

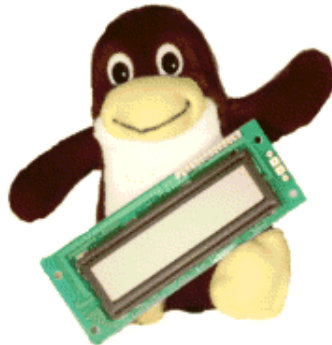
door Jan Svenungsson
<jan.svenungsson(at)linux.nu>

Over de auteur:

Jan gebruikt GNU/Linux al sinds 1996 en heeft in al die tijd 2 onverwachte reboots gehad (op stroomstoringen na).

Vertaald naar het Nederlands door:
Guus Snijders
<ghs/at/linuxfocus.org>

HD44780 compatibele LCD-displays begrijpen



Kort:

Dit artikel probeert een toelichting te geven over de HD44780 compatibele LCD-display.

We zullen de verbinding via de parallele poort en de programmatie van de LCD bespreken aan de hand van een programma met de naam LCDInfo.

De bedoeling is dat je niet alleen de display aansluit, een programma draait en alles wat je nodig hebt op de display krijgt, maar ook dat je leert inzien hoe je de hardware kunt laten doen wat jij wilt.

Introductie

Eerst heb je wat hard- en software nodig. Er wordt vanuit gegaan dat je reeds over een computer beschikt met een standaard parallele (printer) poort, waarop je GNU/Linux met gcc en glibc kunt draaien.

Ook is er een LCD display nodig welke HD44780 compatibel is, kabels om deze met je parallele poort te verbinden en een potentiometer (potmeter) als je het contrast wilt kunnen wijzigen. Voor de voeding van de display heb je bijna zeker meer power nodig dan de parallele poort levert, dus zul je de voeding ergens anders uit je computer moeten halen. De beste manier om dit te doen is met behulp van een standaard +5V stekker (degene die ook gebruikt worden voor floppy drives en harde schijven).

Als je de LCD hebt aangesloten, moet je weten hoe het werkt. Dit is wat meestal achterwege wordt gelaten in andere artikelen over dit onderwerp, maar ik zal proberen een paar achtergronden van de

display uit te leggen, wat verderop het programmeren zal vergemakkelijken.

Het laatste om te doen is om de display iets werkelijk nuttigs te laten weergeven. Als referentie zal ik een klein programma gebruiken, genaamd LCDInfo, welke de meeste features van de HD44780 ondersteunt, maar op dit moment nog niets weergeeft. Dit programma is alpha en ik werk eraan als ik tijd over heb.

Als je nog nooit hebt geprogrammeerd in C, kun je overwegen een beetje over C te lezen. Ik ga ervan uit dat je een beginner bent in C, daar dat ook mijn huidige niveau is.

Hoe aan te sluiten

Laten we eerst eens kijken naar de verschillende aansluitpennen (pins) op de LCD en uitleggen wat ze doen.

Pen 1 is genaamd VSS en wordt verondersteld aangesloten te zijn op GND(aarde).

Pen 2 is genaamd VDD en dit is de voedings pen welke zit op +5V.

Pen 3 is genaamd VLC en is aangesloten op de potmeter om het contrast van de display in te stellen.

Pen 4 is de RS pen; afhankelijk van deze pen bereidt de display zich voor op *instructions* of *data*.

Pen 5 is de R/W pen; deze bepaalt of de LCD *verstuurt* of *ontvangt*.

Pen 6 is de Enable pen. Als deze van laag naar hoog gaat en dan weer naar laag, zal de LCD de pennen 4,5 en 7-14 uitlezen.

De pennen 7-14 vormen samen de *data bus lijn* en zijn genaamd DB0-DB7, dit zijn de eigenlijke data bits die naar de LCD gestuurd worden en die bepalen wat er op het scherm verschijnt.

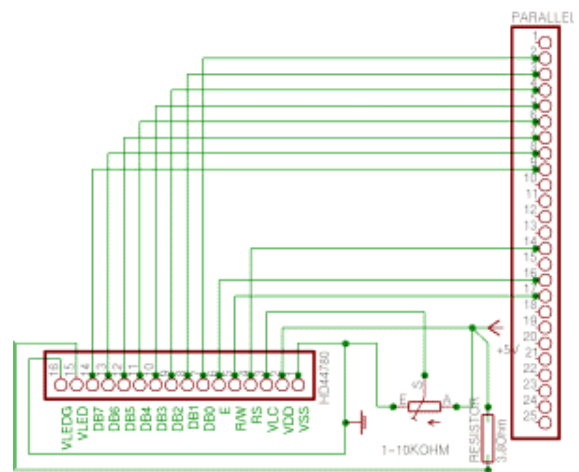
Pennen 15 en 16 zijn alleen aanwezig op displays met achtergrondverlichting (back light) en hangen aan +5V en GND met een 3.8 Ohm weerstand tussen pen 15 en +5V.

