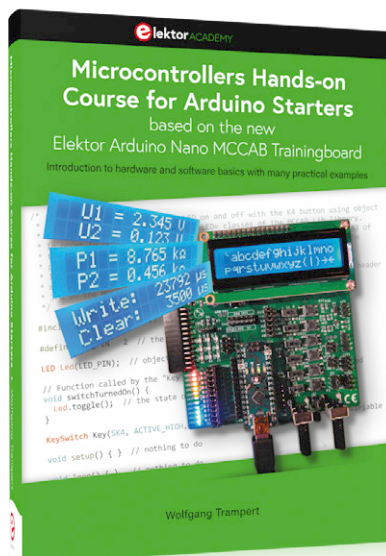




Het MCCAB Arduino Nano Training Board

alle hardware voor de
"Microcontrollers Hands-On Course"



Wolfgang Trampert (Duitsland) en Jan Buiting (Elektor)

Het in elkaar zetten van kleine Arduino-schakelingen is meestal een kwestie van componenten in een breadboard prikken en verbindingen maken met behulp van jumperkabeltjes. Maar al dat inprikken en puzzelen kost vaak meer tijd dan het schrijven van het programma. En bij de componentenopstelling voor bijvoorbeeld een looplicht met 11 LED's en net zoveel serieweerstanden kan gemakkelijk iets misgaan en dan ben je uren bezig met het debuggen van de hardware! Dan is een speciaal training-board, zoals Elektor's nieuwe "MCCAB", een stuk handiger, zeker als die met een mooi instructieboek wordt geleverd.

Opmerking van de redactie: dit artikel is een gedeelte uit het 455 pagina's tellende Elektor-boek *Microcontrollers Hands-on Course for Arduino Starters* (Elektor, 2023). Het is enigszins bewerkt en aangepast aan de stijl van Elektor Magazine. De auteur en de redactie hebben hun best gedaan op die bewerking en zijn graag bereid eventuele vragen te beantwoorden. Je vindt hun contactgegevens in het kader **Vragen of opmerkingen?**

De "MCCAB" (Microcontroller Crash Course for Arduino Beginners), een training-board met Arduino Nano, is door de auteur speciaal ontworpen en gefabriceerd voor Elektor als ondersteuning bij de *Microcontrollers Hands-on Course for Arduino Starters*, die wordt geleverd als een groene Elektor-gids. Het MCCAB-board, de gids (in het Engels of Duits) en een Arduino Nano zijn als bundel verkrijgbaar in de Elektor Store [1].

Op het board, die we van nu af aan MCCAB zullen noemen, is een uitgebreid assortiment van componenten al aangesloten op de ingeprikte microcontrollermodule, of kan met eenvoudige jumperkabels worden aangesloten. Dat betreft onder andere schakelaars /drukknoppen, LED's, potentiometers, buzzers, een LCD, interfaces, niveau-omzetters voor seriële verbindingen en drivers voor externe componenten. Ook complete externe modules kunnen via een connector met de MCCAB worden verbonden, of communiceren met de microcontroller op de kaart via de seriële interfaces. Dit bespaart je het moeizaam opbouwen van experimentele schakelingen, zodat je je kunt concentreren op de hoofdzaken: je eigen programma! Dus we laden de programma's van onze oefeningen in de ATmega328P-microcontroller op de MCCAB, waar ze dan in de vorm van programma-code draaien.

Enkele van de 32 pinnen van de ATmega328P-microcontroller worden gebruikt voor de voeding, voor

het aansluiten van een kristal voor de ingebouwde oscillator van de klokgenerator of als RESET-input voor de resetschakelaar op de MCCAB. (Als we die indrukken, wordt de microcontroller gereset en start het programma opnieuw.) Maar de meeste aansluitingen zijn beschikbaar als vrij te programmeren IO-lijnen, die de microcontroller kan gebruiken om met de buitenwereld te communiceren. Op de MCCAB zijn deze pinnen, op enkele uitzonderingen na, beschikbaar op connectorstrips.

In **figuur 1** zie je het schema van de MCCAB met kleur-gecodeerde functieblokken. De handleiding voor de MCCAB, met een uitgebreide beschrijving van alle componenten, kan gratis worden gedownload van de Elektor-website [2].

