Frederiksen
Passion for science

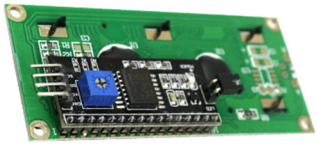
LCD-Display

Arduino-eksperiment	130315	*Stikord*	Biblioteker, 16x2 display	
Version	2018-06-21 / HS	*Niveau*	Videregående	*p.* 1/4

Det lærer du:

- Tilslut et LCD-display til Arduino
- Brug funktioner i eksterne biblioteker

Displays med 2 linjer à 16 karakterer findes i et væld af udgaver fra mange forskellige producenter. De går under betegnelser som 16x2 eller 1602 displays. Heldigvis er de stort set ens, hvad programmeringen angår. Du kan ikke være 100 % sikker på, at et tilfældigt displaymodul opfører sig som beskrevet nedenfor – men det er tæt på.

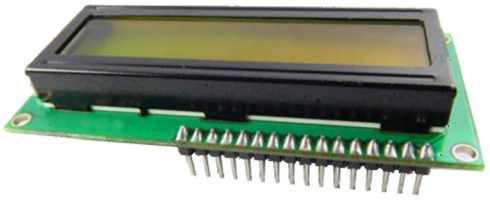


Det forudsætter dog, at displayet kommunikerer "parallelt" via de 16 ben langs displayets overkant (som på billedet i afsnit 1).

Er der monteret et hjælpeprint på undersiden (som vist til højre), skal der kommunikeres "serielt". Det sparer ben, men er en tand mere kompliceret og vil ikke blive gennemgået her.

1 – Lod en stiftrække på displayprintet

Hvis der allerede er monteret en stiftrække på displayet (som vist til højre) – så gå videre med afsnit 2.



Hvis du ikke er så trænet i lodning, er det måske bedst at få lidt hjælp fra en mere erfaren person. Det er en éngangsoperation at lodde stifterne på. Derefter forbinder vi displayet via breadboardet.

Stiftrækken leveres typisk med 40 stifter. Knæk forsigtigt rækken over, så du har en sammenhængende stift-række med 16 stifter.
De korte stumper af stifterne skal loddes på displayprintet.

Lodningen går nemmest med en hjælper, som kan holde stifterne vinkelret på displayprintet med en tang, mens der loddes. Lod først ben 1 og ben 16, og kontroller, at alting ser rigtigt ud, inden du lodder resten.

Uden en hjælper kan man sætte stifternes lange del i breadboardet langs den ene kant. Displayprintet lægges ovenpå, så den korte del af stifterne går gennem printet. *Men hvis du gør dette, skal du være **helt** fokuseret på ikke at varme for meget, når du lodder – ellers smelter du breadboardet. Lod først hver anden stift og hold pauser, så breadboardet kan køle af.*

2 – Forbind displayet til Arduino

Som du vil se nedenfor, er der en meget stor grad af fleksibilitet mht. hvilke ben *på Arduinoen*, der skal bruges – du kan vælge efter, hvordan din opstilling lige vender, og efter hvilke ben, der er reserveret på forhånd. Det betaler sig at lave f.eks. et regneark, hvor du noterer forbindelserne.

På displayet skal følgende ben bruges:

