue Befehle

uses secu_par;	Sammlung von Befehlen für die Alarmanlage Basisadresse 888 (LPT1) wurde eingestellt																		
uses secu_ser;	Sammlung von Befehlen für die Alarmanlage Basisadresse Com2 wurde eingestellt																		
function alarm (Nr:byte): boolean;	1..8 für 8 Sensoren sonst Hard- und Softwareerweiterung Alarm (Nr) liefert true bzw. false bei Aufruf																		
procedure hupe (n:byte);	0 = aus / 1 = ein schaltet die Hupe aus/ein																		
procedure Signal_Lampe (n:byte);	0 = aus / 1 = ein schaltet die Signal_Lampe aus/ein																		
function Datum : date;	liefert bei Aufruf das Datum mit einer max. Stringlänge von 27 Zeichen																		
function Uhrzeit : time;	liefert bei Aufruf die Uhrzeit mit einer max. Stringlänge von 12 Zeichen																		
procedure farbe(hinten,vorne:byte);	Mit Aufruf dieser Prozedur können verschiedene Bildschirm-effekte erzielt werden. Die erste Zahl setzt die Hinter-, die zweite Zahl die Vordergrundfarbe. Beispiele: <table border="0" style="margin-left: 40px;"> <tr> <td>0,7</td> <td>normal</td> <td>schwarz, hellgrau</td> </tr> <tr> <td>7,0</td> <td>revers</td> <td>hellgrau, schwarz</td> </tr> <tr> <td>0,12</td> <td>rot</td> <td>schwarz, rot</td> </tr> <tr> <td>0,14</td> <td>gelb</td> <td>schwarz, gelb</td> </tr> <tr> <td>0,28</td> <td></td> <td>schwarz, rot-blinkend</td> </tr> <tr> <td>12,42</td> <td></td> <td>rot, gelb-blinkend</td> </tr> </table>	0,7	normal	schwarz, hellgrau	7,0	revers	hellgrau, schwarz	0,12	rot	schwarz, rot	0,14	gelb	schwarz, gelb	0,28		schwarz, rot-blinkend	12,42		rot, gelb-blinkend
0,7	normal	schwarz, hellgrau																	
7,0	revers	hellgrau, schwarz																	
0,12	rot	schwarz, rot																	
0,14	gelb	schwarz, gelb																	
0,28		schwarz, rot-blinkend																	
12,42		rot, gelb-blinkend																	
procedure ton;	gibt einen kurzen Ton über den Systemlautsprecher aus																		
procedure Sichere_Alarm (Nr:byte);	Sichert die Aktivität (True oder False) von Sensor Nr. 1..8 mit Angabe von Datum und Uhrzeit ab, ruft selbständig die Funktion Alarm (Nr) auf.																		
procedure Lade_Alarmtext (Nr,x,y:byte);	Lädt den abgespeicherten Text von Sensor Nr. 1..8 mit Angabe von Datum und Uhrzeit und gibt die Informationen an der Position x,y aus.																		


```
unit secu_par;
```

```
interface
```

```
uses crt, dos;
```

```
const io = 888;
```

```
type date = string[27];
      time = string[12];
```

```
function Alarm (Nr:byte): boolean;
procedure hupe (n:byte); {0=aus / 1=ein}
procedure Signal_Lampe (n:byte); {0=aus / 1=ein}
function Datum : date;
function Uhrzeit : time;
```

```
procedure farbe (hinten, vorne :byte);
procedure ton;
```

```
procedure Sichere_Alarm (Nr:byte);
procedure Lade_Alarmtext(nr,x,y:byte);
```

```
implementation
```

```
var
  Std,min,sek,msek : word;
  Jahr,Monat,Tag,Wtag : word;
  Wochentag : String[12];
  Datei : text;
```

```
procedure farbe (hinten,vorne:byte);
```

```
begin
  textbackground (hinten);
  textcolor (vorne);
end;
```

```
procedure ton;
```

```
begin
  sound(200);
  delay(200);
  nosound;
end;
```

```
function alarm (Nr:byte): boolean;
```

```
begin
  case nr of
    1 : port[io]:=0;
    2 : port[io]:=1;
    3 : port[io]:=2;
    4 : port[io]:=3;
    5 : port[io]:=4;
    6 : port[io]:=5;
    7 : port[io]:=6;
    8 : port[io]:=7;
  end;
  if port[io+1] = 111 then alarm := true
  else alarm := false;
end;
```

```
procedure hupe (n:byte);
```

```
begin
  case n of
    1 : port[io+2]:=0;
    0 : port[io+2]:=1;
  end;
end;
```

```
procedure Signal_Lampe (n:byte);
```

```
begin
  case n of
    1 : port[io+2]:=5;
    0 : port[io+2]:=1;
  end;
end;
```

```
function Datum : date;
```

```
var sTag, SMonat: string[2];
    sJahr : string[4];
begin
  getdate(Jahr,Monat,Tag,Wtag);
  case Wtag of
    0 : Wochentag:='Sonntag, ';
    1 : Wochentag:='Montag, ';
    2 : Wochentag:='Dienstag, ';
    3 : Wochentag:='Mittwoch, ';
    4 : Wochentag:='Donnerstag, ';
    5 : Wochentag:='Freitag, ';
    6 : Wochentag:='Samstag, ';
  end;
  str(Tag:2,sTag);
  str(Monat:2,sMonat);
  str(Jahr:4,sJahr);
  Datum := Wochentag +'den ' +sTag +'.'
          +sMonat +'.' +sJahr;
end;
```

```
function Uhrzeit : time;
```

```
var
  sStd, sMin, ssek : string[2];
begin
  gettime(Std,min,sek,msek);
  str(Std:2,sStd);
  str(Min:2,sMin);
  str(sek:2,ssek);
  Uhrzeit := sStd +'.' +sMin +'.' +ssek +' Uhr';
end;
```

```
procedure Sichere_Alarm (Nr:byte);
```

```
begin
  assign (datei,'Security.nr'+chr(nr+48));
  rewrite (datei);
  writeln (datei,Datum);
  writeln (datei,Uhrzeit);
  writeln (datei,'Sensor ',Nr,' = ',alarm(Nr));
  close (datei);
end;
```

```

procedure Lade_Alarmtext(nr,x,y:byte);
var   xDatum  : string[26];
      xUhrzeit : string[12];
      xSensor  : string[17];
begin
  assign (datei,'Security.nr'+chr(nr+48));
  {$I-}
  reset (datei);
  {$I+}
  if ioresult = 0 then
    begin
      readln (datei,xdatum);
      readln (datei,xuhrzeit);
      readln (datei,xsensor);
      close(datei);
      gotoxy(x,y ); write (xdatum);
      gotoxy(x,y+1); write (xUhrzeit);
      gotoxy(x,y+2); write (xSensor);
    end
  else
    begin
      gotoxy(x,y);
      write ('Datei nicht vorhanden');
    end;
end;

{----- Initialisierung -----}

begin
  port[io+2]:=1;   { Hupe und Lampe 0 setzen }
end.

