

# Arduino en LCD

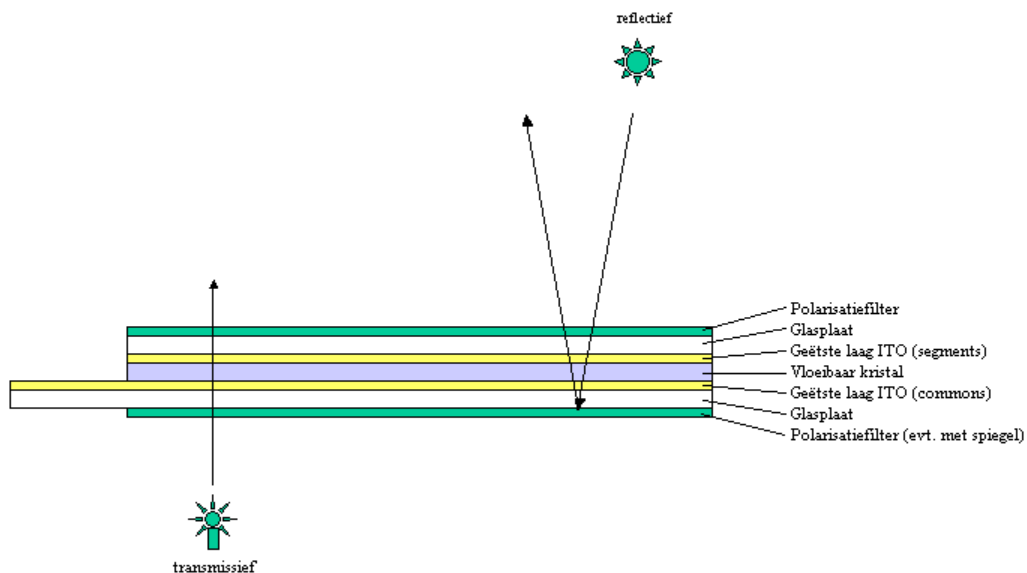
Door Johan ON5EX

Wat we ook meten of verwerken met de Arduino, in de meeste gevallen moeten we een 'ei' kwijt: temperatuurwaarde, luchtdruk, antennerichting (azimut en/of elevatie), tijd, frequentie, enz. Een 'liquid crystal display' (LCD) is een handig uitvoermiddel: klein, gemakkelijk in te bouwen, laag stroomverbruik, enz.

## Werkingsprincipe van een LCD

Een LCD bevat:

- een eerste lichtpolarisatiefilter
- een eerste elektrode
- een tussenlaag 'vloeibaar kristal'
- een tweede elektrode
- een tweede polarisatiefilter
- een spiegel (reflectief LCD) of een kunstmatige lichtbron (transmissief LCD)
- een LCD-driver

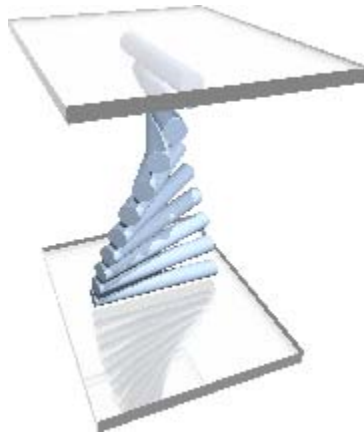


(Bron: wikipedia)

## ***Vloeibaar kristal***

Stoffen worden over het algemeen ingedeeld in vaste, vloeibare en gasvormige stoffen. Bij vaste stoffen liggen de moleculen star op een vaste plaats. Als we een vaste stof verhitten, zal zij op een gegeven moment beginnen te smelten en overgaan in een vloeistof: alle moleculen bewegen vrij rond, maar blijven wel bij elkaar. Verhogen we de temperatuur nog meer, dan verdampt de vloeistof en worden de moleculen gasvormig. Ze bewegen vrij door elkaar en verspreiden zich over de ruimte (ze blijven niet meer bij elkaar zoals in vloeistoffen).

Sommige stoffen – vloeibare kristallen (liquid crystal, LC) of mesogene stoffen geheten – bezitten ook fasen tussen vast en vloeibaar. Hun moleculen kunnen vlot bewegen, maar vertonen toch nog een bepaalde samenhang. Typisch voor LCD zijn 'getwiste nematische' vloeibare kristallen (TN), zie de schets hieronder.



