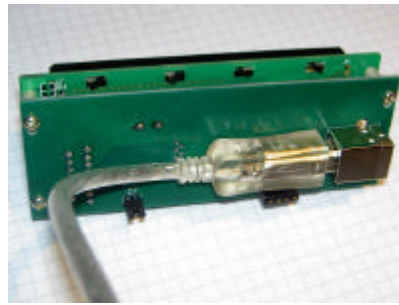


USB-LCD-Textdisplay



Features

- Anschlussfertiges USB-LCD-Display mit 4x20-Zeichen
- Hochwertiges LCD-Textmodul mit Hintergrundbeleuchtung
- Kein Abgleich der Kontrastspannung erforderlich
- Versorgungsspannung: USB
- Display-Controller: HD44780 kompatibel
- USB-Chipsatz: CH341A
- Integrierte I²C/TWI-Master-Schnittstelle
- Systemvoraussetzungen: XP, 2000, Vista
- Software-Treiber: ProfiLab-Expert, LCD-Hype, LCD Smartie, MediaPortal
- Einfache Programmierung per ActiveX-Steuerelement (OCX)
- Programmbeispiele in C++, Delphi und Visual Basic.

Das USB-LCD-Display vereint ein hochwertiges LCD-Textmodul mit 4x20-Zeichen und Hintergrundbeleuchtung und ein USB-Interface zu einer anschlussfertigen Einheit. Besondere Merkmale sind die Spannungsversorgung aus dem USB und eine automatische Kontrasteinstellung ohne manuellen Abgleich der Kontrastspannung. Eine zusätzliche I²C/TWI-Schnittstelle erlaubt den Anschluss eigener Schaltungserweiterungen oder fertiger I²C-Schaltungen, wie z.B. das MicroChip-I²C-Board aus unserem Lieferprogramm. Die Abmessungen und die Lage des USB-Anschluss machen das Gerät neben dem externen Betrieb auch für den PC-Einbau interessant.

Download-Link für Anleitung, Treiber, Beispiele

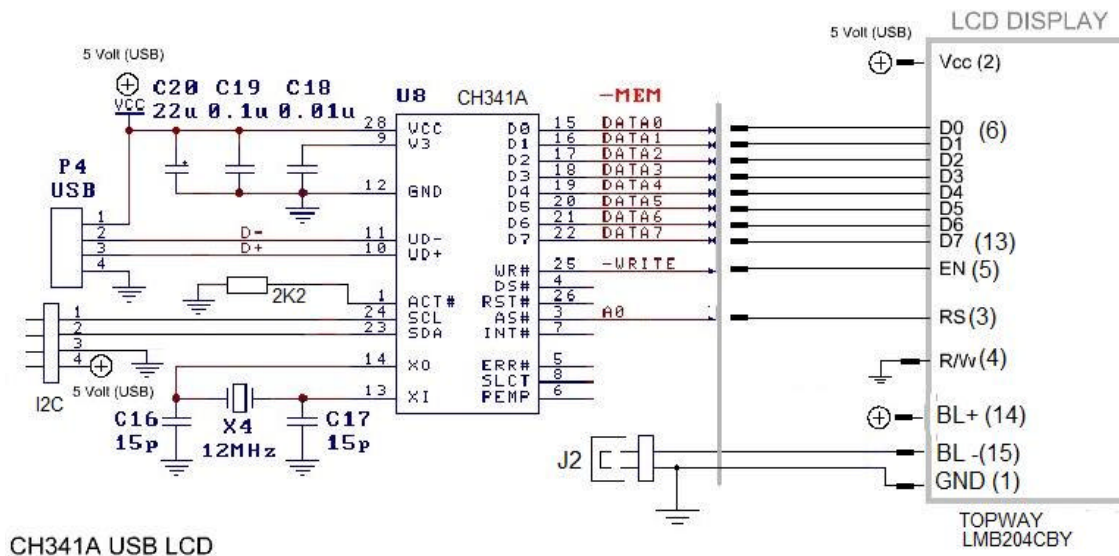
http://www.abacom-online.de/div/setup_usb_lcd.exe

Installation

- Bevor Sie das Display mit dem USB verbinden, laden Sie bitte unbedingt die die zugehörige Software herunter und führen Sie das Installationsprogramm aus.
- Danach können Sie das Display mit dem USB verbinden und die gewohnte Windows-Plug&Play-Installation durchführen.
- Nun ist Ihr Display einsatzbereit. Weitere Informationen zur Verwendung des Display finden Sie jetzt in der Anleitung und in den weiteren Dateien, die das Installationsprogramm installiert hat.

