

HACKER SHEET

LPH7889-1

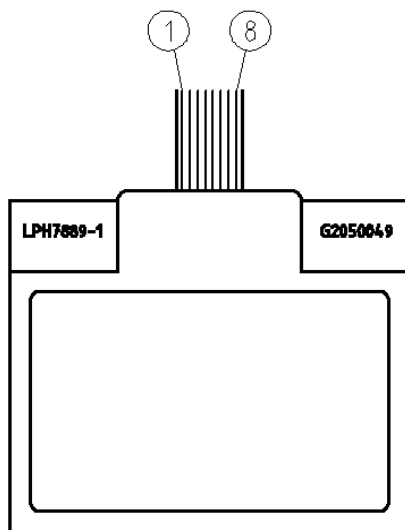
54×101 Pixel Matrix LCD

Achtung!
Das ist kein Offizielle Datenblatt!
Sämtliche Verantwortung Wirt
ausgeschlossen!

Allgemeine Daten:

		Bemerkung
Hersteller:	Unbekannt	Wahrscheinlich Chineses, wie alles heutzutage.
Vermutete Treiber IC:	PHILIPS OM6201	Es ist wahrscheinlich eine lizensierte, oder simpel geklaute fern östliche Kopie dieses IC-s.
Auflösung:	54×101	Memoiren flehe ist größer.
Farbtiefe:	1	Nah ja, 1 ist auch eine zahl. :)
Speise Spannung	2,6V	Vermutlich ist es ein breiteres Bereich bis zu 4,5 V.

Kivezetések és feladatuk:



Pin	Benennung	Aufgabe
1	+U	Speise Spannung
2	CLK	Seriell clock
3	SDIN	Seriell data in
4	D/C	Data/Command
5	SCE	Chip enable
6	GND	Grund
7	VLCD	Lcd Spannung ein/aus
8	RESET	Reset

D/C scheint negiert zu sein, bei der Kommunikation bedeutet das, das die Datentransfer, und befehle Bytes den platz tauschen, wenn du einen selbst geschriebenen Programm block zum Kommunikation verwendest, ziehe es in Betracht, das es daran liegen kann, das dein Programm block nicht funktioniert.

Originell ist diese LCD mit einen fast unbenutzbaren Stecker aufgerüstet, auf den Bild ist diese Stecker nicht sichtbar. Bei den Test wurde diese Stecker simpel abgeschnitten, und ein teil des „flexible Leiterplatten Kabels“ mit Stahlwollen von den stützenden Lackschicht befreit, damit es von oben lötbar ist.

Empfehlung:

Es ist ratsam ein Ebenbild des Kabels auf die Leiterplatte auszubilden, die Kabel ende in 5-10mm Länge auf die Leiterplatte zu kleben, und die nötige elektrische Kontakte mit bloßen Lot, von oben herzustellen.

Das sesshaft die nötige mechanische Stabilität, und entlastet die Lötung.

Auf keinen Fall sollte man diesen flexiblen Kabel ganz entfernen!