De functieblokken van de MCCAB

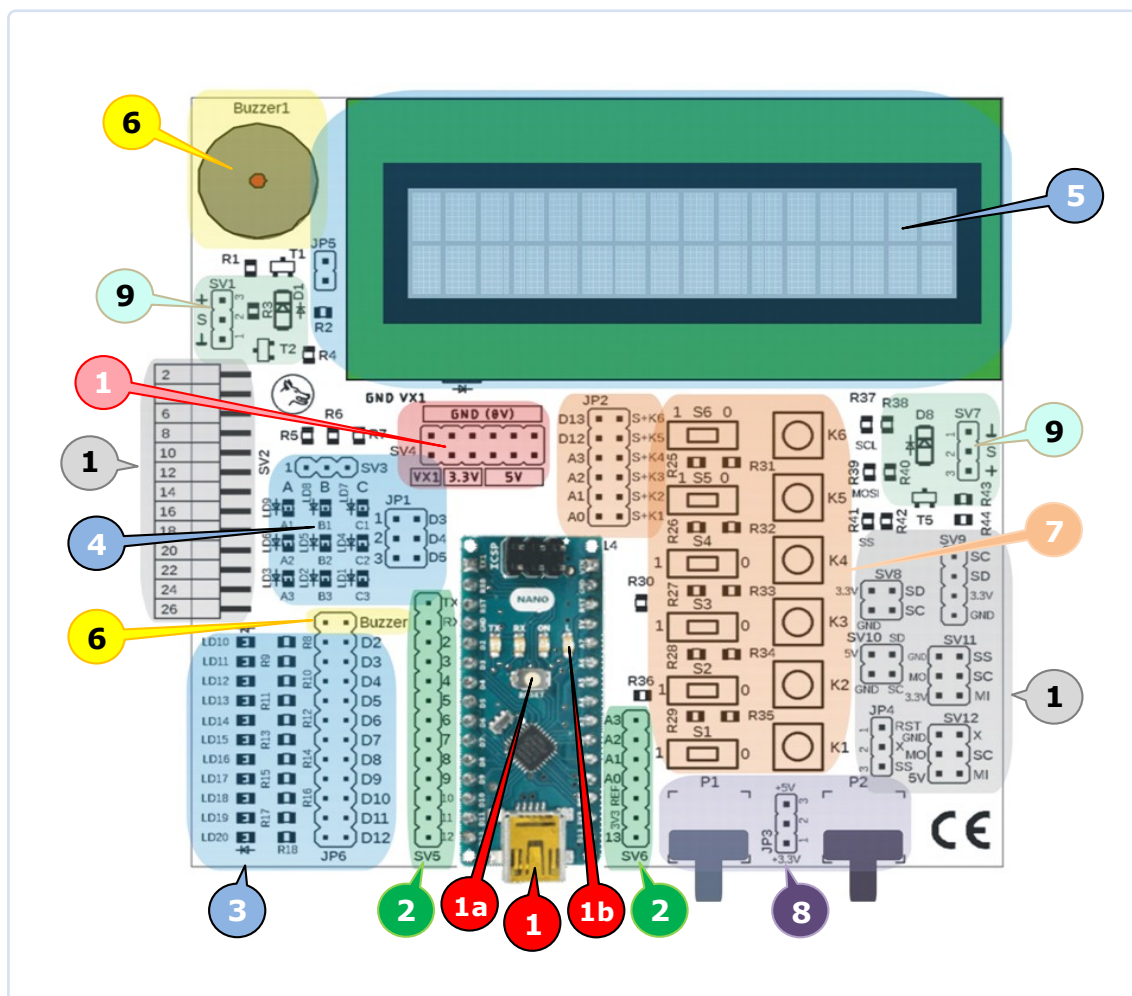
In figuur 1 kunnen we de volgende functieblokken en modules onderscheiden op de MCCAB.

(1) **Arduino Nano-microcontrollermodule** met RESET-knop (pijl 1a), LED L (pijl 1b) en miniUSB-connector voor aansluiting op je PC.

(2) Pinheaders SV5 en SV6 voor de **microcontroller-ingangen en -uitgangen**. Via deze twee pinheaders kunnen de GPIO's (General Purpose Inputs/Outputs) van de microcontroller met behulp van jumperkabels worden verbonden met de interne componenten op de kaart of met externe modules.

