

## Labo datacommunication les 3 : LCD

### Inleiding

In deze les wordt het protocol van de HD44780 controller bestudeerd. Die controller kan verschillende types van LCD displays aansturen.

### Doelstelling

Een bestaand protocol kunnen begrijpen.

De volgorde van de elektrische signalen die bepaald worden door het protocol kunnen vertalen naar een programma.

Als de stuursignalen wijzigen van pinnummering moet je het programma kunnen aanpassen.

Een datasheet kunnen begrijpen.

### LCD pinnummering

Pinnummer 1 en 2 zijn de voeding van het LCD display.

Pinnr. 3 regelt het contrast van het display.

Pinnr. 4,5 en 6 zijn de controle lijnen.

Pinnr. 7 t.e.m.14 vormen de databyte die we naar het LCD display kunnen versturen.

### Pinnummering USB data-acquisitiekaart

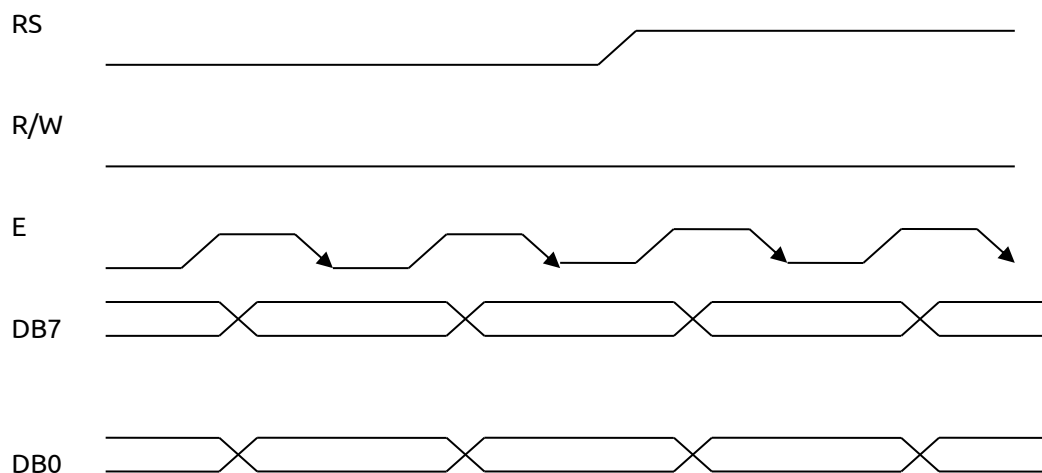
Poort D (8 digitale uitgangen) van de microchip verbinden we met de 8 datalijnen van het LCD display. Om de controle lijnen van het LCD display aan te sturen hebben we nog 3 extra digitale uitgangen nodig. Die drie extra digitale uitgangen kunnen we ook terug vinden op poort D van de microchip. Alles is aangesloten volgens onderstaand schema.

Pinnr. LCD			Aanduiding	Adressering microchip
1	Vss	-	Power supply (GND)	
2	Vcc	-	Power supply (+5V)	
3	Vee	-	Contrast adjust	trimmer
4	RS	0/1	0 = Instruction input 1 = Data input	Poort D bit 10
5	R/W	0/1	0 = Write to LCD module 1 = Read from LCD module	Poort D bit 9
6	E	1, 1-- >0	Enable signal	Poort D bit 8

7	DB0	0/1	Data bus line 0 (LSB)	Poort D bit 0
8	DB1	0/1	Data bus line 1	Poort D bit 1
9	DB2	0/1	Data bus line 2	Poort D bit 2
10	DB3	0/1	Data bus line 3	Poort D bit 3
11	DB4	0/1	Data bus line 4	Poort D bit 4
12	DB5	0/1	Data bus line 5	Poort D bit 5
13	DB6	0/1	Data bus line 6	Poort D bit 6
14	DB7	0/1	Data bus line 7 (MSB)	Poort D bit 7
15	Anode		+5V	
16	Kathode		GND	

## Aansturen LCD displaycontroller (HD44780/KS0066)

Onderstaande grafieken kunnen we terugvinden op de datasheets van het display. Het enable signaal E kunnen we vergelijken met het kloksignaal SCL van I<sup>2</sup>C. Het enable signaal bepaalt wanneer de displaycontroller de data gaat verwerken. In tegenstelling tot I<sup>2</sup>C wordt de data nu verwerkt tijdens de dalende flank. Per klokpuls worden er nu ook 8 bits data verwerkt.