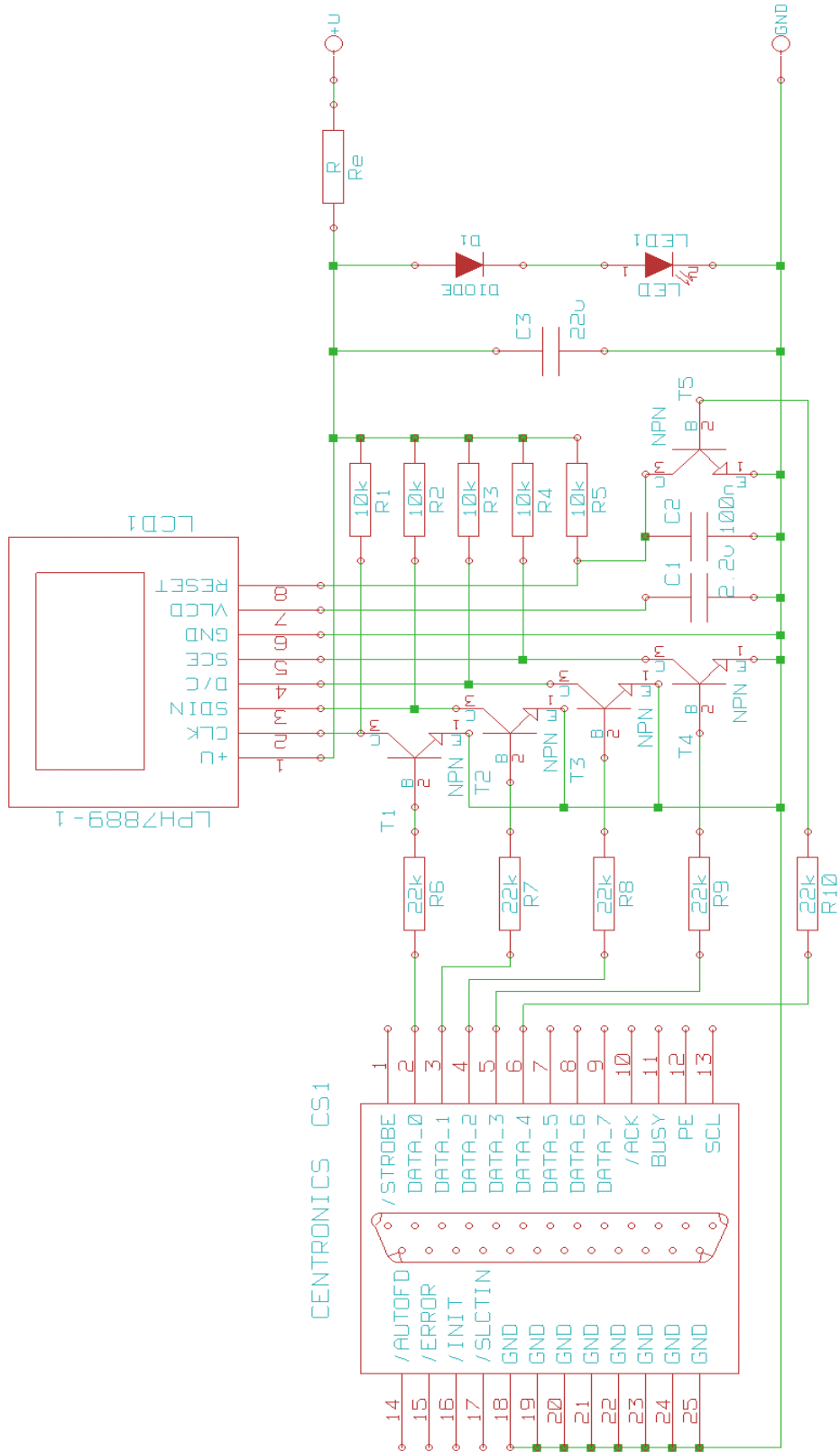
Glaub mir, ich habe es mit eine LCD gemacht! Es lohnt sich nicht!

Behalte von den Kabel, so viel wie es nur möglich ist, damit du Material für mehrere Verzuge hast!

Testschaltplan:

Zum antreiben des LCD-s wurde Der folgenden Elektronik verwendet:

Wert von R_e Widerstand ist von Speisespannung abhängig. Zum Stabilisierung wurde ein rotes LED, und eine normale Silizium Diode parallel geschaltet. (Was eine Fachbarbarese, aber effektive Lösung ist.) VLCD Spannung Wirt von Treiber IC hergestellt.