```

Program Teste_Security;

```

uses crt, secu_par;

var   i       : byte;
      k       : byte;
      Taste   : char;
      SensorNr : byte;

begin
  k:=4;          { Alarmsensor 4 abfragen }
  clrscr;
  farbe(7,0);
  gotoxy(20,5);
  write (' Auslösung durch Sensor-',k );
  repeat until alarm(k) or keypressed;
  farbe(0,7);
  gotoxy(20,7); write ('Datum : ',Datum);
  gotoxy(20,8); write ('Zeit : ',Uhrzeit);
  if alarm(k) then begin
    sichere_alarm(k);
    farbe (0,7);
    gotoxy (20,11);
    write ('Sensor ',k,' ',alarm (4));
    hupe(1);
    delay(200);
    hupe(0);
    farbe(0,7);
    for i:=1 to 15 do begin
      signal_lampe(1);
      delay(200);
      signal_lampe(0);
      delay(200);
    end;
  end;
  lade_alarmtext(k,20,14);
  taste:=readkey;
end.

```

unit secu_ser;

```
{
Für die vorgestellte Unterrichtseinheit werden
zwei Ein-, Ausgabeplatinen des Typs Estu 1911 an
COM2 verwendet. Platine-1 mit der Sekundär-
adresse 0 wird für die Abfrage der Sensoren ver-
wendet, Platine-2 dient der Signalausgabe. Die
Signallampe wird bei der 2. Platine an PIN1, die
Hupe an PIN2 angeschlossen.
}
```