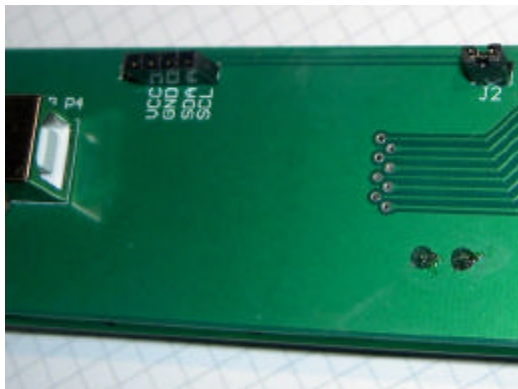
LCD-Display

Die Ansteuerung des HD44780-kompatiblen Display-Controllers erfolgt intern über einen 8-Bit breiten, parallelen Datenbus. Die Umsetzung der seriellen Daten des USB auf den parallelen Bus des Display-Moduls übernimmt ein CH341A-Interface-Chip. Dieser erzeugt neben den acht Datensignalen (D0..D7) für das Display, auch die Signale RS (Register Select) und EN (Enable). Der R/W-Eingang des Display liegt fest auf Masse, so dass Daten zum Display geschrieben werden können.



Hintergrundbeleuchtung

Die Spannungsversorgung der Hintergrundbeleuchtung erfolgt ebenfalls aus dem USB. Dabei ist der Masseanschluss der Beleuchtung über einen Jumper (J2) geführt. Dieser kann bei Bedarf entfernt und z.B. durch einen mechanischen Schalter oder einen Transistors ersetzt werden, um die Beleuchtung abschaltbar zu machen. Die Stromaufnahme der Beleuchtung liegt bei etwa 120mA.



I²C-Master-Anschluss

Eine vierpolige Buchsenleiste stellt die I²C-Signale und die USB-Versorgungsspannung für eigene Schaltungserweiterungen bereit. Dabei darf die Gesamtstromentnahme aus dem USB-Port 500mA nicht übersteigen. Die I²C-Pullup-Widerstände (meist 2K2) müssen in der Anwenderschaltung vorgesehen werden.

Software

Laden Sie die Software zum Display von unseren Internetseiten herunter. Das Setup-Programm kopiert folgende Ordner auf Ihre Festplatte:

API

Dieser Ordner enthält Dateien und Beispiele für die eigene Programmierung des Displays mit verschiedenen Programmiersprachen.

DOC

Enthält einige Datenblätter und evtl. nützliche Zusatzinformationen.

DRIVER:

Dieser Ordner enthält die USB-Treiber die Sie für die Windows Plug&Play-Installation benötigen.

Nach dem Anstecken am USB muss die Treiberinstallation für den CH341A-Interfacechip erfolgen. Den Treiber finden Sie im Ordner DRIVER. Danach ist das Display einsetzbar. Die Adressierung des Displays erfolgt über eine fortlaufende Gerätenummer des CH341A-Interface-Chips. Alle am System angeschlossenen CH341A-Chips erhalten automatisch eine fortlaufende Nummer beginnend mit 0.

Programme die Daten an das Display senden wollen, öffnen zunächst anhand der Gerätenummer einen Datenkanal zum CH341A-Chip, über den dann die Datenübertragung mit dem HD44780-Display-Controller erfolgt. In der Anwendungssoftware muss in der Regel nur die richtige Gerätenummer eingestellt werden, um das Display anzusprechen.

TEST

enthält ein einfaches Testprogramm für einen ersten Funktionstest. Es zeigt Ihnen die aktuelle Gerätenummer auf dem Display an.

PLUGINS

Hier finden Sie die Dateien die Sie für die Verwendung des Displays mit Software anderer Hersteller benötigen: LCD Hype, LCD Smartie und MediaPortal. In den Ordner befinden sich jeweils ReadMe-Dateien die die Verwendung erläutern.

Funktionstest

Für einen ersten Funktionstest und zum Ermitteln der richtigen Gerätenummer finden Sie ein Testprogramm im Ordner \TEST. Starten Sie das Testprogramm (USB_LCD_TEST.EXE) und betätigen Sie die Schaltfläche TEST. Auf dem LCD-Display wird nun zugehörige Gerätenummer (Device #) angezeigt.