Om uit te vinden waar je deze zou moeten verbinden op de printer poort kun je kijken op het schema rechts waar ik geprobeerd heb om het zo duidelijk mogelijk te maken. Klik op het schema voor een groter plaatje.

Dit schema is alleen van toepassing als je het contrast van de display wilt wijzigen. Ik heb pen 3 en pen 1 simpelweg verbonden met GND wat goed werkt, als je een vreemd licht hebt in de kamer, zou je kunnen overwegen de potmeter toe te voegen.

Wees vooral voorzichtig wanneer je de voeding door de PC laat leveren. Als je de voeding vanaf de verkeerde kabel haalt krijg je +12V, waarmee je LCD geroosterd wordt. De kabel die je wilt is de rode. Geel is +12V en zwart is GND.

Als je dit hebt gedaan zou de eerste regel van de LCD (en de derde, indien aanwezig) zwart moeten zijn als je de PC inschakelt.



De werking van de LCD

De LCD doet niks totdat je hem vertelt wat te doen, het wacht eenvoudig totdat het een geldige stijging en daling leest (dit gebeurt wanneer we de enable pen hoog maken, even wachten en dan weer laag maken). Op dat moment bekijkt de display of het instructies of data ontvangt, en daarna of het data verstuurt of ontvangt en ten slotte worden de data bits verzonden of ontvangen.

In dit artikel ontvangen we geen informatie van de LCD en dus zal de R/W pen altijd laag zijn (wat schrijven betekent).

De RS pen zal laag zijn behalve wanneer we karakters weergeven, alles wat anders is wordt beschouwd als instructies.

Dit maakt het werkelijk eenvoudig om de display te programmeren.

Nu we dit allemaal weten, zullen we beginnen met de display in te schakelen en klaar te maken om informatie te ontvangen. Dit wordt gedaan met de initialisatie sequentie waar we de display vertellen dat hij ingeschakeld moet blijven, welke "function set" er gebruikt moet worden, enz.

De voeding zou reeds aangesloten moeten zijn als je spanning krijgt van een voedingskabel in de PC, anders is dit het eerste ding om in orde te maken.

Het volgende is de 'Functie set' welke afhankelijk is van het soort display dat je hebt.

Om het begrijpelijker te maken zal ik exact uitleggen wat er gebeurt tijdens het gebruik van de functie set.

DB2 is het 'Character Font' (lettertype) bit en deze zou *laag* moeten zijn, wat betekent '5x7 dot matrix'.

DB3 is de 'Display Lines' bit en zou *hoog* moeten zijn, wat '2 regels' betekent. Wat als je 4 regels op de display hebt? Geen zorgen, de eerste en de derde regel zijn identiek in het geheugen van de display, dus je gebruikt gewoon *hoog*.

DB4 is de Data Lengte bit en deze bepaalt of je 4 of 8 DB hebt, als je de display aansluit volgens mijn schema zou deze DB *hoog* moeten zijn. Dan wordt *DB5* hoog gezet om de display te vertellen dat dit inderdaad een "function set" instructie is. Controleer daarna dat *RS* en *R/W* *laag* zijn en doe een enable stijging en daling. Voor exacte timings moet je in de handleiding kijken, ik neem aan dat we slechts een paar microseconden pauzeren wanneer we wachten op de display; dit zou meer dan genoeg moeten zijn.

Hoe zit het met de code?

Hier zullen we de delen van het LCDInfo programma bespreken, waarvoor je zult moeten begrijpen hoe de interface van de HD44870 werkt. Je kunt het LCDInfo programma downloaden aan het eind van het artikel of kunt meteen een blik werpen op de C-code bestanden `iolcd.c` en `lcdinfo.c` door hier te klikken.

Nu hebben we de bovengenoemde instructies nodig, geschreven in C, en geloof me als ik zeg dat het simpel is. Ik zal stap voor stap door de code gaan en zelfs als je een C beginner bent, zul je het begrijpen.