interface

```
uses crt, dos, i2c;
```

```
type date = string[27];
time = string[12];
```

```
function alarm (Nr:byte): boolean;
procedure hupe (n:byte); {0=aus / 1=ein}
procedure Signal_Lampe (n:byte); {0=aus / 1=ein}
function Datum : date;
function Uhrzeit : time;
```

```
procedure farbe(hinten,vorne:byte);
procedure ton;
```

```
procedure Sichere_Alarm (Nr:byte);
procedure Lade_Alarmtext(nr,x,y:byte);
```

implementation

```
var
Std,min,sek,msek : word;
Jahr,Monat,Tag,Wtag : word;
Wochentag : String[12];
Datei : text;
```

```
procedure farbe (hinten,vorne:byte);
begin
textbackground(hinten);
textcolor(vorne);
end;
```

```
procedure ton;
begin
sound(200);
delay(200);
nosound;
end;
```

```
function alarm (Nr:byte): boolean;
var zahl : byte;
begin
i2c_selectdevice(7,0);
i2c_read(zahl);
case nr of
1 : if zahl=254 then alarm:=true else alarm :=false;
2 : if zahl=253 then alarm:=true else alarm :=false;
3 : if zahl=251 then alarm:=true else alarm :=false;
4 : if zahl=247 then alarm:=true else alarm :=false;
5 : if zahl=239 then alarm:=true else alarm :=false;
6 : if zahl=223 then alarm:=true else alarm :=false;
7 : if zahl=191 then alarm:=true else alarm :=false;
8 : if zahl=127 then alarm:=true else alarm :=false;
end;
end;
```

```
procedure hupe (n:byte); {0=aus / 1=ein}
begin
i2c_selectdevice (7,1);
case n of
1 : i2c_write (4);
0 : i2c_write (0);
end;
end;
```

```
procedure Signal_Lampe (n:byte); {0=aus / 1=ein}
begin
i2c_selectdevice (7,1);
case n of
1 : i2c_write (2);
0 : i2c_write (0);
end;
end;
```

```
function Datum : date;
var sTag, SMonat: string[2];
sJahr : string[4];
begin
getdate(Jahr,Monat,Tag,Wtag);
case Wtag of
0 : Wochentag:='Sonntag, ';;
1 : Wochentag:='Montag, ';;
2 : Wochentag:='Dienstag, ';;
3 : Wochentag:='Mittwoch, ';;
4 : Wochentag:='Donnerstag, ';;
5 : Wochentag:='Freitag, ';;
6 : Wochentag:='Samstag, ';;
end; {case}
str(Tag: 2,sTag); str(Monat: 2,sMonat);
str(Jahr: 4,sJahr);
Datum := Wochentag +'den ' +sTag +'.'
+sMonat +'.' +sJahr;
end;
```

```

function Uhrzeit : time;
var
  sStd, sMin, ssek : string[2];
begin
  gettime(Std,min,sek,msek);
  str(Std: 2,sStd); str(Min: 2,sMin); str(sek :
2,ssek);
  Uhrzeit := sStd + ':' + sMin + ':' + ssek + ' Uhr';
end;

procedure Sichere_Alarm (Nr:byte);
begin
  assign (datei,'Security.nr' +chr(nr+48));
  rewrite (datei);
  writeln (datei,Datum);
  writeln (datei,Uhrzeit);
  writeln (datei,'Sensor ',Nr,' = ',alarm(Nr));
  close (datei);
end;

procedure Lade_Alarmtext(nr,x,y:byte);
var xDatum : string[26];
    xUhrzeit : string[12];
    xSensor : string[17];
begin
  assign (datei,'Security.nr'+chr(nr+48));
  {$I-}
  reset (datei);
  {$I+}
  if ioreult = 0 then begin
    readln (datei,xdatum);
    readln (datei,xuhrzeit);
    readln (datei,xsensor);
    close(datei);
    gotoxy(x,y ); write (xdatum);
    gotoxy(x,y+1); write (xUhrzeit);
    gotoxy(x,y+2); write (xSensor);
  end
  else
  begin
    gotoxy(x,y ); write ('Datei nicht
vorhanden');
  end;

end;

begin
  i2c_initport(2); {com2-Schnittstelle}
end.