ProfiLab Expert

Beispielprojekte für ProfiLab Expert 4.0 finden Sie im Ordner \PROFILAB.

Für die Ansteuerung des Displays sind im wesentlichen zwei ProfiLab-Komponenten erforderlich. Die Komponente "LCD-Display" erzeugt die Daten für die Ansteuerung des HD44780-LCD-Controllers. Die Komponente "CH341A" überträgt anschliessend diese Daten an den CH341A-Interface-Chip, der diese unmittelbar an den Display-Controller weitergibt.

Der HD44780-Controller besitzt zwei Register (Steuer- und Datenregister) . Um Daten in das gewünschte Register zu schreiben, werden zwei CH341A-API-Funktionen verwendet:

CH341AMEMWRITEADDR0
CH341AMEMWRITEADDR1

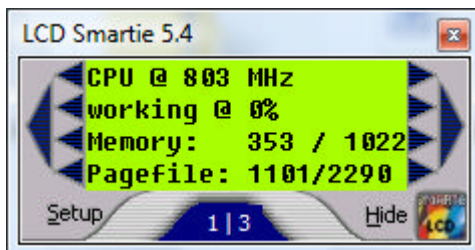
Auch die Ansteuerung des I²C erfolgt in ProfiLab über die CH341A-API-Funktionen

Weitere Software

Das LCD-Display kann mit verschiedenen Programmen anderer Hersteller verwendet werden. Die dazu notwendigen Dateien finden Sie im Ordner \PLUGINS. Lesen Sie auch die ReadMe-Datei in den jeweiligen Ordnern.



LCD-Hype ist ein kostenloses Programm zur Ansteuerung von LCD-Displays.
(<http://www.lcdhype.de.vu>)



LCD Smartie ist ein weiteres kostenloses Programm zur Ansteuerung von LCD-Displays. (<http://lcdsmartie.sourceforge.net>) .



MediaPortal ist eine Multimedia-Software, die ebenfalls LCD-Displays unterstützt.
(<http://www.team-mediaportal.de>)

Programmierschnittstelle

Programmierbeispiele für verschiedene Programmiersprachen finden Sie im Ordner \API der Softwareinstallation.

Die Programmierung des Displays und der I²C-Schnittstelle erfolgt mit Hilfe eines ActiveX-Steuerelements. Dieses befindet sich in der Datei ABACOM_USB_LCD.OCX. Das Steuerelement wird bei der Installation der Software automatisch unter dem Namen "ABACOM USB LCD" registriert und kann mit allen Programmiersprachen verwendet werden, die den Import von ActiveX-Steuerelementen unterstützen. Grundlegende Informationen zum Import von ActiveX-Steuerelementen entnehmen Sie bitte der Anleitung zu Ihrer Programmiersprache. Dies nachstehenden Programmauszüge stellen keinen lauffähigen Programmcode dar. Sie dienen nur zur Erläuterung des Funktionsprinzips. Bitte beachten Sie auch die mitgelieferten Programmbeispiele.

Für die Programmierung erzeugt man zunächst ein "ABACOM USB LCD" – Objekt vom Typ USBLCDX. Dies kann entweder dynamisch zur Laufzeit oder mit Hilfe des Formular-Designers der Programmiersprache geschehen.

```
TUSBLCDX *LCD ;

LCD = new TUSBLCDX(this);
LCD->Parent = this;
LCD->Left = 24;
LCD->Top = 40;
```

Neben den gewohnten Eigenschaften zur Verwaltung des Steuerelements, wie .z.B. LEFT, TOP, etc. verfügt das Steuerelement über Methoden und Eigenschaften, die speziell der Programmierung des LCD-Displays und der I²C-Schnittstelle dienen, wie z.B.

```
LCD->LCD_ROW1TEXT = ...
```

oder

```
LCD->I2C_Write(...)
```

Diese Namen dieser Eigenschaften und Methoden beginnen entsprechend Ihrer Verwendung mit "LCD_" bzw. "I2C_".

Programmierung des LCD-Displays

Den vier Textzeilen des Displays sind vier VARIANT-Eigenschaften zugeordnet. Das Setzen dieser Eigenschaften gibt den Text auf dem Display aus:

```
LCD->LCD_ROW1TEXT = "Text für Zeile 1";
LCD->LCD_ROW2TEXT = "Text für Zeile 2";
LCD->LCD_ROW3TEXT = "Text für Zeile 3";
LCD->LCD_ROW4TEXT = "Text für Zeile 4";
```

Das Displayformat, d.h. die Anzahl von Textzeilen und Spalten kann bei Bedarf so eingestellt werden:

```
LCD->LCD_Columns = 20;
LCD->LCD_Rows = 4;
```

Das Standardformat ist 4x20 Zeichen.