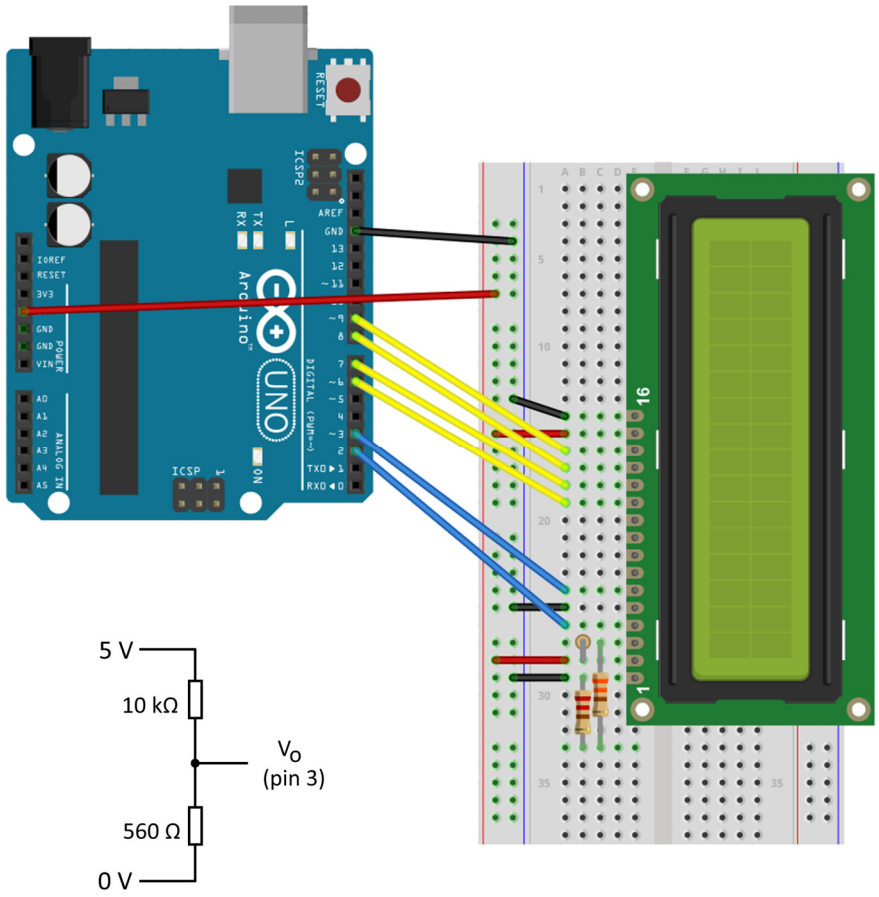
1	VSS	0 V (GND)
2	VDD	+ 5 V
3	Vo	Kontrast – se nedenfor
4	RS	Styresignal
5	RW	0 V (GND)
6	E	Styresignal
11	D4	Databit 4
12	D5	Databit 5
13	D6	Databit 6
14	D7	Databit 7
15	+LED	+ 5 V
16	-LED	0 V (GND)

Som du kan se, springes ben 7 til 10 over, og en del af resten er faste spændinger. Den reelle kommunikation med Arduinoen foregår via følgende ben: RS, E, D4, D5, D6 og D7.

Ben 3 (kontrast) er lidt specielt. Det skal have en lav jævnspænding, som vi vil danne med en spændingsdeler. Nogle gange ser man et trimme-potentiometer her, så kontrasten kan justeres – vi vil nøjes med en fast indstilling.

Så er vi klar til at lave en konkret opstilling.

Det er klogt at koble Arduinoen fra USB-forbindelsen, så der ikke er spænding på opstillingen, mens du arbejder.



Vi bruger som før de to lange yderste rækker til 5 V og GND. De forbindes som vist: Display-ben 2 og 15 til 5 V, display-ben 1, 5 og 16 til GND.

Spændingsdeleren omkring ben 3 udføres således:

Monter en 10 kΩ modstand mellem display-benene 2 og 3 – den skal stå lodret, pga. den korte afstand.

Mellem display-ben 1 og 3 monteres en modstand på ca. 560 Ω. På tegningen er der vist, hvordan man kan sætte to modstande i serie: 220 Ω + 330 Ω = 550 Ω. Det er godt nok, hvis man ikke lige har 560 Ω i sin samling.

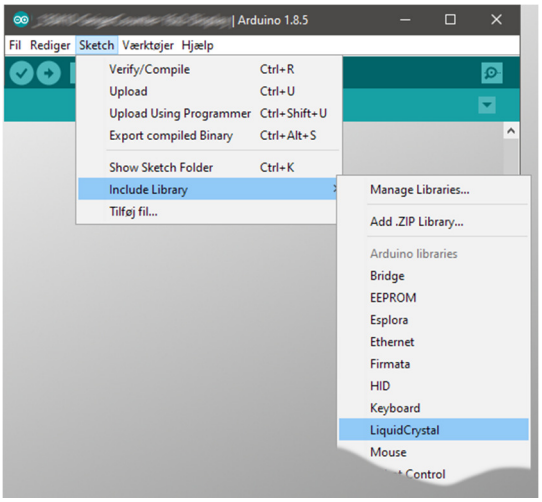
Signalledningerne forbindes efter denne ben-tabel:

Arduino	Display	Funktion
2	4	RS
3	6	E
6	11	D4
7	12	D5
8	13	D6
9	14	D7

Udfordring 1

Lav opstillingen som vist – og gå den omhyggeligt efter for fejl. Tilslut USB-kablet; det skal tænde for baggrundslyset.

3 – Biblioteket LiquidCrystal



Hidtil har vi anvendt en del af Arduino-systemets indbyggede funktioner. Styring af et LCD-display er dog en lidt mere speciel opgave, som ikke er med i den centrale kerne af systemet. Den slags kode placeres i såkaldte *biblioteker* (engelsk: *libraries*).

For at kunne bruge et bibliotek, skal det *inkluderes* i koden:

Start et nyt program og brug menuen *Sketch* / *Include Library*, og vælg *LiquidCrystal*.