De volgorde van de signalen is ook zeer belangrijk. We brengen eerst het enable signaal naar een hoog spanningsniveau, vervolgens zetten we de data klaar.

Als we nu het enable signaal terug naar 0V brengen wordt de data door de displaycontroller verwerkt. Via een wait instructie moet de displaycontroller wat extra tijd krijgen om de data te verwerken.

Samengevat moet ons programma er als volgt uitzien :

=> E = 5V => Data klaarzetten => E = 0V => wait=> E = 5V => ....

Er zijn nu echter nog twee controle lijnen:

- Het controle signaal R/W bepaalt of we data lezen of schrijven naar het display. Een laag signaal is schrijven, een hoog signaal is lezen. Wij beperken ons tot het schrijven van data.
- Tenslotte hebben we nog het register select signaal (RS), dat signaal bepaalt of er een instructie of een karakter gestuurd wordt. De cursor verplaatsen, de vorm van de cursor,... zijn voorbeelden van instructies. Als het RS signaal een waarde van 0V heeft sturen we instructies naar de displaycontroller, bij een hoog signaal sturen we karakters naar de displaycontroller.

## Programma

Aangezien we ons beperken tot het schrijven naar het LCD display, hebben we 4 verschillende combinatie van de controlesignalen. Het signaal R/W blijft immers altijd op een laag spanningsniveau staan.

We gaan dus de volgende 4 subroutines gaan programmeren :

Let op : de controlelijnen zitten op de tweede byte van poort D

EHoogInstructie (E = 1, RS = 0, RW = 0)

ELaagInstructie (E = 0, RS = 0, RW = 0)

EHoogData (E = 1, RS = 1, RW = 0)

ELaagData (E = 0, RS = 1, RW = 0)

### B.v. Routine EHoogInstructie()

Met de EHoogInstructie() routine willen we de controller klaarzetten om een instructie te verwerken. De controlesignalen (aan de LCD kant) moeten dus de volgende waarden aannemen :

E = 1 => 5V     bit 8

RW = 0 => 0V     bit 9

RS = 0 => 0V     bit 10

Die controlesignalen staan op de tweede byte van poort D. Ze zijn aangesloten op bit 8, 9 en 10. We moeten er dus voor zorgen dat we de bits 0 t.e.m. 7 van poort D ongewijzigd laten als we onze stuurlijnen klaar zetten.

X	X	X	X	X	0	0	1	X	X	X	X	X	X	X	X
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

RS RW E

Om de andere bits van het stuurregister niet te beïnvloeden gaan we opnieuw gebruikt maken van booleaanse operatoren.

X = readDigitalOutPortD()	xxxx	xxxx	xxxx	xxxx
Y = X & 0xF8 FF (stuurlijnen op nul)	<u>1111</u>	<u>1000</u>	<u>1111</u>	<u>1111</u>
	xxxx	x000	xxxx	xxxx
Z = Y   0x01 00	<u>0000</u>	<u>0001</u>	<u>0000</u>	<u>0000</u>
writeDigitalPortD(Z)	xxxx	x001	xxxx	xxxx

## WriteDataLijnenLCD

Nu moeten we de datalijnen van het LCD display kunnen aanspreken zonder de stuurlijnen te beïnvloeden. Daarvoor moet je zoals hierboven te werk.

We lezen eerst de data van poort D in. Vervolgens plaats je de 8 bits van de 8 datalijnen op nul. Vervolgens geef je de gewenste data mee.

## Enkele basisinstructie

In de onderstaande tabel vind je de belangrijkste instructies terug om het display te initialiseren. In de datasheets vind je de volledige lijst terug.

Aan de eerste, meest beduidende bit herkent de controller de instructie, met de volgende bits kunnen we dan verschillende opties meegeven of een cursorpositie weergeven.

Om het display te initialiseren moeten we met function set het type van het display meegeven. Ons display beschikt over 2 lijnen en karakters van 5x7 pixels. De datalengte mag je instellen op 8 bit.

Eventueel kunnen we ook nog de cursor naar zijn home positie brengen.

