

18

Weer een eendraads LCD-interface...

met een microcontroller

Rob van Hest (Nederland)

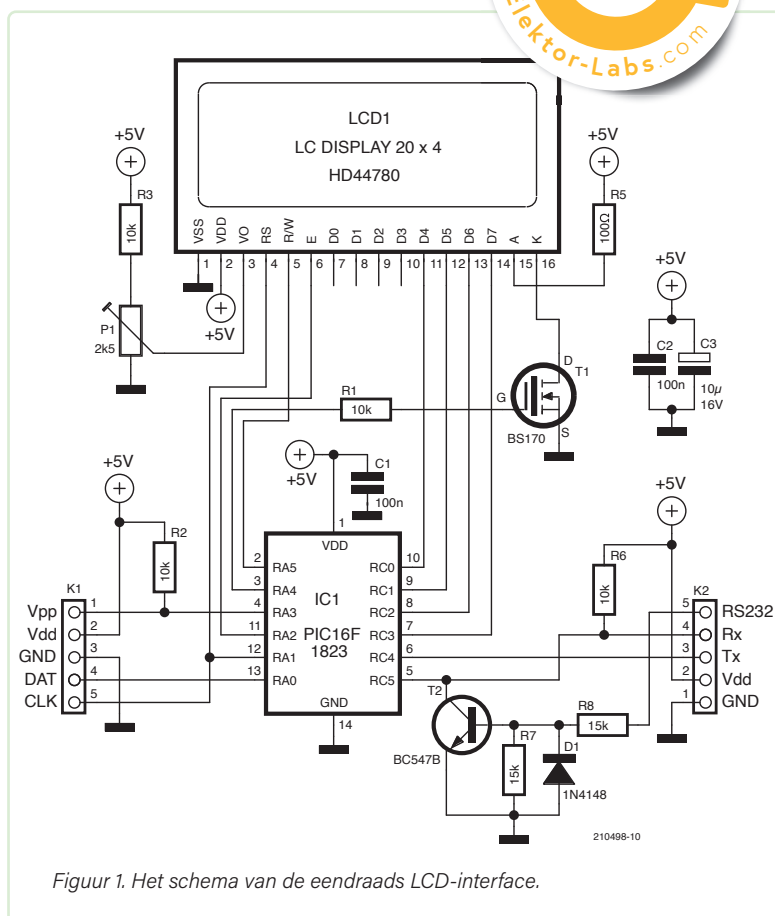
Er bestaan veel interfaces om data via één poort naar een standaard alfanumeriek LC-display te sturen, onder gebruikmaking van een schuifregister-IC. Hier gebruiken we een microcontroller die de data van het hostsysteem via een UART ontvangt.

Ik had voor een project een LC-display nodig, maar op de host was slechts één poort beschikbaar om dit aan te sturen. Geen probleem: er zijn veel voorbeelden beschikbaar (onder andere van *Elektor*) om zo'n interface te bouwen. De meeste daarvan maken gebruik van een schuifregister, maar die had ik toevallig niet voorradig. Daarom moest de schuiffunctie softwarematig worden gerealiseerd met een kleine microcontroller.

Maar wacht eens even – waarom een nieuwe interface aan beide zijden implementeren als er al standaardinterfaces bestaan? Dus besloot ik een UART te gebruiken voor deze verbinding. De onderste lagen van de interface zijn al gerealiseerd in de microcontroller-hardware, waardoor de rest van het project eenvoudig bleef.

Hardware

Figuur 1 toont het schema van deze seriële LCD-interface. Ik heb een 14-pins Microchip PIC16F1825-microcontroller van Microchip gebruikt, maar de goedkopere PIC16F1823 doet het ook. De meeste pinnen zijn nodig voor het aansturen van het display, maar er blijft één pin beschikbaar voor een extra functie (of zelfs twee indien laagspanning-programmeren is uitgeschakeld). Er zijn twee seriële ingangen beschikbaar: een die compatibel is met RS-232 en die bijvoorbeeld op een PC kan worden aangesloten (die heb ik gebruikt om de firmware te testen) en een UART/TTL-interface die kan worden gebruikt om de schakeling aan te sluiten op de microcontroller van het hostsysteem. In dat laatste geval kunnen T2 en omringende componenten worden weggelaten.

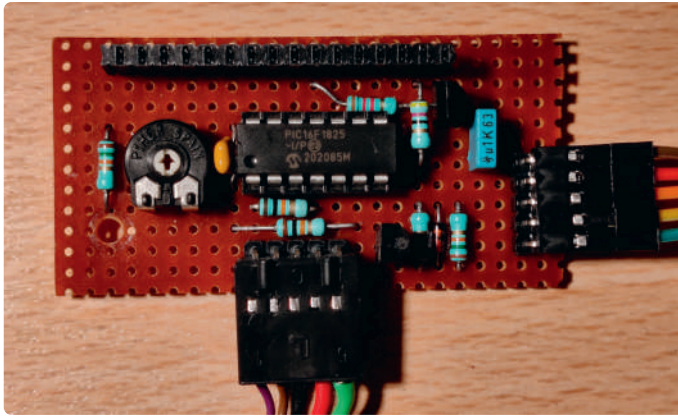


Figuur 1. Het schema van de eendraads LCD-interface.

De baudrate is slechts 9600; daarom kan de simpele opzet met T2 worden gebruikt om een RS-232 signaal om te zetten naar het logische niveau van de RC5-ingang van de microcontroller. De baudrate is ook laag gekozen om te voorkomen dat het hostsysteem wachtcycli moet toevoegen na bepaalde opdrachten.