```

Program Teste_Security;

```
uses crt, secu_ser;
```

```
var i : byte;
    k : byte;
    Taste : char;
    SensorNr : byte;
```

```
begin
  k:=4; { Alarmsensor 4 abfragen }
  clrscr;
  farbe(7,0);
  gotoxy(20,5);
  write (' Auslösung durch Sensor-',k );
  repeat until alarm(k) or keypressed;
  farbe(0,7);
  gotoxy(20,7); write ('Datum : ',Datum);
  gotoxy(20,8); write ('Zeit : ',Uhrzeit);
  if alarm(k) then begin
    sichere_alarm(k);
    farbe(0,7);
    gotoxy(20,11);
    write ('Sensor ',k,' ',alarm(4));
    hupe(1);
    delay(200);
    hupe(0);
    farbe(0,7);
    for i:=1 to 15 do begin
      signal_lampe(1);
      delay(200);
      signal_lampe(0);
      delay(200);
    end;
  end;
  lade_alarmtext(k,20,14);
  taste:=readkey;
end.

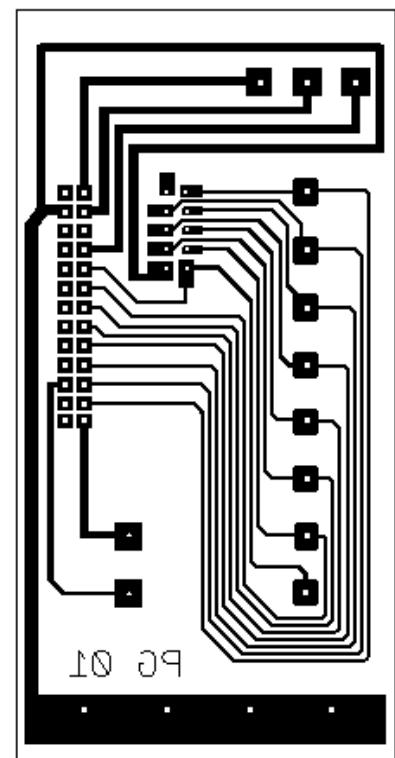
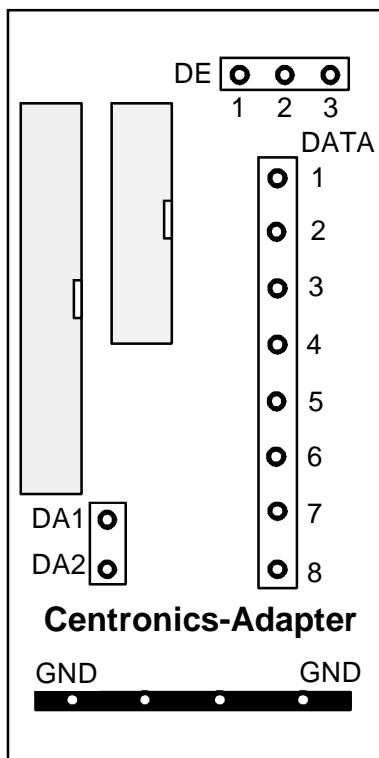
```

Adapter für die Druckerschnittstelle

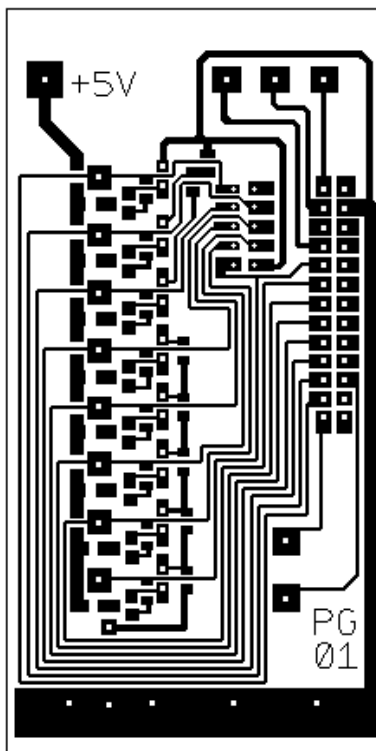
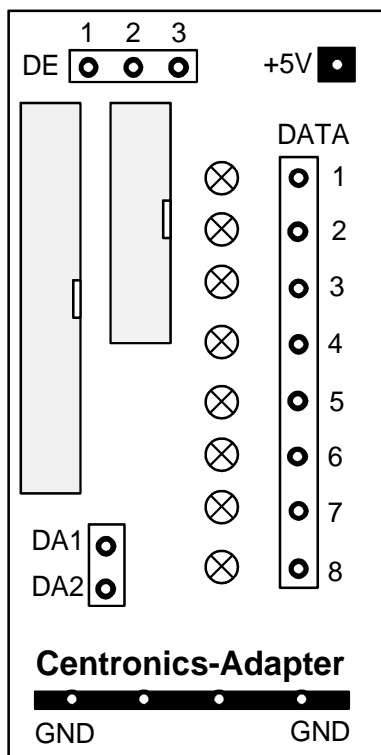
Die Platine hat die Abmaße 50*100mm².