Übersteigt die Textlänge einer Zeile die Anzahl der Textspalten des Displays, so erfolgt automatisch ein "Roller" der Textzeile (Autoscroll).

Alternativ kann eine Textausgabe mit der Print-Methode erfolgen. Dabei wird der Text in die unterste Textzeile des Displays ausgegeben, nachdem zuvor alle darüber liegenden Zeilen nach oben verschoben wurden. Dies eignet sich z.B. für Listen.

```
LCD->LCD_Print( time );
```

Zum Löschen einer Zeile gibt man einfach einen Leerstring aus:

```
LCD->LCD_ROW1TEXT = "";
```

Für Umlaute und ß werden automatisch Ersetzungen vorgenommen:

```
LCD->LCD_ROW1TEXT = "Olivenöl";
```

Wird z.B. als "Olivenoel" ausgegeben. Ferner besteht die Möglichkeit benutzerdefinierte Zeichenersetzungen vorzunehmen, um Anpassungen zwischen den Windows-Zeichensatz und dem Display-Zeichensatz vorzunehmen. So befindet z.B. das °-Zeichen chr(176) im Text-Display als chr(223). Dazu besitzt unser Steuerelement eine Liste mit Zeichenentsprechungen in die die Übersetzungen eingetragen werden können:

```
LCD->LCD_AddCharTranslation(176,223); // Grad Zeichen ersetzen  
LCD->LCD_AddCharTranslation(65 , 66); // aus "A" wird "B"
```

...

Sobald die Liste geleert wird, finden keine Ersetzungen mehr statt und die Original-Zeichen werden wieder ausgegeben:

```
LCD->LCD_ClearCharTranslations;
```

Der HD44780 erlaubt die Definition von acht benutzerdefinierten Zeichen chr(0)...chr(7), deren Bitmuster sich aus jeweils acht Datenbytes zusammensetzt.

```
LCD->LCD_DefineCustomChar(1, "1F 1F 1F 1F 1F 1F 1F 1F");
```

Hier wird z.B. chr(1) als schwarzer Block definiert, der dann überall erscheint, wo ein chr(1) im Text ausgegeben wird. Für Berechnung der Datenbytes stehen im Internet Werkzeuge zur Verfügung.

(z.B. <http://www.quinapalus.com/hd44780udg.html>) Die Übergabe der Datenbytes erfolgt als lesbare Zeichenkette in denen die acht Datenbytes in HEX-Darstellung – getrennt durch Leerzeichen – aufgelistet sind. (HEX-Dump)

Weitere Funktionen des HD44780-Display-Controllers lassen sich bei Bedarf ausführen, in dem man direkt in das Steuerregister des Controllers schreibt:

```
if (CheckBox1->Checked)  
{LCD->LCD_WriteToControlRegister(12); // Display ON  
else  
{LCD->LCD_WriteToControlRegister(8); // Display OFF
```

Die Funktionen sind dem Datenblatt des HD44780 zu entnehmen.

Für die Verwendung mehrerer Display legt man ggf. mehrere LCD-Objekte an und setzt die Gerätenummer LCD_DeviceNo entsprechend:

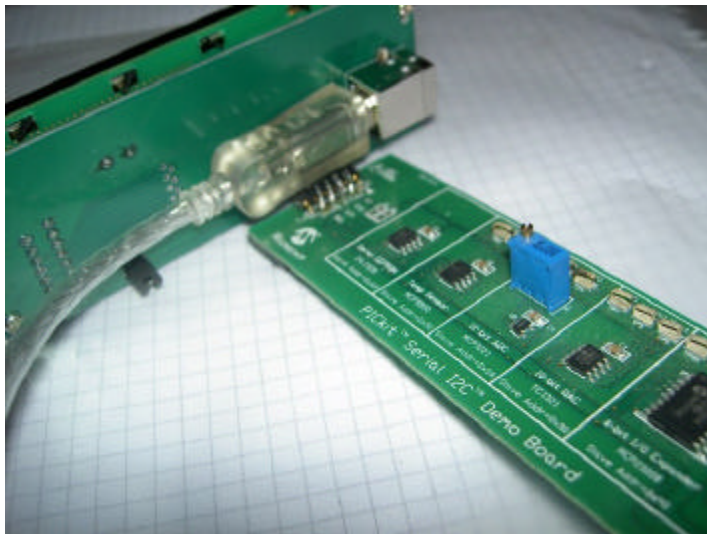
```
LCD1->LCD_DeviceNo = 0; //erstes Display  
LCD2->LCD_DeviceNo = 1; //zweites Display  
Usw.
```