T1 is toegevoegd om de achtergrondverlichting van het display te schakelen; trimmer P1 stelt het contrast in. K1 is voor de aansluiting van een Microchip Pickit-programmeerinterface maar kan worden weggelaten in de definitieve uitvoering.

Hoewel ik al een print had ontworpen, is de uiteindelijke schakeling gerealiseerd op gaatjesprint (**figuur 2**) omdat ik maar één exemplaar nodig had. De print is dan ook nooit geproduceerd en getest. De Eagle-bestanden kunnen echter worden gedownload van de projectpagina op Elektor Labs [1]; gebruik ervan is op eigen risico.



Figuur 2. Het ontwerp is gerealiseerd op gaatjesprint.

Software

Het programma is gecompileerd met de B. Knudsen CC5X-compiler [2]. De broncode voor deze interface en het HEX-bestand kunnen worden gedownload van de projectpagina op Elektor Labs [1].

Om het onderstaande te begrijpen, kunt u een kijkje nemen op enkele webpagina's over de HD44780 LCD-controller [3]. Om een byte naar het display te sturen moeten we achtereenvolgens twee nibbles (halve bytes) op de vier aansluitingen D4...D7 zetten.

Aanvankelijk was ik van plan om het display als een kleine VT52- of VT100-terminal te realiseren, maar dit bleek veel te veel werk voor zo'n simpel ontwerp. De uiteindelijke versie is dan ook beduidend minder 'intelligent'.

Er zijn drie modi geïmplementeerd in mijn firmware om data naar het display te verzenden: 4-bit, 8-bit data en 8-bit besturing.

In de 4bit-modus plaatst het programma eenvoudig de vier LSB's van elk ontvangen teken (bit 0...3) op de vier verbindingen naar het display, waarbij bit 4 wordt gebruikt om de RS-ingang aan te sturen (0 = control, 1 = data). Met de ASCII-tekenen 0x2...0x3f (' ' tot '?; zie voorbeelden) kan het display volledig worden aangestuurd. Voor het verzenden van één byte aan data naar het display hebben we twee nibbles nodig, dus twee 4bit-segmenten, die via de UART naar de micro worden gestuurd. Die wacht op de ontvangst van het tweede nibble voordat hij het eerste nibble gaat verwerken. Om te voorkomen dat de interface 'hangt' zit er een time-out op de ontvangst van dit tweede nibble: indien dit niet binnen één seconde wordt ontvangen, wordt de interface gereset.

In de 8bit-modi splitst de microcontroller de binnenkomende data in twee nibbles voor het LCD. Aangezien er geen extra bit beschikbaar is voor de RS-ingang, zijn er afzonderlijke 8bit-control- en 8bit-datamodi nodig. Voor het verzenden van een string wordt het LCD in 8bit-datamodus gezet en wordt de tekenreeks in de vorm van ASCII-tekenen verzonden. Het gebruik van 8bit-control is minder handig want is in veel gevallen niet efficiënter dan 4bit-control. De volgende control-codes zijn beschikbaar (gebruik deze codes dus niet voor tekens in de 8bit-datamodus!):

- > CTRL-A (0x01) achtergrondverlichting uit
- > CTRL-B (0x02) achtergrondverlichting aan
- > CTRL-C (0x03) 8 bit control
- > CTRL-D (0x04) 8 bit data (ASCII)
- > CTRL-E (0x05) wis het display
- > Escape (0x1b) 4 bit control en data

Dankzij het gebrek aan intelligentie kunnen bijna alle displays op basis van de HD44780 en daarmee compatibele controllers worden gebruikt. Informatie over het aantal regels en tekens dient door het hostsysteem te worden verzorgd. ◀

210498-03

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via trainer99@ziggo.nl of naar de redactie van Elektor via redactie@elektor.com.

Aansturing van het display: enkele voorbeelden

Wis het display, achtergrondverlichting aan en print "test 1" op regel 1 bij een 8bit-interface:

```
0x05,0x02,0x04,'test 1' (ctrl-e,ctrl-b,ctrl-d,'test 1')
```

Hetzelfde bij een 4bit-interface:

```
0x1b,0x20,0x21,0x02,0x37,0x34,0x36,0x35,0x37
,0x33,0x37,0x34,0x32,0x30,0x33,0x31 (Esc,'!',
ctrl-b,'746573742031')
```

Print "line 2" op de tweede regel van het display bij een 8bit-interface:

```
0x03,0xc0,0x04,'line 2'
```

Hetzelfde bij een 4bit-interface:

```
0x1b,0x2c,0x20,0x36,0x3c,0x36,0x39,0x36,0x3e,0x36,
0x35,0x32,0x30,0x33,0x32 (Esc,',6<696.652032')
```

↓ Project-download

www.elektormagazine.nl/summer-circuits-22



Gerelateerde producten

- > IIC/I²C Serial Interface Adapter Module (SKU 19332)
www.elektor.nl/19332
- > 2x16 Character LCD blue/white (SKU 17757)
www.elektor.nl/17757

WEBLINKS

[1] B Knudsen Data, CC5X-compiler: <https://www.bknd.com/cc5x/>

[2] Projectpagina op Elektor Labs: <https://www.elektormagazine.nl/labs/yet-another-1-wire-lcd>

[3] Wikipedia "Hitachi HD44780 LCD Controller": https://en.wikipedia.org/wiki/Hitachi_HD44780_LCD_controller