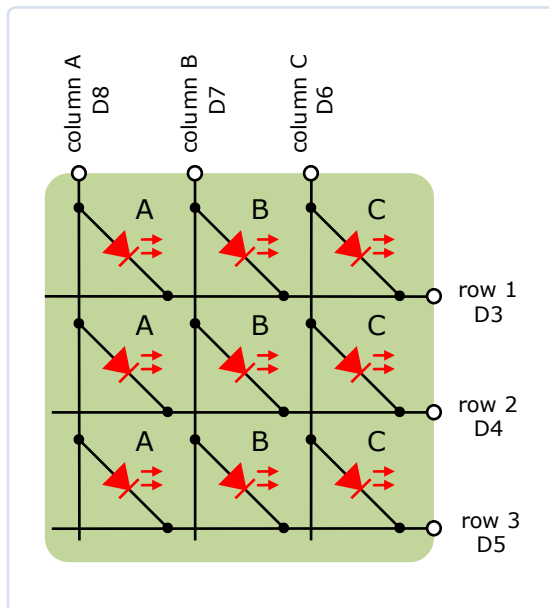
(3) **11 LED's**, LD10...LD20 (statusindicatoren voor de ingangen/uitgangen D2...D12 van de microcontroller). Elke LED kan met de toegewezen GPIO, D2...D12, worden verbonden met een jumper op header JP6.

(4) **3x3 LED-matrix** LD1...LD9 (negen rode LED's). De kolommen zijn permanent verbonden met uitgangen D6, D7 en D8 van de microcontroller, en de rijen kunnen met GPIO's D3, D4 en D5 worden verbonden met behulp van de jumpers op connector JP1.



Figuur 1. De functieblokken van de 'MCCAB' – het Elektor Arduino Nano Training Board.

Figuur 2. 3x3 LED-matrix.



Als de rijen van 3x3 LED-matrix zijn verbonden met de GPIO's D3, D4 en D5 via jumpers op JP1 of via jumperkabels met andere GPIO's, mogen die pinnen, evenals de kolomaansluitingen D6, D7 en D8, nooit worden gebruikt voor andere taken in een sketch. Dubbele toewijzing van de matrix-GPIO's zou kunnen leiden tot storingen of zelfs schade aan de MCCAB!

(5) **LC-display** met 2x16 tekens, via de I²C-bus verbonden met pinnen A4 en A5 van de microcontroller. Door JP5 te verwijderen kan voeding van het LCD worden uitgeschakeld voor oefeningen of experimenten waarbij het display niet wordt gebruikt.

(6) **Piezo-elektrische buzzer** Buzzer1 kan met GPIO D9 worden verbonden via een jumper op de positie 'Buzzer' van pinheader JP6.

(7) **6 schuifschakelaars S1...S6**, parallel geschakeld met **6 drukknoppen K1...K6**. Ze kunnen worden verbonden met de microcontroller-ingangen A0...A3 en D12...D13 via jumpers op pinheader JP2.

(8) **Potentiometers P1 en P2**, waarvan de lopers zijn verbonden met de analoge ingangen A6 en A7 van de microcontroller. De voedingsspanning van 3,3 V of 5 V kan op de potentiometers worden gezet via pinheader JP3.



Let op: de pinnen A6 en A7 van de ATmega328P zijn altijd geconfigureerd als analoge ingangen vanwege de interne architectuur van de chip. Het is niet toegestaan ze met `pinMode()` te configureren. Doe je dat toch, dan kan dat leiden tot een onjuist gedrag van het programma.

(9) Pinheaders SV1 en SV7 zijn **schakeluitgangen voor externe apparaten**.



(10) Pinheaders voor seriële verbinding met **externe SPI- en I²C-modules**.

(11) Connectorstrip SV2 met 2x13 pinnen voor het **aansluiten van externe modules**.

(12) Pinheader SV4 is het **verdeelpunt voor de voedingsspanningen van het board**. De spanningen kunnen met behulp van jumperkabels worden verbonden met interne componenten op het training-board of met externe modules.

De twee wat ingewikkelder functieblokken, de 3x3 LED-matrix (in figuur 1) en het LCD (ook in figuur 1) worden verderop nog in meer detail besproken. Zie voor alle andere de gedetailleerde beschrijving in het document *MCCAB Operating Instructions* [2].



