

# Maar eerst..... wat moet je weten als je met een arduino gaat werken

## Inhoudsopgave

Introductie .....	2
Het Breadboard.....	4
LED's .....	5
Schakelaars: .....	6
Het openen van voorbeelden .....	9
Opslaan van Arduino bestanden.....	11
Openen van Arduino bestanden .....	12
Libraries.....	13

## Introductie

### ▪ Wat is een Arduino?

Arduino is de naam van een microprocessorbord. Een microprocessor is een kleine computerchip die geprogrammeerd kan worden. Je kunt met een Arduino allerlei apparaten maken. Een robotje, een elektronische dobbelsteen, een looplicht, een alarmsysteem, enz.

Er zijn veel verschillende Arduino's. Sommige groot, andere veel kleiner. De eigenlijke chip is piepklein, dingen er om heen als een USB converter chip of een spanningsregelaar maken het geheel wat groter.



Een Arduino programmeer je met een computer, meestal met een laptop.

Eerst schrijf je een programma, daarna stuur je het naar de Arduino via een USB kabel.

Als het allemaal goed werkt kun je daarna de kabel weghalen en werkt jouw Arduino project op zichzelf (zoals de bedoeling is van een microprocessor) (Na het aansluiten van externe voeding!)

Er bestaat ook een Arduino Webeditor. Zie <https://create.arduino.cc/>

Maar deze cursus gaat uit van een vast geïnstalleerde Arduino programmeeromgeving.

Download de installer van <https://www.arduino.cc/en/Main/Software> (versie 1.8.5)

Heel vaak kun je de Arduino programma code (van iemand anders, van internet, van voorbeelden) gewoon knippen en in je eigen Arduino programma plakken.

Je hoeft het nooit allemaal over te typen! (Voor 'Knippen' en voor 'Plakken' zijn handige sneltoetsen namelijk Control-C en Control-V)

### ▪ Arduino en externe voeding

Na het uploaden is het programma in de Arduino geladen. Er is geen enkele manier om dat programma er weer "uit te krijgen". Het staat in de Arduino in 'gecompileerde' vorm opgeslagen.

De enige plek waar de oorspronkelijke bestanden staan is op de laptop (of "in the cloud" bij gebruik van de Arduino Webeditor).

Maar de USB verbinding met de laptop is niet nodig om een programma te 'draaien'.

Na het aansluiten van een externe voedingsspanning zal het laatst geladen programma automatisch starten.

Met de reset-button op de Arduino kan het desnoods opnieuw gestart worden.

Een externe 9V batterij is een prima externe voeding, zolang er niet al te veel stroom wordt getrokken. Bij gebruik van motoren of Neopixel led's zal al snel een netadaptor nodig zijn.

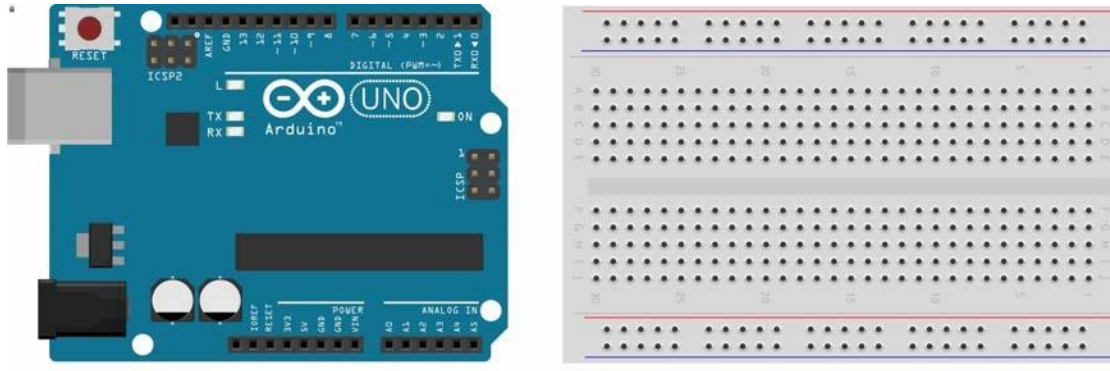
Deingangsspanning moet tussen de 6 en 20 Volt liggen. Batterijvoeding vanuit  $4 \times 1.5V = 6$  Volt is erg krap voor een Arduino Uno. Er is dan grote kans dat -zodra er een beetje stroom wordt gevraagd- de voedingsspanning op de Arduino onder de 5 Volt daalt en dan is het onzeker wat er gaat gebeuren. De Arduino kan ineens in reset gaan. Gebruik dus bij voorkeur tenminste 7V als voedingsspanning.