Neben der Platine sind noch die folgenden Bauteile erforderlich:

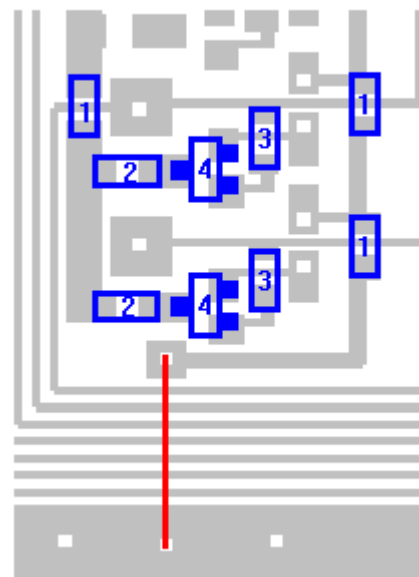
1. 26-polige Stiftleiste
2. 10-polige Stiftleiste
3. 17 Lötnägel 1,3 mm




Adapter für die Druckerschnittstelle, Entwurf mit integrierter LED-Anzeige in SMD-Technik



Detailzeichnung der Lötseite



Die Platine hat die Standardabmaße 50*100r c  E SMD Transistor
BC 848C im
SOT-23 Gehäuse

Neben der Platine sind noch die folgenden Bauteile erforderlich:

1. 26-polige Stiftleiste
2. 10-polige Stiftleiste
3. 18 Lötnägel 1,3 mm
4. 8 LED 3mm, I ca. 2mA

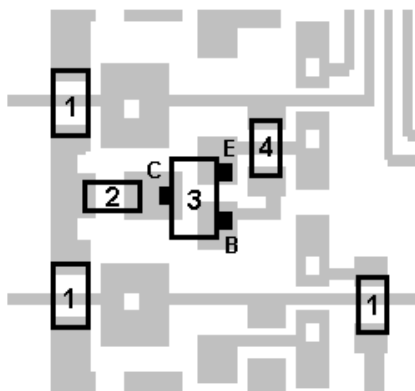
SMD-Bauteile

5. 14 0Ω Widerstände, Brückenwiderstände
6. 8 LED Vorwiderstand 1,5kΩ
7. 8 Basisvorwiderstand 47 kΩ
8. 8 Transistor BC 848C im SOT-23 Gehäuse

In der Detailzeichnung

- Bauteil Nr. 1
Bauteil Nr. 2
Bauteil Nr. 3
Bauteil Nr. 4

Detailzeichnung der Lötseite

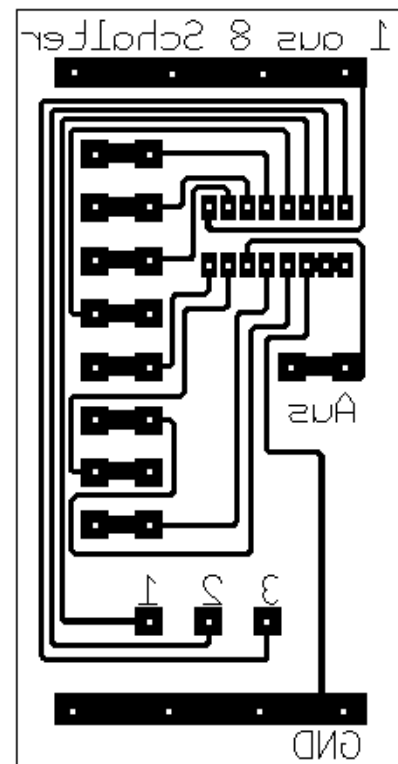
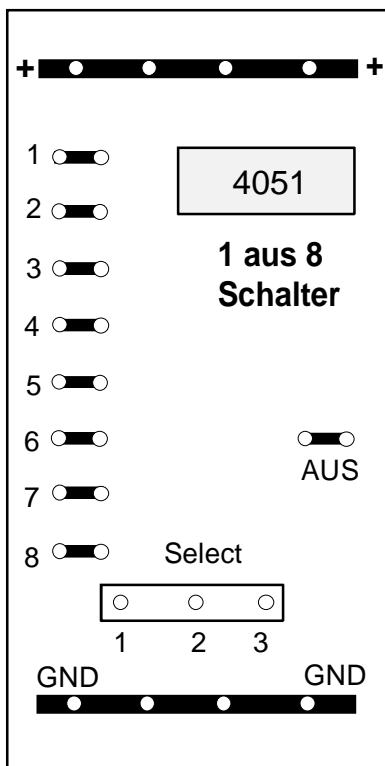


1-aus-8-Schalter

Die Platine hat die AbmaÙe 50*100mm².