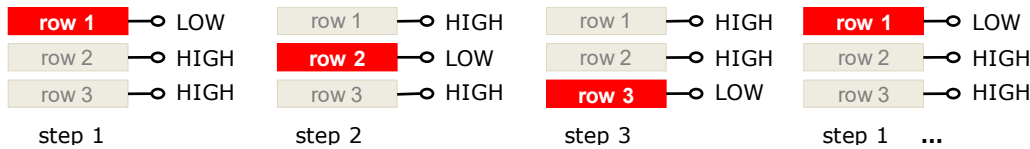
De *MCCAB_Lib*-bibliotheek is beschikbaar voor het aansturen van de 3x3 LED-matrix, de LED's LD10...LD20, de drukknoppen K1...K6, de schakelaars S1...S6 en Buzzer1. Wanneer je een MCCAB bezit, kun je de bibliotheek gratis downloaden en integreren in de Arduino IDE.

Voor het besturen van het LC-display gebruiken we de bibliotheek *LiquidCrystal_I2C*, die gratis kan worden gedownload van het internet en in de Arduino IDE kan worden 'ingevoegd'.

De 3x3 LED-matrix

De MCCAB bevat negen LED's die in een matrix zijn gerangschikt (zie figuur 1). In **figuur 2** zie je het principeschema.

De matrix bestaat uit drie rijen en drie kolommen. De kolommen heten A, B en C en de rijen zijn genummerd 1, 2 en 3. Op elk van de negen kruisingen van rijen en kolommen is een LED aangesloten. De anode is verbonden met een kolomlijn en de kathode met een rijlijn. De namen van de negen LED's corresponderen met hun positie: zo is LED 'B2' verbonden met kolom B en rij 2. Om een LED te laten oplichten, moet zijn kolomlijn logisch 1 zijn (+5 V) en zijn rijlijn logisch 0 (0 V). Volgens de handleiding voor de MCCAB zijn de kolomlijnen permanent verbonden met de GPIO's D6...D8 van de microcontroller. De rijlijnen kunnen met GPIO's D3...D5 worden verbonden via jumpers op header JP1. Als de rijlijnen van de matrix zijn verbonden met de GPIO's D3, D4 en D5 via jumpers op pinheader JP1, of met andere GPIO's van de microcontroller met behulp van jumperkabeltjes, mogen de rijlijnen en kolomlijnen D6...D8 niet worden gebruikt voor andere taken in een sketch. Dat zou kunnen leiden tot storingen of, in het ergste geval, zelfs de MCCAB kunnen beschadigen! **Als de matrix niet wordt gebruikt in een sketch, moeten de jumpers op MCCAB-header JP1 worden verwijderd.**



Figuur 3.
Achtereenvolgens
activeren van de drie
rijen van de matrix.

Voordelen van de matrixopstelling

We kennen LED-matrices bijvoorbeeld van reclameborden in voetbalstadions, waar gekleurde high power-LED's worden gebruikt in een matrixopstelling om schijnbaar bewegende afbeeldingen te genereren. Als we negen LED's individueel zouden aansturen, zijn daar negen GPIO's van de microcontroller voor nodig. Door de LED's in een matrix aan te sluiten, hebben we genoeg aan zes GPIO's. Hoe meer LED's we willen besturen, hoe groter het voordeel van matrixbesturing: Met een matrix van acht rijen en acht kolommen kunnen we 64 LED's aansturen. Bij individueel aanstuurde LED's zouden we 48 GPIO's méér nodig hebben!

De matrix aansturen in multiplexmodus

In figuur 2 zien we dat elk van de van de kolomlijnen en rijlijnen van de MCCAB is verbonden met drie LED's. We kunnen ze dus niet op de normale manier aansturen, want dan zouden er meerdere LED's tegelijk oplichten. In plaats daarvan is er op elk moment slechts één rijlijn actief (zie **figuur 3**), terwijl op dat moment de kolomlijnen actief zijn voor het bitpatroon op die rij. De beide andere rijlijnen moeten op dat moment open zijn, of een logisch-1-niveau hebben, zodat daar geen stroom kan lopen.

Als de drie rijlijnen snel genoeg om beurten worden geactiveerd, zoals in figuur 3, zien we door de traagheid van het menselijke oog, een statisch beeld van alle negen LED's. Het gebruikersprogramma bestuurt de LED-matrix in een eindeloze lus, waarin steeds één van de drie rijen 1, 2 of 3 logisch 0 wordt gemaakt, terwijl de andere twee logisch 1 zijn. De kolomaansluitingen van alle LED's die moeten oplichten in de op dat moment actieve rij worden logisch 1 gemaakt.