Arduino heeft een ingebouwd voedingscircuit. Ongeacht de ingangsspanning (mits boven de 6 Volt) is er altijd een stabiele 5 Volt beschikbaar als uitgang. De meeste shield werken op 5 Volt en niet op hogere spanningen! Gebruik daarom steeds de 5Volt van de Arduino op het breadbord.



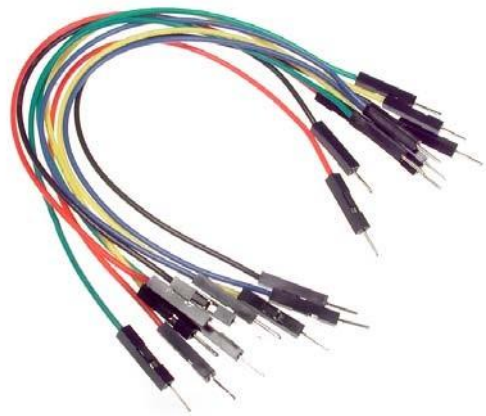
## Het Breadboard

Om onderdelen met een Arduino te verbinden gebruiken we een *breadboard*.



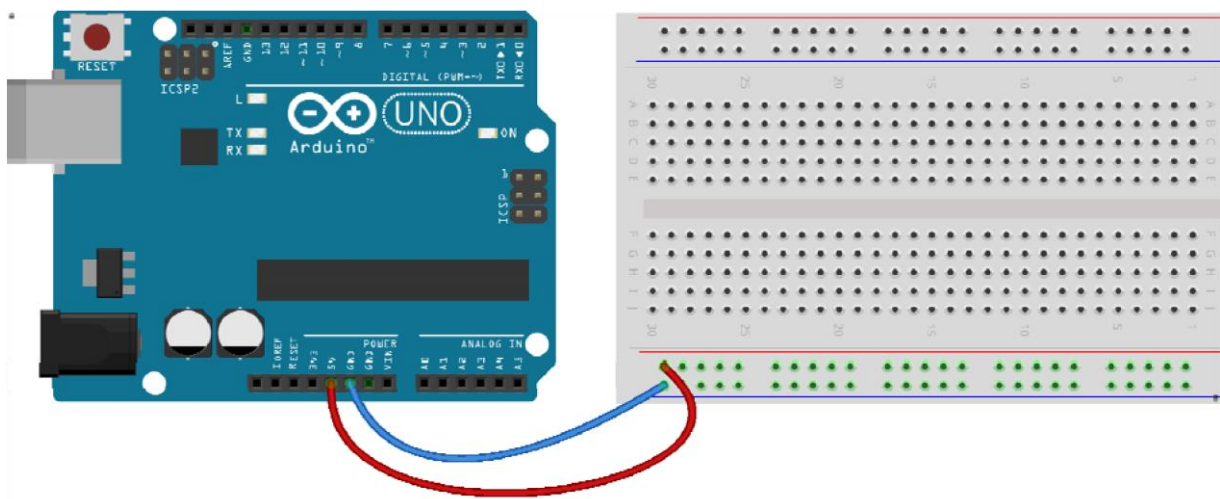
Een breadboard is een soort gaatjesbordje waar je onderdelen kunt insteken. De bovenste twee *Rijen* en de onderste twee *rijen* zijn in het bordje vormen een horizontale verbinding. De verticale rijen daar tussen zijn met elkaar doorverbonden. In het midden is een scheiding. Dus de bovenste helft en de onderste helft zijn niet met elkaar verbonden.

Verbindingen (tussen de Arduino en het breadbord) maken we met 'jumpers'. Dat zijn gekleurde kabeltjes met verschillende kleuren. Er zijn "Male-Male" kabeltjes, "Male-Female" en "Female-Female" Zulke verbindingen zijn vaak een bron van problemen.