(Bron: wikipedia)

Een van de belangrijke kenmerken – vanuit het standpunt van een LCD bekeken - van een getwist nematisch vloeibaar kristal is dat onder invloed van een aangelegd elektrisch veld de draaiing (twist) wordt opgeheven en de moleculen zich oplijnen.

### **Lichtpolarisatie**

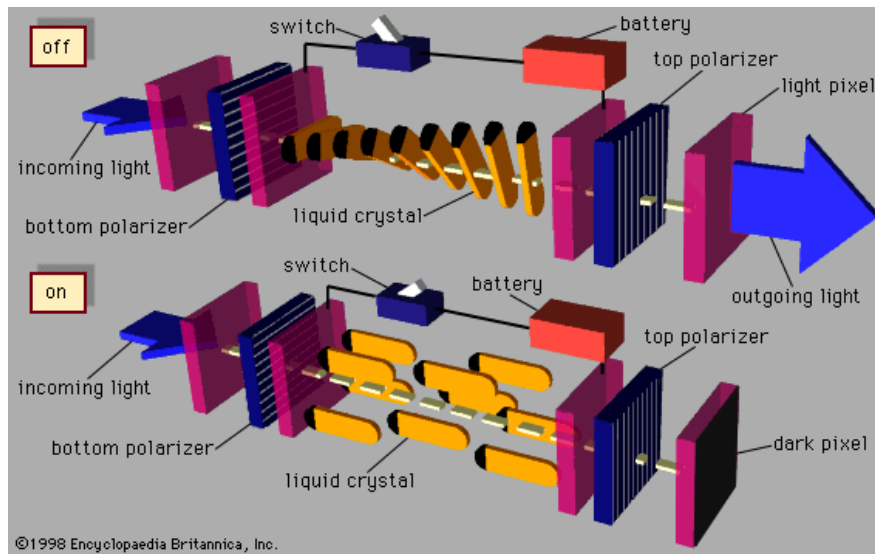
Licht is een elektromagnetische golf, gekenmerkt door amplitude, frequentie (kleur) en polarisatie. De polarisatie is de trillingsrichting van het elektrisch veld van de golf. Een lichtgolf plant zich meestal voort in vele trillingsvlakken (steeds haaks op de voortplantingsrichting). Dit is onpolariseerd licht. Het is echter mogelijk om via een polarisatiefilter enkel het licht dat trilt in een bepaald vlak door te laten. Een welbekend voorbeeld is de 'polaroid' zonnebril met een verticaal polarisatiefilter om het horizontaal gepolariseerd licht van lichtreflecties (op water, sneeuw, wegen) weg te filteren.

Belangrijk voor de werking van een LCD: als men 2 gelijke polarisatiefilters boven elkaar legt, maar 90° verschoven ten opzichte van elkaar, dan wordt er geen licht doorgelaten. Een interactief applet hierover kan je [hier](#) terugvinden.

### **Elektroden**

De elektroden van een LCD bestaan doorgaans uit op glas geëtste baantjes lichtdoorlatend indiumtinoxide (ITO). De ene elektrode bevat rijen, de andere kolommen. Op elk snijpunt van een rij en een kolom (de plaats van een pixel of beeldelement) kan een spanning worden aangelegd via de aansluitingen met een uitwendig stuur-IC.

## Werking



1. OFF: LCD 'in rust' = geen spanning op de elektroden  
Ongepolariseerd licht – bijvoorbeeld afkomstig van enkele LEDs aan de randen van de LCD-module ('backlight') dat vervolgens gelijkmatig wordt verstrooid over het LCD-oppervlak – komt terecht op het onderste polarisatiefilter, dat enkel horizontaal gepolariseerd licht doorlaat. De onderste elektrode laat het licht ongehinderd door. De moleculen van het vloeibaar kristal zijn van nature gedraaid (wenteltrapstructuur) en geleiden het licht tot bij de bovenste elektrode. Door de draaiing is het licht nu verticaal gepolariseerd, gaat door de bovenste elektrodel laag en bereikt het verticaal polarisatiefilter. Dit laat het verticaal gepolariseerd licht ongehinderd door: we zien helder licht.