Dette tilføjer følgende linje øverst i programmet:

```
#include <LiquidCrystal.h>
```

(Hvis man ved, hvad der skal stå, kan man også bare selv skrive dette uden at bruge menusystemet.)

Herefter minder situationen meget om brugen af `Serial`-objektet (som vi har brugt til at kommunikere med PC'en). De funktioner, vi skal bruge, ligger som objektets metoder og noteres med punktum mellem objektets navn og funktionen – som f.eks.:

```
Serial.println("Hej");
```

Objektet `Serial` eksisterer på forhånd. Derimod skal det `LiquidCrystal`-objekt, som vi skal bruge, først oprettes. (Se programstump øverst næste side.)

Det er under oprettelsen af objektet, at vi specificerer de ben, som skal anvendes.

Som altid betaler det sig at bruge konstanter med fornuftige navne fremfor bare at sætte tal ind. Alt i alt kommer det til at se således ud:

```
const int pinRS = 2;
const int pinE  = 3;
const int pinD4 = 6;
const int pinD5 = 7;
const int pinD6 = 8;
const int pinD7 = 9;

LiquidCrystal lcd(pinRS, pinE, pinD4, pinD5, pinD6, pinD7);
```

Bemærk her, at `LiquidCrystal` er objektets *type*, mens selve objektet er oprettet som `lcd`. Vi opretter `lcd` som en global variabel udenfor såvel `setup()` som `loop()`.

Nu "ved" `lcd`-objektet, hvilke ben, der skal signaleres på. LiquidCrystal-biblioteket er meget generelt, så vi skal også specificere display-typen. Det gøres ved at indsætte denne linje i `setup()`:

```
lcd.begin(16,2);
```

Herefter kan vi skrive tekst og tal ud – helt som vi kunne i monitorvinduet på PC'en. Dog bruges der ikke linjeskift. I stedes flyttes cursoren (indsættelsespunktet) hen, hvor teksten skal stå. Se eksemplet herunder:

```
lcd.setCursor(0,0);       // Helt til venstre, første linje
lcd.print("Hej verden!");
lcd.setCursor(0,1);       // Helt til venstre, anden linje
lcd.print("Tid:");
lcd.setCursor(7,1);       // Midt på anden linje
lcd.print(millis()/1000);
```

Udfordring 2

Skriv programmet færdigt. De seks sidst nævnte linjer hører til nede i `loop()`-funktionen.

Prøv af, om det virker: Tallet på nederste linje skal tælle én op pr. sekund.

4 – Lær displayet at kende

Der er en del flere metoder defineret i LiquidCrystal-biblioteket:

```
lcd.home();       // Som setCursor(0,0);
lcd.clear();      // Sletter displayet, flytter til 0,0.
lcd.noDisplay(); // Blanker displayet uden at glemme teksten
lcd.Display();    // Viser teksten igen
```

Udfordring 3

Skriv programmet om, så tiden tæller ned fra 100 til 0 og derefter starter forfra.

Du vil opdage, at gammel tekst ikke forsvinder af sig selv. Hvis der stadigt står uønskede (gamle) tegn *efter* noget, der udskrives, kan man blot tilføje en linje, som skriver nogle blanke tegn ud oveni: `lcd.print("    ");`

Alternativt kan man rense hele displayet med `lcd.clear();` Prøv det. Det afhænger lidt af sammenhængen, om man oplever, at displayet kommer til at flimre, når det hele slettes inden ny udskrivning. Ofte er den bedste løsning at skrive oveni.

Udfordring 4

Undersøg, hvad der sker, når du udskriver tekst, som går lidt ud over displayets højre kant.

Prøv også hvad der sker, hvis du skriver 40-50 tegn ud over kanten.

Der er flere, mere eksotiske muligheder med LiquidCrystal-biblioteket – som man kan finde beskrevet på nettet:

https://www.arduino.cc/en/Reference/LiquidCrystal

Der er f.eks. anvisning på, hvordan man kan definere egne specialtegn (såsom smiley eller danske tegn æ, ø og å).

Bemærk i øvrigt, at dette display er en ganske langsom komponent – det kan f.eks. ikke betale sig at forsøge at opdatere det konstant. Meget mere end 10 gange i sekundet vil kun føre til flimmer, 5 gange er passende til det meste.

Udfordring 4

Skriv et program, som tager tid på udskrivning til displayet. (Resultatet skal naturligvis udskrives i displayet).

Prøv både med udskrivning af 16 tegn ad gangen og med en opdelt udskrift, hvor cursoren flyttes manuelt flere gange.

Skriv hele programmet i `setup()` –funktionen, det behøver kun køre én gang.