Het Breadboard krijgt spanning via de Arduino. (En die krijgt weer zijn voedingsspanning via de computer of via de voedingsaansluiting (7V minimaal, 12V maximaal)

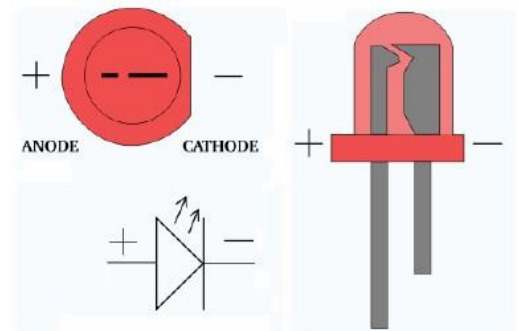
We verbinden de Gnd (*Ground*) en de 5V aansluiting van de Arduino met de rode en blauwe 'lijnen' op het breadboard. Gebruik **ALTIJD** rood voor de + draad en blauw (of zwart) voor de - draad.



## LED's.

Een LED is een Light Emitting Diode, een licht uitstralende diode. Dat werkt heel anders dan een gloeilamp. Een LED heeft een + kant en een - kant. Het lange pootje is de +, het korte pootje de -

De min is ook de platte kant van de Led.



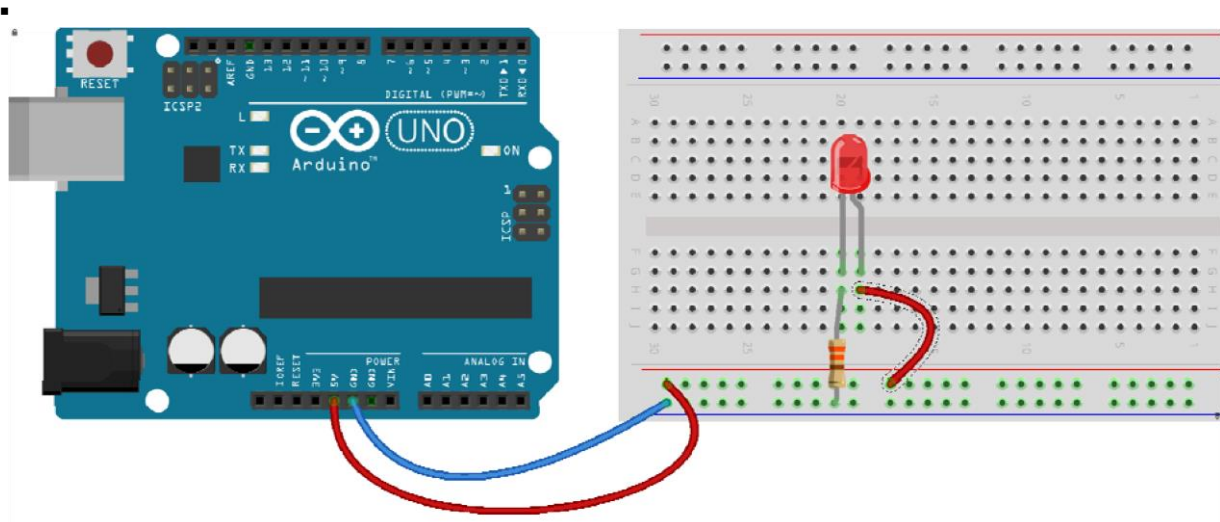
Led's kunnen doorbranden. Daarom zetten we altijd *in serie* met de Led een weerstand. Dat beperkt de stroom. Het maakt niet uit of die weerstand aan de + of aan de - kant zit.

We gebruiken bij Led's altijd een weerstand van 330 Ohm.

De waarde van een weerstand wordt met kleuren aangegeven.

Als je een weerstand zó houdt dat de goud of zilverkleurige band rechts zit dan zie je op een weerstand van 330 Ohm de kleuren Oranje-Oranje-Bruin.

Probeer eens de volgende schakeling met een rode Led en een 330 Ohm weerstand.



Als je nu de Arduino met de laptop verbindt via de USB kabel zal de Led gaan branden.  
 (De Arduino doet hier nog helemaal niets behalve dat hij de spanning van de computer doorgeeft aan het breadboard.)

### ■ Weerstanden:

De grootte van een weerstand wordt uitgedrukt in Ohm. Met de gekleurde ringen wordt de grootte van de weerstand aangegeven. Daarvoor geldt een kleurentabel.

De weerstand in het plaatje hiernaast is 1000 Ohm (1kΩ) Want: **1** omdat de eerste ring bruin is, **0** omdat de 2<sup>e</sup> ring zwart is gevolgd door **00** omdat de derde ring rood is. De vierde ring geeft de tolerantie aan. Goudkleurig staat voor 5% tolerantie.