VLCD Spannung:

Funktionierende Bias Systeme wofür die nötige VLCD Spannung herstellbar ist: sind 4,5,6 und 7

die Justierung des VLCD Parameters sind von den benutzten Bias System, und den Speisespannung abhängig. Die von Hersteller vorgeschriebene werte sind wahrscheinlich System 5 oder 6. (Aladins ist das nur eine Vermutung)

Wenn die VLCD Parametern ein zu hohes VLCD wert Ergeben, werden auf den LCD, die nicht gesetzte Punkte mit einen schwächeren Kontrast ebenfalls angezeigt. (Das kann so weit gehen, das sämtliche Segmente mit den selben Kontrast angezeigt werden, unabhängig davon ob sie gesetzt sind oder nicht und eine Schwarzes anzeige ergeben)

Eine Konfiguration die ein zu niedriges VLCD Spannung bestimmen, ergeben einen schwachen Kontrast, oder überhaupt kein Bild.

Die VLCD Spannung grundgesetzlich beschreibenden Parametern sind:

Hvgen stages	(beschreibt Multiplikation)
Set HIGH or LOW program range Vop verstanden werden muss)	(Beschreibt wie die wert des Vop Registers)
Set Vop Parameter)	(Beschreibt die Intensität des vorigen)

Für detaillierte Information, sehe in DATASHEET des OM 6201 IC-s nach!

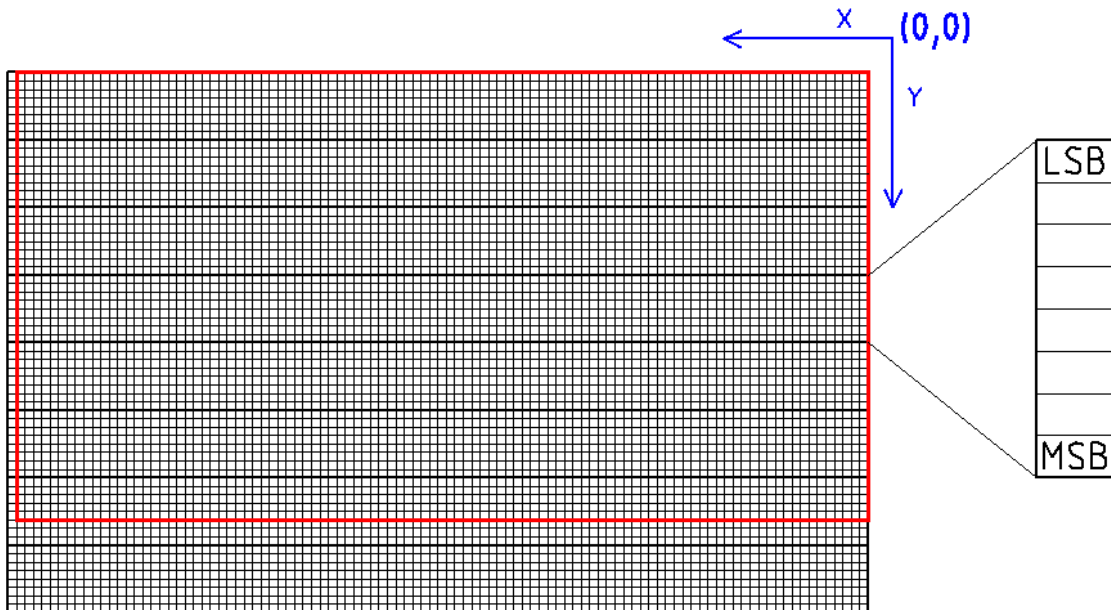
Eine funktionierende Einstellung für diese LCD ist:

Speisespannung = 2.6 V

00100001b	(Einschalten H auf 1)
00010101b	(Bias System 5 ausgewählt)
00001010b	(Hvgen mode 2)
11110000b	(Vop=104)
00100001b	(Einschalten H auf 0)
00010000b	(Vlcd Programmierung Bereich hoch)

Das ergibt eine VLCD Spannung von ungefähr 6,4V

Adressierungssystem des LCD-s:



Die Memorien flehe ist in 8 horizontal liegenden Linien (Y wert), und 102 vertikal liegenden Linien adressierbar.

Mit den Datentransfer werden 8 vertikal liegenden Pixel gleichzeitig gesetzt. (LSB oben MSB unten)

Horizontale Adressierung (V=0):

Die Adressierungspoiter Wirt nach jede Datentransfer Befehl, nach links inkrementiert, wenn es die letzte Zeile erreicht hat, Wirt es zurück-, und auf die nächste Y Linie gesetzt.