De LED's die uit moeten zijn in de actieve rij krijgen een logische 0 op hun kolomaansluiting.

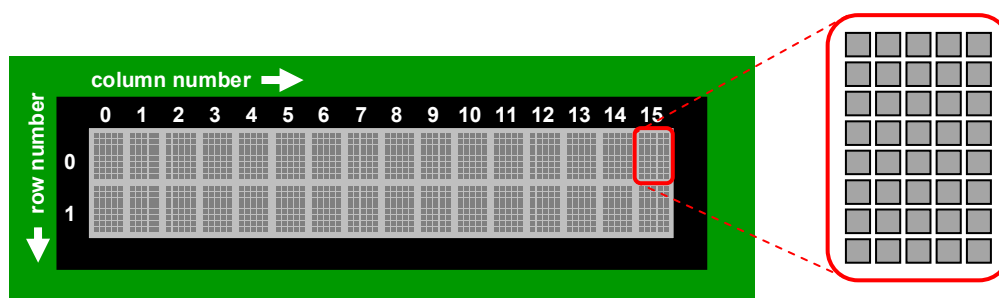
Om bijvoorbeeld de LED's A3 en C3 te laten oplichten moet rij 3 laag zijn en moeten kolommen A en C hoog zijn, terwijl rij 1 en rij 2 hoog zijn en kolom B laag.

Het Liquid Crystal Display

De Elektor MCCAB is uitgerust met een LC-display (LCD) waarmee uitvoer van tekst, numerieke waarden en ook zelf-gedefinieerde bijzondere tekens mogelijk is. Het display heeft twee regels van 16 karakters. Elk karakter wordt gevormd uit de punten van een 5×8-matrix, zoals te zien is in **figuur 4**. De benodigde 5×8-bitpatronen voor elk individueel karakter zijn opgeslagen in de LCD-controller conform de ASCII-tabel.

ASCII is een 7-bits code, terwijl onze ATmega328P microcontroller een '8bit-machine' is. Hij kan dus één byte tegelijk verwerken en slaat zijn data op in byteformaat. Het achtste bit moet altijd 0 zijn bij het opslaan van ASCII-codes. Maar de ontwikkelaars van de HD44780-displaycontroller (de standaard in deze categorie en te vinden op bijna alle LCD-modules) wilden dit achtste bit van het byte niet onbenut laten en dus hebben ze de ASCII-karakterset uitgebreid met 128 extra karaktercodes (waarbij het achtste bit logisch 1 is). Daarom gebruikt ons LCD ook de karaktercodes 128 tot 255 (bijna allemaal).

Naast de 95 afdrukbare tekens (codes 32...126) kent ASCII ook 33 besturingstekens, namelijk 0...31 en 127, die 'niet afdrukbaar' zijn. Omdat deze tekens bedoeld zijn voor besturingsdoeleinden, geeft het display ze niet weer. De HD44780-displaycontroller is verkrijgbaar in twee versies: de ene met de A00 ROM-code en de andere met de A02 ROM-code. Versie A00 geeft



Figuur 4. Opbouw van
het LC-display op de
MCCAB met zijn 16×2
weer te geven tekens.

Tabel 1. Indeling van het karaktergeheugen van de twee HD44780-varianten.

Karaktercode	ROM Code A00	ROM Code A02
0...7	gebruikergedefinieerde tekens	gebruikergedefinieerde tekens
8...15	gebruikergedefinieerde tekens	gebruikergedefinieerde tekens
16...31	(geen weergave)	zichtbare tekens
32...127	zichtbare tekens	zichtbare tekens
128...159	(geen weergave)	zichtbare tekens
160...255	zichtbare tekens	zichtbare tekens

blanco tekens voor de codes 16...31 en 128...159. Versie A02 geeft bijzondere tekens weer voor deze codes (zie **tabel 1**). Welke van de twee ROM-versies is geïnstalleerd in de MCCAB is niet te voorspellen; de fabrikant van de LCD-module maakt deze keuze.

De gebruiker kan maximaal acht bijzondere tekens definiëren en op het LCD weergeven. De ASCII-codes 0 tot 7 zijn toegewezen aan deze acht speciale tekens. Optioneel zijn dezelfde acht tekens ook te adresseren via de ASCII-codes 8...15.