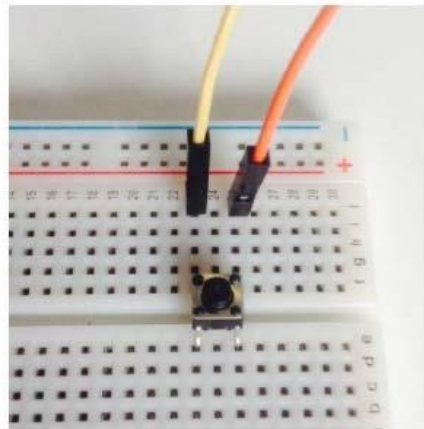
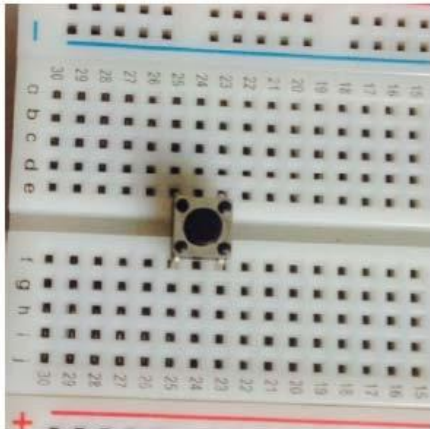


■ zwart	0	■ groen	5
■ bruin	1	■ blauw	6
■ rood	2	■ paars	7
■ oranje	3	■ grijs	8
■ geel	4	□ wit	9

ring 1 = cijfer  
 ring 2 = cijfer  
 ring 3 = aantal nullen

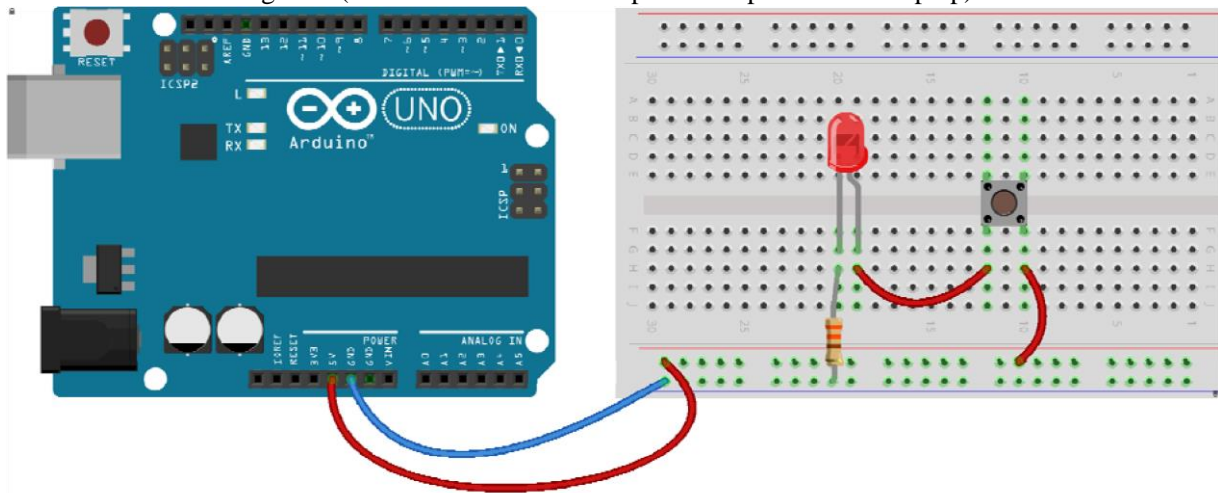
### Schakelaars:

Bij de kleine en de grote drukschakelaars zijn de aansluitingen aan de linkerkant met elkaar doorverbonden. Ook de twee aansluitingen aan de rechterkant vormen een verbinding.



Tussen de linker en de rechterkant wordt een verbinding gemaakt als je op de schakelaar drukt. Het is een *Push-button*.