	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear display & Cursor home	0	0	0	0	0	0	0	0	0	1
Cursor home	0	0	0	0	0	0	0	0	1	*
Display on Sets On/Off of all display (D), cursor On/Off (C) and blink of cursor position character (B).	0	0	0	0	0	0	1	D I=ON 0=off	C I=ON 0=off	B I=ON 0=off
Function set Sets interface data length (DL), number of display line (N) and character font(F).	0	0	0	0	1	DL I=8bit 0=4bit	N I=2lines 0=1line	F 0:5x7 I:5x10	*	*
Sets DDRAM	0	0	1	DDRAM adres						

De adressen van de cursorposities kunnen we uit de onderstaande tabel halen.

Ddram adres

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Als we b.v. de function set willen instellen, herkent de controller de functie omdat we bit 5 (DB5) hoogzetten. Met DB2,3,4 kunnen we dan het type van ons display bepalen. Op de plaats van het sterretje mogen we om het even wat instellen.

## Voorbeeld clear display & cursor home instructie

```
EHoogInstructie();
WriteDataLijnenLCD(0x01);
ELaagInstructie();
Wait(1);
```

## Voorbeeld 1 char doorsturen

Om tekst door te sturen naar de LCD controller moeten we gebruik maken van de routines EHoogData() en ELaagData().

```
EHoogData()
WriteDataLijnenLCD (char)
ELaagData()
Wait(1)
```

## Voordeel Hexadecimaal rekenen aantonen (Per 4 bits)

Het grote voordeel van het hexadecimale rekenen is dat je bijna geen rekenwerk hebt bij het omzetten van een binair getal naar een hexadecimaal getal. Eén hexadecimaal cijfer komt immers overeen met 4 bits.

8 4 2 1

1 1 1 1  $\Rightarrow F = 15$

1 1 1 0 => E = 14

1 1 0 1  $\Rightarrow D = 13$

1 1 0 0  $\Rightarrow C = 12$

1 0 1 1  $\Rightarrow$  B = 11

1 0 1 0  $\Rightarrow A = 10$

$$1001 \Rightarrow 9 = 9$$
$$1\ 0\ 0\ 0 \Rightarrow 8 = 8$$

0 1 1 1 => 7 = 7

B.v.	1111 0100 0011 1101
&H	F 4 3 D

[illegible]

## Opdracht week 3

Deze week wordt er een nieuw project opgestart. Het project wordt op het einde van de les ingediend. Zip of rar het project en dien in met de volgende naam : Groepsnr-loginnaam1-LCD-03

- Maak de 4 routines EHoogData, ELaagData, EHoogInstructie en ELaagInstructie.
- Plaats een button init op de Gui. Na het drukken op de button moet het display geïnitieerd worden, de eventuele tekst wordt gewist en de cursor staat te knipperen op de homepositie.
- Voeg ook een tekstvak en button 'schrijf tekst' toe aan de gui. Na het drukken op de button schrijf wordt de tekst in het tekstvak op de LCD display geplaatst. Zorg ervoor dat het 17<sup>e</sup> karakter automatisch op de tweede lijn verschijnt.
- Voorzie een menu in het programmaatje waarin we enkele opties van het display kunnen instellen. Het display aan en afzetten. De cursor al of niet laten knipperen. De cursor al of niet zichtbaar maken.
- Als er meer dan 32 karakters in het tekstvak zitten begint de tekst op het lcd display automatisch te scrollen. Gebruik hiervoor een instructie van de display controller.

## 4 bits mode

### Inleiding

Soms beschikken we maar over 8 digitale uitgangen en moet er toch een LCD display aangestuurd worden. Dit kan perfect als we het LCD display in de stand 4 bits mode plaatsen. We kunnen dan 4 digitale uitgangen gebruiken om data door te sturen en 3 digitale lijnen worden er gebruikt om de stuurlijnen te bedienen.

Op het onderstaande schema zien we dat er in de 4 bits mode met de 4 meest significante bits wordt gewerkt. Gebruik de shiftoperator << om meest beduidende bits te selecteren.