Programmierung der I²C/TWI-Schnittstelle

Das Display besitzt eine I²C-Master-Schnittstelle, die es erlaubt zusätzliche I²C-Peripherie-Chips am PC zu betreiben. Dazu sind Kenntnisse der I²C-Technik und das Verständnis der Datenblätter der verwendeten Chips unerlässlich.

Die Programmbeispiele beziehen sich auf eine I²C-Demonstrationsschaltung von MicroChip(TM), die anschlussfertig als Zubehör bei uns erworben werden kann.
http://www.electronic-software-shop.com/product_info.php?pName=microchip-ic-board-p-17&cName=hardware-ic-twi-c-2_13

Ebenso lassen sich - bei entsprechender Programmierung - natürlich auch andere I²C-Schaltungen an der Schnittstelle betreiben.



Für die Kommunikation mit I²C-Chips stehen folgende Funktionen zur Verfügung:

Byteweises Schreiben und Lesen Chip-Registern

Die Methode

```
LCD->I2C_Write(0x20,0x09,LEDStatus)
```

schreibt ein Byte (hier LEDStatus) an die Chip-(Slave)-Adresse 0x20 in ein Chip-Register (hier 0x09).

```
LCD->I2C_Read(0x20,0x09,InData)
```

Liest ein Register (hier 0x09) eines I²C-Chips, dessen Chip-Adresse 0x20 ist. Der Rückgabewert InData enthält den gelesenen Byte-Wert als lesbare Zeichenkette (HEX-Dump; z.B "A7")

Direktes Schreiben und/oder Lesen von Busdaten

Ein direktes Schreiben und Lesen von Datenbytes über den I²C-Bus ist mit der Funktion STREAM-Funktion möglich:

```
OutData = "92 00";  
LCD->I2C_Stream(outData, 2, inData);
```

Die Datenübergabe (inData; outData) erfolgt als lesbare Zeichenkette (Hex-Dump). Das Beispiel schreibt zwei Bytes 0x92 und 0x00 auf den Bus. Die Anzahl der zu schreibenden Bytes wird aus der Länge des Hex-Dumps (OutData) implizit ermittelt.

Die Anzahl der zu lesenden Bytes muss explizit angegeben werden (hier zwei). Der Rückgabewert inData liefert wiederum eine lesbare Zeichenkette der eingelesenen Bytes, wie z.B. "FE B6".

Die Funktion schreibt/liest die Daten wie gesagt – ohne jede eigene Logik – direkt über den I²C-Bus und ist damit sehr universell einsetzbar.

EEPROM

Das Schreiben und Lesen eines Standard-EEPROMS wird durch die folgende Funktionen ermöglicht:

```
LCD->I2C_WriteEEPROM(eprom24C02, 0, "FF FE");  
LCD->I2C_ReadEEPROM(eprom24C02, 0, 7, Data);
```

Der Funktion wird zum Schreiben der EEPROM-Typ, die Speicher-(Start-)Adresse und die Daten als lesbarer Hex-Dump übergeben. Zum Lesen muss zusätzlich die Anzahl der zu lesenden Bytes (hier: sieben) übergeben werden.