Probeer deze schakeling eens (en sluit de Arduino aan op de USB poort van de laptop)



Variaties

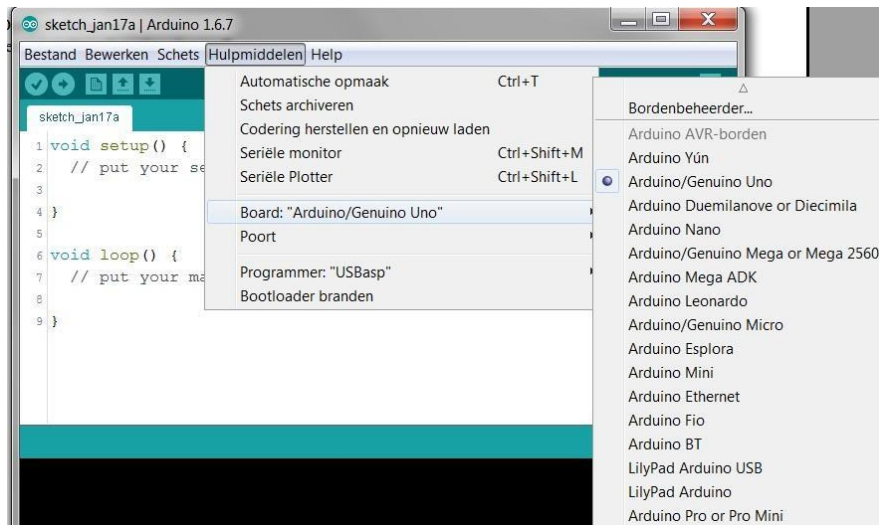
Voeg nog een groene of blauwe Led, weerstanden en extra schakelaars toe.

### Arduino in de praktijk

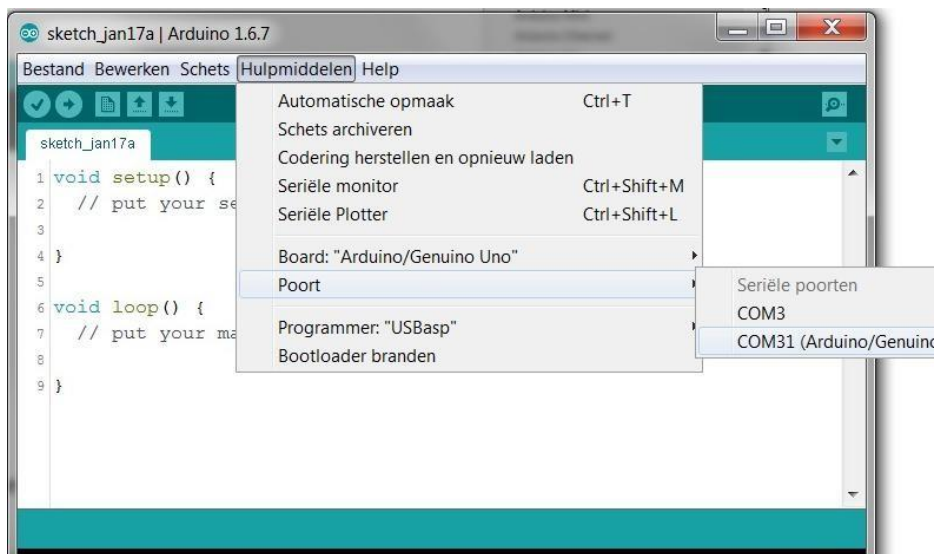
Sluit je Arduino aan op een USB poort. Arduino en jouw PC gaan met elkaar communiceren via een USB kabel. Open Arduino.

Kies voor *Hulpmiddelen* en kies als *Board* voor een Arduino/Genuino Uno





Kies voor *Hulpmiddelen* en controleer bij *Poort* of de laptop de Arduino 'ziet'. Als het goed is herkent de laptop vanzelf dat er een bepaalde Arduino is aangesloten.