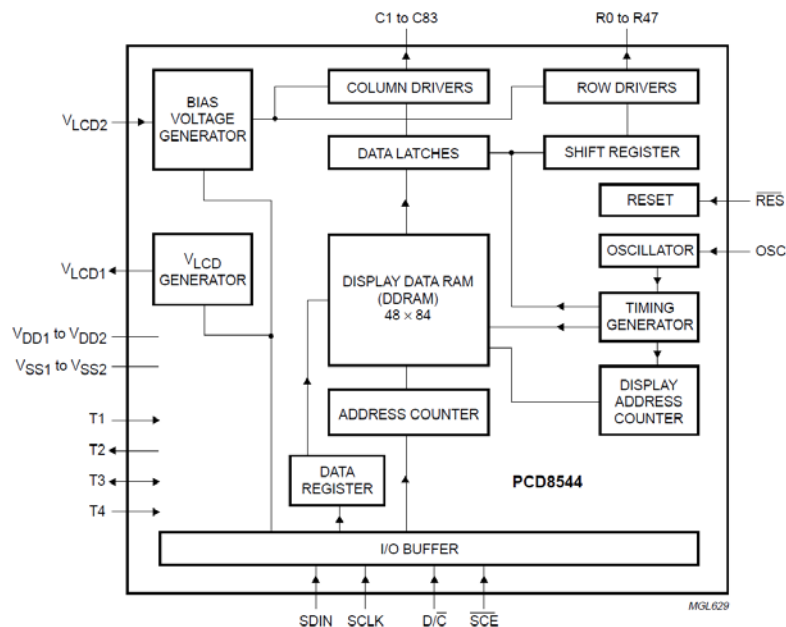


2. ON: een spanning over één van de kruispunten van de elektroden maakt dat de moleculen van het vloeibaar kristal zich daar richten naar het opgelegde elektrische veld. Het licht verandert nu niet van horizontale polarisatie-richting en wordt geblokkeerd door het bovenste, verticale polarisatiefilter. Resultaat: een zwart pixel.



## LCD-drivers

Om de pixels gemakkelijk aan te kunnen sturen vanuit een microcontroller, hebben fabrikanten specifieke LCD drivers of controllers ontworpen. Voorbeeld: de PCD8544 van Philips, een “48 x 84 pixels matrix LCD controller/driver”, specifiek bedoeld voor LCD in GSM-toestellen.



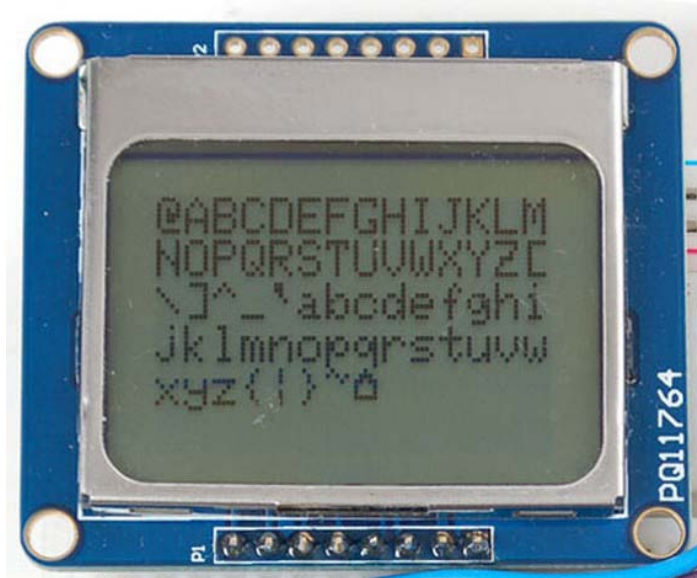
De chip is verbonden met de LCD-elektroden via de kolomdrivers C0-C83 en de rijdrivers R0 tot R48. De communicatie tussen een microcontroller en de LCD-driver verloopt volgens de Serial Peripheral Interface (SPI) standaard voor synchrone seriële datalinks. De volgende vijf ingangssignalen zijn essentieel voor de werking van de driverchip:

- Reset ( $\overline{RST}$ ): dit signaal (actief laag) “must be applied to properly initialize the chip”. “Attention should be paid to the possibility that the device may be damaged if not properly reset”.
- Serial Data Input ( $SDIN$ ): seriële data- of commandobytes (volgens Mode Select, zie hieronder)
- Serial Clock ( $SCLK$ ): het kloksignaal voor het inlezen van de seriële data
- Chip Enable ( $\overline{SCE}$ ): dit signaal (actief laag) bepaalt of er gegevens kunnen worden ingelezen. Zolang dit signaal hoog is, worden de klokpulsen genegeerd.
- Mode Select ( $D/\overline{C}$ ): dit signaal bepaalt of er commando's of data worden ingelezen

## Arduino en LCD

### **Hardware**

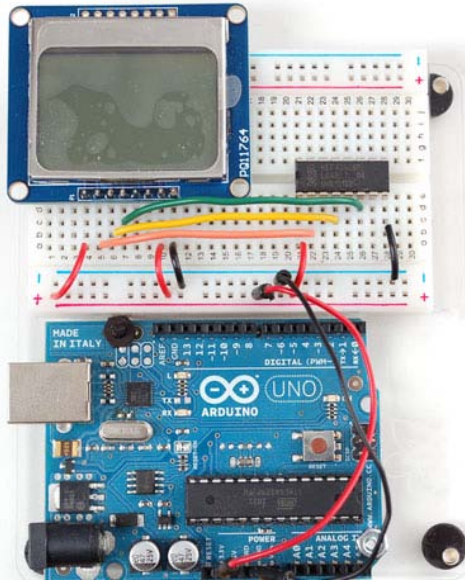
Een populaire LCD-module binnen de Arduino-gemeenschap is deze van de (intussen voorbijgestreefde) Nokia 5110 of 3110 GSM. Het is een monochroom 84x48 pixels LCD met de Philips PDC8544 als controller. De module (met breakout) is voor een prikke (5 tot 8 euro) te koop.



Op het breakout printje zitten 8 aansluitingen met de PDC8544 controller:

- RST: de resetingang
- CE: de chip enable ingang
- DC: de mode select ingang
- DIN: de data-ingang
- CLK: de klokpulsingang
- Vcc: de voedingsspanning
- GND: de massa
- BL: LED backlight

Alle spanningen van de LCD zijn typisch 3,3 V. Vcc en BL (backlight) verbinden we met de 3,3 V uitgang van de Arduino, GND met GND. De overige ingangen moeten elk met een (willekeurige) digitale uitgang van de Arduino worden verbonden. Deze schakelen echter tussen 0 en 5 V. Theoretisch moeten we dus de data-uitgangen via een spanningsconverter (bijv. een 4050 IC) naar 3,3 V omzetten.



(Bron: <http://ladyada.net/products/nokia5110/>)

In de praktijk blijken de 5 V uitgangsniveaus geen probleem te vormen en kan men de level-converter uitsparen. Er kan ook bespaard worden op het aantal digitale lijnen. Zo kan de resetlijn worden aangesloten op de resetpin van de Arduino, en de chip enable aan massa gelegd worden.

### Software

Moeten we de Arduino van naaldje tot draadje programmeren om een LCD, of beter uitgedrukt, een LCD-driver aan te sturen? Gelukkig kunnen we dankbaar gebruik maken van vrij beschikbare programmabibliotheken. De eerste regel van elke Arduinist luidt immers: “**Check the Libraries**” (<http://arduino.cc/en/Reference/Libraries> + googelen). Nemen we bijvoorbeeld de hiervoor vermelde PCD8544 LCD-driver als voorbeeld, dan vinden we hiervoor specifieke libraries, waarvan de meest gekende op naam van Limor Fried (alias [Ladyada](#)), een opmerkelijke dame, zaakvoerder van [Adafruit](#). Dank zij deze libraries wordt het sturen van een LCD een fluitje van een cent.