Neben der Platine sind noch die folgenden Bauteile erforderlich:

1. 16-Pin IC-Fassung
2. CMOS 4051, 1-aus-8-Schalter
3. 29 Löt Nägel 1,3 mm

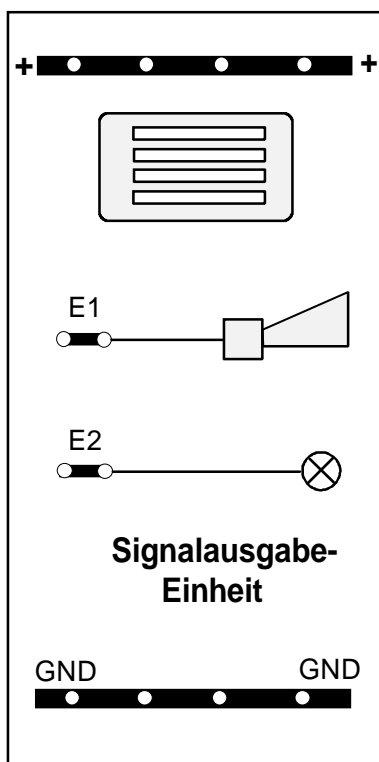


Signalausgabeeinheit

Die Platine hat die Abmaße 50*100mm².

Neben der Platine sind noch die folgenden Bauteile erforderlich:

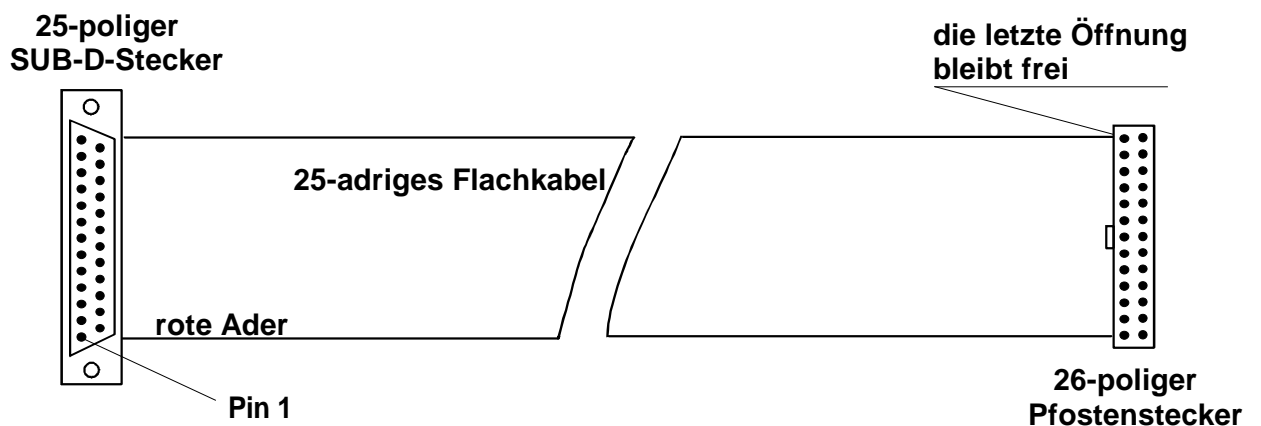
1. Summer
2. 3,3k Ω Widerstand 2 mal, 220 Ω Widerstand 1 mal
3. BC238 bzw. 548 2 mal
4. 12 Lötnägel 1.3 mm



Verbindungskabel Computer-Centronicsadapter

Zur Herstellung des Adapterkabels sind folgende Bauteile erforderlich:

1. 25-adriges Flachkabel 1m
2. 26-poliger Pfostenstecker mit Zugentlastung
3. 25-poliger SUB-D-Stecker mit Zugentlastung

Darstellung ohne Zugentlastung**Darstellung mit Zugentlastung**