Allereerst nemen we een paar header bestanden op en definiëren we een paar functies (check de broncode voor informatie). Daarna komt het leuke gedeelte.

```
#define D_REGISTER 0
#define I_REGISTER 2
#define WRITE_DATA 8
#define BASE 0x378
```

```
int main(void)
```

```

{
  ioperm(BASE, 3, 1);
  [CUT]
}

```

Dit is de eerste instructie in de main functie welke ons permissies geeft op de parallele poort. Base zou 0x378 of zoiets moeten zijn en het "3" deel betekent dat we toegang hebben tot 0x378, 0x379 en 0x380 welke in principe de volledige poort vormen.

De reden voor drie adressen is dat de poort is verdeeld in data, status en control (besturing). Voor ons betekent dit dat we eerst de data pennen moeten zetten en daarna de control pennen - we kunnen dit niet doen in één commando.

Het volgende om te doen is de functie set die hierboven beschreven is.

```

void function_set(void)
{
  outb(56, BASE);
}

```

Dit zet de DB pennen op 5x7 dot matrix, 2 regels, enz.

```

  outb(I_REGISTER + WRITE_DATA, BASE + 2);

```

Dit zet de RS en R/W pennen op 'instructie' en 'schrijven'. Ik heb globale variabelen gemaakt, I_REGISTER en WRITE_DATA en ze zijn gelijk aan 2 en 8.

Hierna volgt de 'enable' stijging en daling.

```

  outb(ENABLE + I_REGISTER + WRITE_DATA, BASE + 2);
  usleep(0);
  outb(I_REGISTER + WRITE_DATA, BASE + 2);
}

```

Deze code zet simpelweg de enable op hoog, wacht dan even en zet de enable terug laag. Het commando voor het usleep(0); is niet echt ideaal maar de timing code voor de display heb ik nog niet afgewerkt.

Een aantal onder u zal zich misschien afvragen waarom ik de RS en R/W aanzet in de code wanneer ik zeg dat ze laag moeten staan in de instructies. Dit is omdat de pennen 1, 14 en 17 hardwarematig geïnverteerd zijn, wat betekent dat als pen 14 "uit" is voor wat de printerpoort betreft, de pen in werkelijkheid "aan" is! Wel, ik had toch gezegd dat het eenvoudig zou zijn, nietwaar?

Het afbeelden van karakters

Misschien wil je ook een praktisch doel hebben voor je display, zoals het afbeelden van tekst? Geen probleem.

De code (code als in commando's) is gewoon het printen van een karakter en het zetten van de functies. Het enige dat we hoeven te doen is een paar variabelen aanpassen. Om dit te doen zetten we de RS niet naar instructies maar naar data. Dit maakt dat de functie print_character() er als volgt uitziet:

```

void print_character(int character)
{
  outb(D_REGISTER + WRITE_DATA, BASE + 2);
  outb(character, BASE);
  outb(ENABLE + D_REGISTER + WRITE_DATA, BASE + 2);
  usleep(0);
  outb(D_REGISTER + WRITE_DATA, BASE + 2);
}

```

}

Zoals je kunt zien hebben we "I_REGISTER" veranderd naar "D_REGISTER" en "56" naar "character", maar wat betekent dit? Als je de karakter codes in je manual bekijkt zul je het begrijpen.

We hoeven de functie alleen maar een karakter te voeren (omdat we C gebruiken hoeven we er niet eens eerst een integer van te maken) en het karakter zal verschijnen op de display. Makkelijk he?

Met deze code heb je het basis-skelet van een LCD programma, pas het maar aan volgens je eigen wensen. Laat het bijvoorbeeld vrij geheugen weergeven, actieve http connecties, of wat dan ook. Het LCDInfo programma bevat enkele voorbeelden van dingen die beschikbaar zijn in het proc bestandssysteem op een GNU/Linux computer.

Referenties

- Voor meer info over de printer poort, zie <http://et.nmsu.edu/~etti/fall96/computer/printer/printer.html> welke enkele voorbeelden bevat (een lokale kopie van dit artikel staat >hier<).
- Voor info over LCD programma's, check <http://lcdproc.omnipotent.net/>, een goed LCD programma.

Met dank aan Sven en Reinhold voor aanwijzingen.

- De broncode van het lcdinfo programma: `lcdinfo-0.02.tar.bz2`.
Updates zullen beschikbaar zijn op: <http://savannah.gnu.org/download/lcdinfo>

Site onderhouden door het LinuxFocus editors team © Jan Svenungson "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Vertaling info: en --> -- : Jan Svenungson < jan.svenungson@linux.nu > en --> nl: Guus Snijders < ghs@linuxfocus.org >
---	---