### Initialisatiestappen

1. Uiteraard moeten de nodige programmabibliotheken worden opgenomen in de sketch. In het geval van de Nokia 5110:

```
// Library Adafruit_GFX: https://github.com/adafruit/Adafruit-GFX-Library  
// Library Adafruit_PCD8544: https://github.com/adafruit/Adafruit-PCD8544-Nokia-5110-LCD-library  
#include <Adafruit_GFX.h>  
#include <Adafruit_PCD8544.h>
```

2. Vervolgens moet vermeld worden welke Arduino-pennen worden verbonden met de LCD-module, bijvoorbeeld:

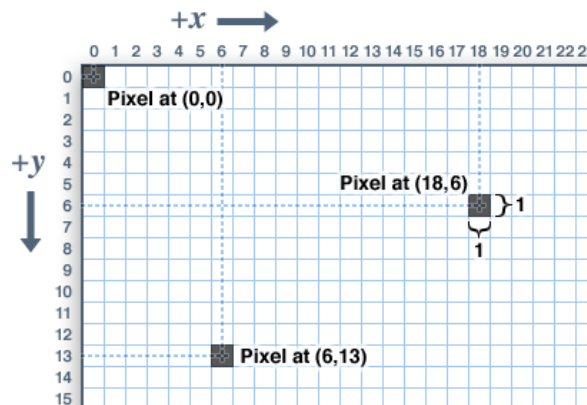
```
/* Toegepaste penconfiguratie:  
   pen 9 - Serial clock out (SCLK)  
   pen 10 - Serial data out (DIN)  
   pen 11 - Data/Command select (D/C)  
   pen 4 - LCD chip select (CS)  
   pen 3 - LCD reset (RST)  
*/  
Adafruit_PCD8544 display = Adafruit_PCD8544(9, 10, 11, 4, 3);
```

- 3. Tenslotte moet het LCD geïnitieerd worden:

```
display.begin();
```

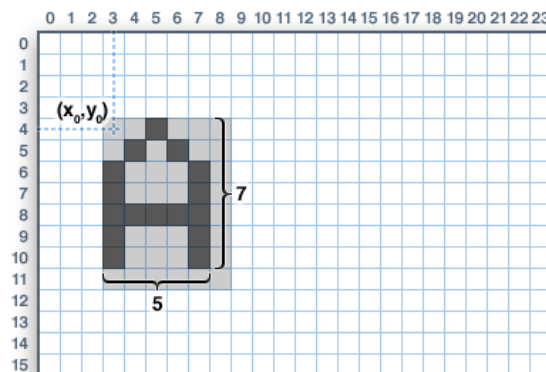
### Tekst afbeelden

- De Adafruit-library definieert één lettertype (font)
- Elk teken omvat omvat 6 x 8 pixels (breedte x hoogte). Dit betekent dat maximum 6 lijnen van 14 tekens op het LCD-scherm kunnen afgebeeld worden
- Coördinaten worden uitgedrukt in pixels. (x=0, y=0) is de bovente pixel in de linkerbovenhoek van het scherm; (x=83, y=47) is de onderste pixel in de rechterbenedenhoek.



(Bron: <http://learn.adafruit.com/adafruit-gfx-graphics-library/coordinate-system-and-units>)

- clearDisplay() maakt buffer en scherm leeg
- setCursor(x, y) bepaalt de beginpositie van een teken of tekenreeks



(Bron: <http://learn.adafruit.com/adafruit-gfx-graphics-library/coordinate-system-and-units>)

- Met print() en/of println() wordt de tekst in de buffer geplaatst, bijvoorbeeld:

```
display.println("Frequentie");
display.print(freq);
```

Voor meer info over println() en print(): raadpleeg de language reference op de Arduino website, onderdeel [Serial](#).

- Met display() wordt de inhoud van de buffer op het scherm geplaatst.

Hierna volgt een summier voorbeeld van tekstafhandeling:

```
/*
*****
* Nokia 5110 LCD-module (Philips PDC8544 controller) *
* 15/08/2012 Demo: tekst afbeelden *
*****
// Library Adafruit_GFX: https://github.com/adafruit/Adafruit-GFX-Library
// Library Adafruit_PCD8544: https://github.com/adafruit/Adafruit-PCD8544-Nokia-5110-LCD-library
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>

/* Toegepaste penconfiguratie:
   pen 9 - Serial clock out (SCLK)
   pen 10 - Serial data out (DIN)
   pen 11 - Data/Command select (D/C)
   pen 4 - LCD chip select (CS)
   pen 3 - LCD reset (RST)
*/
Adafruit_PCD8544 display = Adafruit_PCD8544(9, 10, 11, 4, 3);

// freq definiëren (in de praktijk bv. afkomstig van een software frequentieteller)
float freq = 3580.021;

void setup()
{
  // LCD initialiseren
  display.begin();
  // buffer en scherm leegmaken
  display.clearDisplay();
  // LCD-contrast instellen (waarde 32)
  display.setContrast(30);
  // Cursor op x=15, y=0 plaatsen
  display.setCursor(15,0);
  // Titel oplaan in buffer
  display.println("Frequentie");
  // Cursor op x=15, y=15 plaatsen
  display.setCursor(15,15);
  // freq opslaan in buffer, 3 decimalen
  display.print(freq, 3);
  // Buffer wegschrijven naar display
  display.display();
}
void loop() {}
```