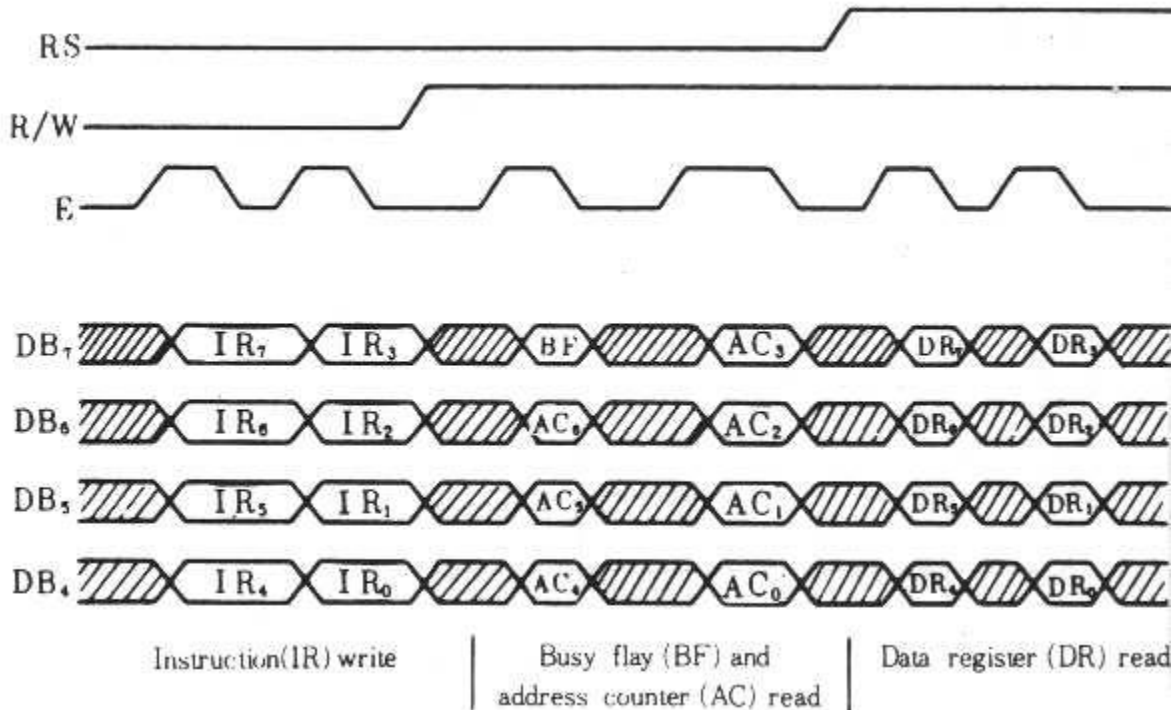
De vier minst significante bits hebben we enkel nodig voor de instructie 'function set'. **Die instructie moet dan ook als eerste uitgevoerd worden.**

De instructie "function set " is de enige instructie die we in één keer doorklokken.

Voor die ene instructie moet je de 8 bits van poort D gebruiken. Na het instellen van een datalengte van 4 bits via function set, moeten alle instructies en data in 2 keer doorgeklokt worden. Wie effectief digitale uitgangen wil uitsparen kan die 4 bits hardware matig aansluiten op de lcd controller m.a.w. die 4 bits krijgen een vaste 0V of 5V. Die 4 minst significante bits van de LCD controller wijzigen toch nooit in de 4 bits mode. Wie slechts over 8 digitale uitgangen beschikt kan dus toch het LCD display aansturen. (4 datalijnen + 3 controlelijnen).

Er rest ons nog één probleem. Alle data en instructies bestaan uit 1 byte, in de vier bits mode kunnen we maar een halve byte per klokslag doorsturen. Eén byte zullen we dus in 2 klokpulsen moeten doorklokken. Op het schema is duidelijk te zien dat we eerst de **4 meest significante bits** van onze byte moeten doorklokken en daarna de 4 minst beduidende bits. Beide nibbles (groep van 4 bits) moeten naar lijn DB7,DB6, DB5 en DB4 gestuurd worden.

Schema = voorbeeld van een datatransfer in de 4-bits mode.



## Opdracht 3b

- Voeg twee radiobuttons toe zodat je de keuze kan maken tussen 4 bits en 8 bits modus.
- Let op : als je van modus wilt veranderen moet je de USB kabel eens verwijderen zodat het LCD display volledig gereset wordt.
- Nu pas je het programma aan zodat je de display kan aansturen in de 4 bits mode. Programmeer eerst de instructie 'function set'. Voor die ene instructie moet je de 8 bits van poort D gebruiken. Na het instellen van een datalengte van 4 bits via function set, moeten alle instructies en data in 2 keer doorgeklokt worden. Let op : eerst de meest beduidende bits doorklokken, vanaf nu kan je enkel de 4 meest beduidende bits van poort D gebruiken.