Vertikale Adressierung (V=1):

Die Adressierungspoiter Wirt nach jede Datentransfer Befehl, nach unten inkrementiert, wenn es die letzte Zeile erreicht hat, Wirt es zurück-, und auf die nächste X Linie gesetzt.

Über Positionierung des Datenpointers, und den „V“ Bit, sehe in DATASHEET des OM 6201 IC-s nach!

Sichtbar aus den Memorienfleche ist nur, die mit rot eingekreiste flehe. Die daraus liegenden Bereiche sind zwar adressierbar, und beschreibbar, werden aber nicht angezeigt.

Rutin:

```
//File:      LPH7889-1.cpp
//Lisens:   GPL
//Author:   MB
//Date:     2007 Okt.

#include <sys/io.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/resource.h>    //for priority control.
#include <binhex.h>        //Für Binären -> Hexadecimelen umwandlung

//Achtung!
//binhex.h ist eine einfache header wo die Binären werte,
//mit hexadecimalen werte verbunden werden, undswar so:
//#define B00000000B    0x00
//#define B00000001B    0x01
//#define B00000010B    0x02
//.
//.
//#define B11111111B    0xFF
//Ohne diese Definicionen reie ist dieses rutin nicht funktionsfehig.
//Wenn du dieses File nicht herstellen willst, musst du semtliche binar
//angegebene werte zum hexadecimalen wehrt umschreiben.
////////////////////////////////////
//  XXXX  XXXXX  XXXXX  X    X  XXXXX  X    X  XXXX  //
//  X    X    X    X    XX  XX  X    XX  X  X    X  //
//  X    X    X    X    X  X  X    X  X  X  X  //
//  XXXX  X    XXX  X  XX  X  XXXX  X  X  X  XXXX  //
//  X    X    X    X    X    X  X    X  X  X  X  //
//  XXXX  XXXXX  XXXXX  X    X  XXXXX  X    X  XXXX  //
////////////////////////////////////
/*SIEMENS LPH7889-1 kijelző lábkiosztása:
XXXXXXXXXXXXXXXXX
X                X
X  XXXXXX  X--1 +U
X X        X  X--2 CLK
X X        X  X--3 SDIN
X X        X  X--4 D/C
X X        X  X--5 SCE
X X        X  X--6 GND
X X        X  X--7 VLCD
X  XXXXXX  X--8 RESET
X                X
XXXXXXXXXXXXXXXXX
alkalmazott meghajtás:
                                +U=3,6V
                                |
                                | | R=4,7k
                                | |
                                | |
                                | |----- CLK, SDIN, D/C, SCE
                                |
                                r=33k
                                b|/c
Printer port Dx---[====]--|    NPN pl:BC107 BC182 stb.
                                | \e
                                |
                                ---
*/
```

```

////////////////////////////////////
//----[Port Zugriff für unseren Programm sichern]----//
////////////////////////////////////
void rootaces()
{
    setpriority(PRIO_PROCESS,-10,0);          //Prioritás beállítása
    if (iopl(3))
    {
        perror("\niopl(3) Fejreált. Hozzáférés a porthoz sikertelen. Lehet, hogy nem
vagy root?");
    }
    setgid(getgid());
    setuid(getuid());
}

////////////////////////////////////
//-[LPH7889 konfigurálható struktúrája]-//
////////////////////////////////////
typedef struct
{
    long int          Bitrate;                //Órajel us-ben
    unsigned char     Configmask[5];         //vonalak maszkjai a portra
    int               Portnum;              //Printerport címe
    ///Config Wirt zwar nicht benutzt, aber es sieht schön aus.
    unsigned char     Config;               //Speciális működést előíró bitek
    ///////////////////////////////////
    //   Helyiérték  Ponált      Negált      //
    //   Bit0        CLK ponált    CLK negált  //
    //   Bit1        DAT ponált    DAT negált  //
    //   Bit2        Várakozás ACK-ra ACK figyelmen kívül hagyása//
    ///////////////////////////////////
}LPH7889_config;

//Configmaszk értékei//
#define S_CLK        0
#define S_SDIN       1
#define S_D_C        2
#define S_SCE        3

//--[LPH7889 konfigurálható deklarációja]--//
LPH7889_config LPH7889config;

/*-----[Konfigurálás]-----*/
void LPH7889_conf(void)
{
    LPH7889config.Portnum=          0X378;    //Printerport címe
    LPH7889config.Bitrate=          0;        //Ja. Null ist auch möglich.
    LPH7889config.Configmask[S_CLK]= 0x01;    //printer port D4
    LPH7889config.Configmask[S_D_C]= 0x04;    //printer port D2
    LPH7889config.Configmask[S_SCE]= 0x08;    //printer port D3
    LPH7889config.Configmask[S_SDIN]= 0x02;    //printer port D0
    //Ja also, Einmal Wirt es auch was machen
    LPH7889config.Config=           B00000011B;
}