Er zijn nog andere tekstafhandelingsroutines die interessant kunnen zijn, bijvoorbeeld een functie `setTextSize(1/2/3)` om de tekengrootte te verdubbelen of te verdriedubbelen. Aanbevolen lectuur: "[Adafruit GFX Graphics Library](#)".

### **De beste contrastinstelling zoeken**



`setContrast(contrastwaarde)` stelt het beeldcontrast in, dat voor elk LCD kan variëren. Een behulpzame routine om de beste contrastwaarde te bepalen ziet eruit als volgt:

```
/*
 * Nokia 5110 LCD-module (Philips PDC8544 controller
 *   Vind de beste contrastwaarde
 */

// Library Adafruit_GFX: https://github.com/adafruit/Adafruit-GFX-Library
// Library Adafruit_PCD8544: https://github.com/adafruit/Adafruit-PCD8544-Nokia-5110-LCD-library
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>
/* Toegepaste penconfiguratie:
   pen 9 - Serial clock out (SCLK)
   pen 10 - Serial data out (DIN)
   pen 11 - Data/Command select (D/C)
   pen 4 - LCD chip select (CS)
   pen 3 - LCD reset (RST)
*/
Adafruit_PCD8544 display = Adafruit_PCD8544(9, 10, 11, 4, 3);
void setup()
{
  // initialisatie van het LCD
  display.begin();
}
void loop()
{
  // findBestContrast(contrastMin, contrastMax)
  findBestContrast(25, 50);
}
void findBestContrast(int contrastMin, int contrastMax)
{ for (int contrast=contrastMin; contrast<=contrastMax; contrast++)
  {
    // veeg display schoon
    display.clearDisplay();
    // contrastwaarde = tellerwaarde
    display.setContrast(contrast);
    // normale tekengrootte
    display.setTextSize(1);
    // buffer "Contrast" + lijnsprong
    display.println(" Contrast");
    // dubbele tekengrootte
    display.setTextSize(2);
    // plaats de cursor op x=30, y=20
    display.setCursor(30, 20);
    // buffer contrastwaarde
```

```
    display.print(contrast);  
  // display buffer  
  display.display();  
  // 1 sec pauze  
  delay(1000);  
}  
}
```

### ***Grafische mogelijkheden***

De grafische mogelijkheden zijn legio:

- drawPixel(x, y, kleur): teken een pixel op (x, y)
- drawLine(x0, y0, x1, y1, kleur): teken een lijn van (x0, y0) tot (x1, y1)
- drawRect(x0, y0, b, h, kleur): teken een rechthoek vanaf (x0, y0), b pixels breed, h pixels hoog
- fillRect(x0, y0, b, h, kleur): vul een rechthoek vanaf (x0, y0), b pixels breed, h pixels hoog
- drawCircle(x0, y0, r, kleur): teken een cirkel met middelpunt op (x0, y0) en straal r
- fillCircle(x0, y0, r, kleur): vul een cirkel met middelpunt op (x0, y0) en straal r
- drawRoundRect(x0, y0, b, h, r, kleur): teken een rechthoek vanaf (x0, y0), b pixels breed, h pixels hoog, straal r voor de afgeronde hoeken
- fillRoundRect(x0, y0, b, h, r, kleur): idem, maar gevuld
- drawTriangle(x0, y0, x1, y1, x2, y2, kleur) : teken een driehoek met de hoekpunten op (x0, y0), (x1, y1) en (x2, y2)
- fillTriangle(x0, y0, x1, y1, x2, y2, kleur) : idem, maar gevuld
- drawBitmap(x, y, \*bitmap, b, h, kleur) : teken een in het programmeergeheugen opgeslagen bitmap vanaf (x, y) met breedte b en hoogte h

Nog meer functies worden beschreven in "[Adafruit GFX Graphics Library](#)".