In figuur 4 zie je ook de nummering van de rijen en kolommen van het display. In beide gevallen beginnen we bij 0 te tellen. Met behulp van van deze coördinaten kunnen tekens op een specifieke positie op het display worden geschreven. Het LCD heeft een cursor die aangeeft waar het volgende karakter wordt geschreven. De in het boek beschreven LCD-library bevat methoden voor het positioneren van de cursor. Hoewel het display slechts 16 tekens per regel kan

tonen, zijn er in het geheugen van de displaycontroller 40 geheugenlocaties voor elke regel. Zoals je ziet in **figuur 5** zit er een gat van 24 geheugenplaatsen tussen het laatste displayadres (39) van de eerste regel en het startadres (64) van de tweede regel.

Met behulp van *shift*-functies in de LCD-library kan het in de display zichtbare gedeelte van het geheugengebied als een venster worden verschoven over het gehele displaygeheugen.

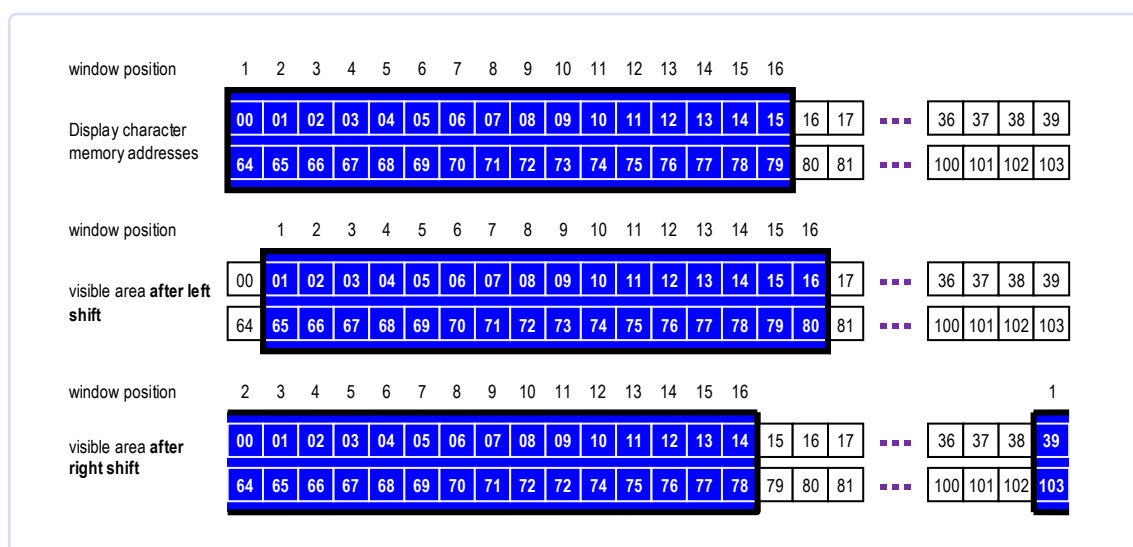
Figuur 5 bestaat uit drie gedeelten:

- boven: het zichtbare venster van het display zonder eerdere schuifoperaties;
- midden: de inhoud van het display na één keer schuiven naar links;
- onder: de inhoud van het display na één keer schuiven naar rechts.

Contrastregeling van het LCD

De MCCAB wordt geleverd met een niet-afgesteld contrast voor het LCD. Daarom en omdat het contrast van het display kan veranderen door omgevingsinvloeden (zoals temperatuur) of veroudering, heeft het LCD een instelpotentiometer voor het instellen van het contrast. Deze zit aan de onderkant van het board en is gemarkeerd met een pijl. Stel het contrast in met een kleine schroevendraaier bij de eerste keer opstarten of wanneer nodig.

i Hint: als er de eerste keer dat je het LCD gebruikt geen tekens zichtbaar zijn op het display hoewel dat wel het geval zou moeten zijn, dan ligt dat waarschijnlijk aan een verkeerde LCD-contrastinstelling!



Figuur 5. De op het display zichtbare inhoud voor en na verschuiving.