```

```

////////////////////////////////////
//-----[ Kijelző meghajtó rutin ]-----//
////////////////////////////////////
void LPH7889(char DPC, unsigned char Data)
{
char hurok; //Hurokváltozó
char value; //Port puffer
unsigned char bitmask; //Dat-ot maszkoló változó
outb(0x00, LPH7889config.Portnum); //0x00 Kiírása a portra
usleep(LPH7889config.Bitrate); //1 ütem várakozás
//-[1. init]-----
value=inb(LPH7889config.Portnum); //Portérték beolvasása
hurok=0; //Hurokváltozó inicializálása
bitmask=B10000000B; //Maszk beállítása
for(;;)
{
switch(hurok)
{
case 0: //-----[Inicializálás]-----
value=value|LPH7889config.Configmask[S_CLK]; //CLK lehúzása
//D/C kiírása
if (DPC==0) value=value|LPH7889config.Configmask[S_D_C];
else value=value&~LPH7889config.Configmask[S_D_C];
//Adatbit kiírása
if(Data&bitmask) value=value&~LPH7889config.Configmask[S_SDIN];
else value=value|LPH7889config.Configmask[S_SDIN];
//-----[1. ütem vége]-----
outb(value, LPH7889config.Portnum); //Kiírás portra
usleep(LPH7889config.Bitrate); //1 ütem várakozás
//Engedélyezés
value=value|LPH7889config.Configmask[S_SCE];
break;
case 7: //Utolsó ütem
value=value^LPH7889config.Configmask[S_D_C]; //D/C negálása
//itt okkal nincs break!!!
default://Átvitel
value=value^LPH7889config.Configmask[S_CLK]; //CLK negálása
//adatbit kiírása
if(Data&bitmask) value=value&~LPH7889config.Configmask[S_SDIN];
else value=value|LPH7889config.Configmask[S_SDIN];
break;
}
outb(value, LPH7889config.Portnum); //Kiírás portra
usleep(LPH7889config.Bitrate); //1 ütem várakozás
value=value^LPH7889config.Configmask[S_CLK]; //CLK negálása (Felhúzás)
outb(value, LPH7889config.Portnum); //Kiírás portra
usleep(LPH7889config.Bitrate); //1 ütem várakozás
if (hurok==7) break; //Ha utolsó bit, kilép
hurok++; //Hurokváltozó növelése
bitmask=bitmask>>1; //Mask forgatása
}
value=value^LPH7889config.Configmask[S_CLK]; //CLK negálása (Lehúzás)
outb(value, LPH7889config.Portnum); //Kiírás portra
usleep(LPH7889config.Bitrate); //1 ütem várakozás
//Engedélyezés elvétele
value=value&~LPH7889config.Configmask[S_SCE];
outb(value, LPH7889config.Portnum); //Kiírás portra
usleep(LPH7889config.Bitrate); //1 ütem várakozás
}

////////////////////////////////////
//-----{Fájl vége}-----//
////////////////////////////////////

```