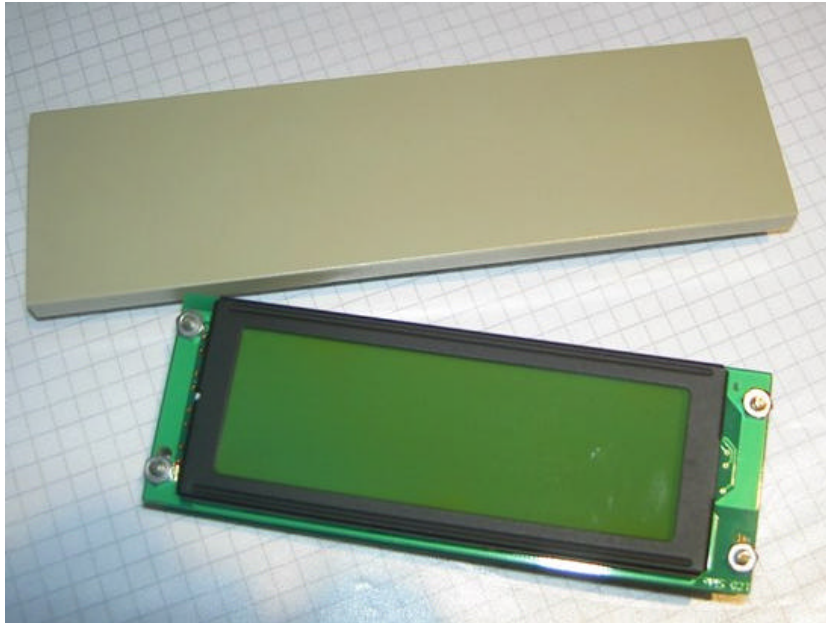
Taktfrequenz

Die I²C-Übertragungsgeschwindigkeit, d.h. die Taktfrequenz ist in vier Stufen einstellbar:

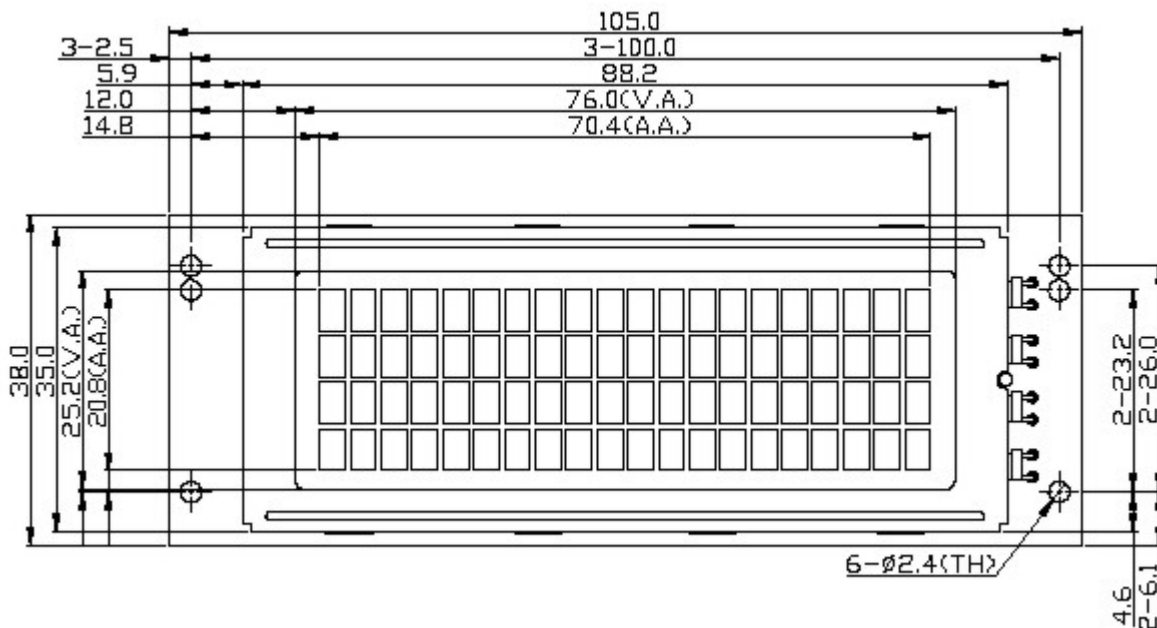
```
LCD->I2C_Speed = slow_20kHz  
LCD->I2C_Speed = normal_100kHz  
LCD->I2C_Speed = fast_400kHz  
LCD->I2C_Speed = high_750kHz
```

Hohe Taktraten sind nicht unbedingt für alle I²C-Chips geeignet.

Abmessungen



Größenvergleich mit einer Frontplattenabdeckung einer 5,25"-PC-Einschubs. Die Einbautiefe liegt bei etwa 35 mm.



Links

<http://www.sprut.de/electronic/lcd/>

<http://www.quinpalus.com/hd44780udg.html>

http://en.wikipedia.org/wiki/HD44780_Character_LCD

http://www.electronic-software-shop.com/product_info.php?pName=ch341a-usbinterfacemodule-dip28-p-30

http://www.electronic-software-shop.com/product_info.php?pName=microchip-ic-board-p-17&cName=hardware-ic-twi-c-2_13