Data van de microcontroller naar het display sturen

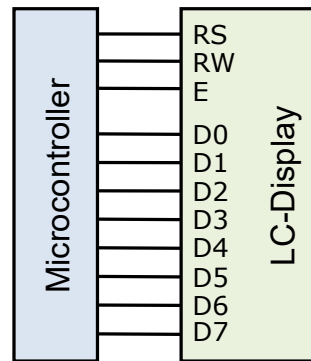
Op elke LCD module zit een HD44780-displaycontroller die de ASCII-codes van de door de microcontroller verzonden tekens ontvangt via een interface, de corresponderende tekens van de eerdergenoemde 5x8-puntmatrix genereert en die weergeeft.

De LCD-controller van het type HD44780 heeft alleen een parallelle interface voor het ontvangen van de weer te geven data. De microcontroller schrijft steeds een databyte bestaande uit acht bits, Do...D7, in het ingangsregister van de displaycontroller met behulp van de drie stuursignalen RS, RW en E (zie **figuur 6**). Deze manier van parallelle data-overdracht zou een groot deel van de resources van de MCCAB in beslag nemen, omdat het LCD alleen al 11 van de 16 beschikbare pinnen van de ATmega328P nodig zou hebben! De tweede manier voor data-overdracht naar de HD44780-controller, waarbij de acht databits worden geschreven in twee pakketjes van elk vier bit, helpt ook niet veel omdat ook in dat geval, nog steeds zeven pinnen van de microcontroller worden gebruikt door het LCD (omdat ook dan de drie stuursignalen, RS, RW en E nodig zijn).

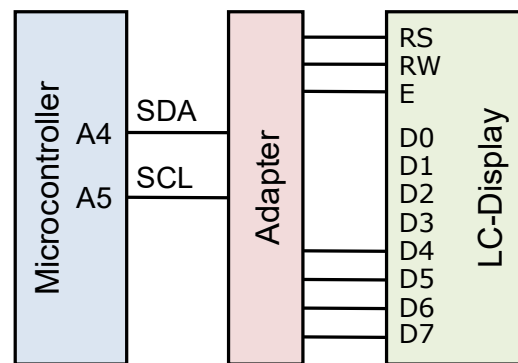
Daarom wordt de LCD-module op de MCCAB aangestuurd via de I²C-bus, een synchrone seriële bus die met slechts één datalijn (SDA) en één kloklijn (SCL) werkt en de data bit voor bit verzendt. Het dataverkeer via de I²C-interface wordt verzorgd door twee lijnen van de ATmega328P-microcontroller: A4 (SDA) en A5 (SCL) (zie **figuur 7**).

Een extra adapter aan de onderzijde van de LCD-module zet de I²C-signalen om in parallelle signalen. Deze adapter gebruikt de eerdergenoemde methode van data-overdracht in twee groepjes van vier bits via de datalijnen D4...D7. Hij stuurt dus eerst de databits D4...D7 en dan de databits Do...D3 via dezelfde lijnen D4...D7. Omdat de lijnen A4 en A5 van de microcontroller op de MCCAB toch al zijn gereserveerd voor de I²C-interface, zijn er geen extra resources nodig voor het aansturen van het display; er kunnen in principe meerdere deelnemers tegelijk worden verbonden met de I²C-bus. Omdat elke deelnemer op de bus een eigen I²C-adres heeft, 'ziet' hij alleen data wanneer die bestemd is voor zijn eigen adres. Het LCD op de MCCAB zit meestal op I²C-adres 0x27. Als de fabrikant een ander adres heeft gekozen, staat dat vermeld op het display. ◀

vertaling Evelien Snel — 230215-03



Figuur 6. LCD-aansturing met acht databits en drie stuurlijnen.



Figuur 7. LCD-aansturing via de I²C-bus in plaats van parallel (vergelijk met figuur 6).

Vragen of opmerkingen?

Hebt u technische vragen naar aanleiding van dit artikel? Stuur een e-mail naar de redactie van Elektor via redactie@elektor.com.



Gerelateerde producten

- **Microcontrollers Hands-on Course for Arduino Starters (Bundle) (SKU 20440)**
 - 1 x Guide: *Microcontrollers Hands-On Course for Arduino Starters*
 - 1 x MCCAB Arduino Training Board
 - 1 x Arduino Nano
- www.elektor.nl/20440

WEBLINKS

- [1] Elektor Arduino Nano Training Board MCCAB en Companion Guide: <http://www.elektor.nl/20440>
- [2] Handleiding voor het Elektor Arduino Nano Training Board MCCAB: <http://www.elektor.nl/20440>