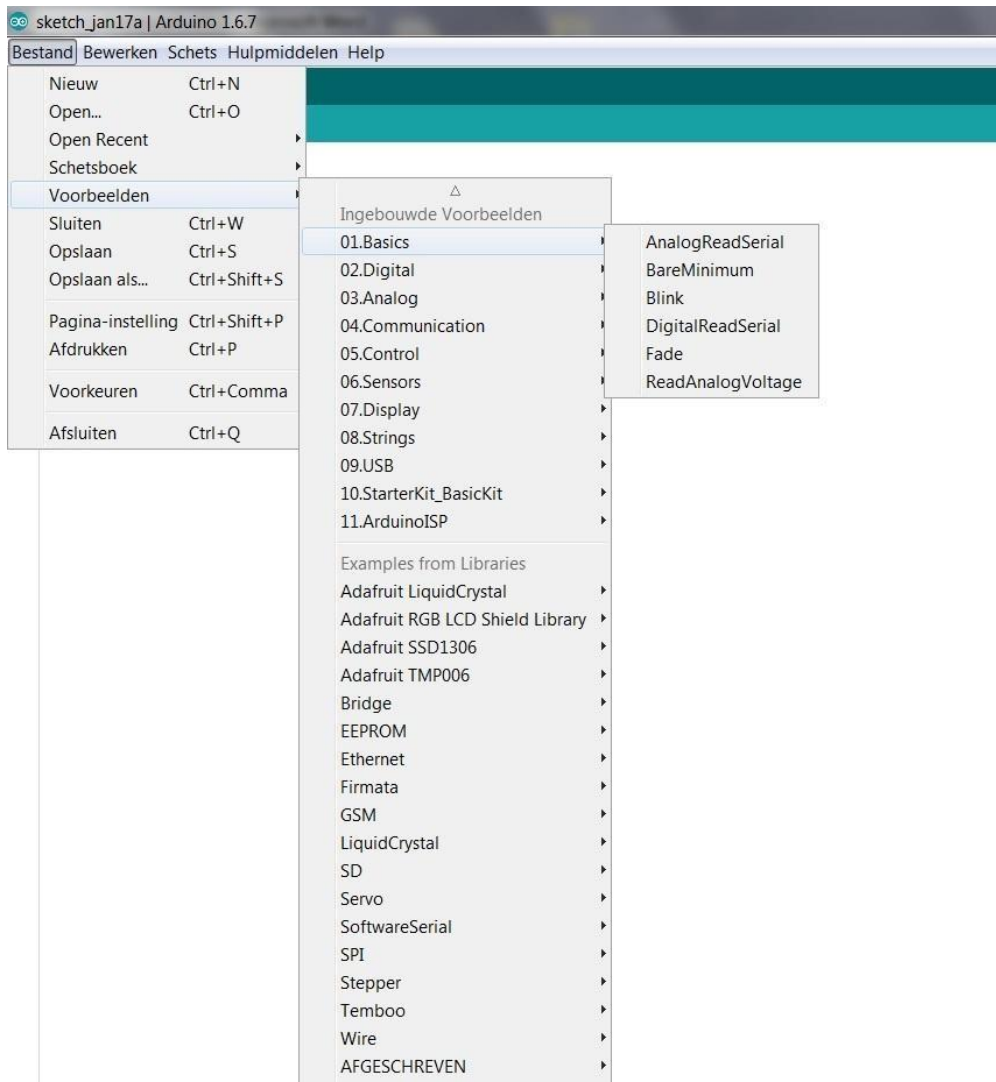


## Het openen van voorbeelden:

In de Arduino programmeeromgeving zijn veel voorbeelden te vinden van basis manieren om iets te doen. Uit zulke voorbeelden zullen we stukjes naar ons eigen programma's brengen door te 'knippen' en te 'plakken'. Kies voor *Bestand* en *Voorbeelden* en je krijgt de keuze uit heel veel voorbeelden.

Een handige groep programma's is *01. Basic*.

Kies (en open) het programma *Blink*.



Op je scherm zie je nu het voorbeeldprogramma *Blink*  
Een paar opmerkingen:

- Alles tussen */\** en *\*/* is commentaar en dat wordt door Arduino verder helemaal overgeslagen! Het is heel erg handig om in programma's veel commentaar te schrijven. Want al snel weet je anders niet meer wat er gebeurt.
- Elke *regel* die met *//* begint is voor Arduino ook een commentaar regel.

(Bijna) elke *programmaregel* wordt afgesloten met een *;*  
Een snel gemaakte fout is dat ergens een *;* is vergeten. Dat levert vreemde foutmeldingen op!

- Elk Arduino programma heeft twee belangrijke blokken namelijk het blok *setup* en het blok *loop*. Zo'n blok begint met een *{* en eindigt ook weer met een *}*.  
(Als je die accolades niet goed hebt staan krijg je hele vreemde foutmeldingen!)  
In het blok *setup* staan dingen die je éénmaal wilt vastleggen.  
Bijvoorbeeld dat je een Led hebt vastgemaakt aan pin 13.  
En b.v. dat pin 13 gebruikt gaat worden om iets naar buiten te sturen.  
De opdracht daarvoor is *pinMode(13, OUTPUT);*

Het blok *loop* wordt door de Arduino voortdurend 'doorlopen'. Alles wat daar staat wordt voortdurend, zonder ooit te stoppen herhaald en uitgevoerd door de Arduino. Er staan een paar regels die veel op elkaar lijken.

<i>digitalWrite(13, HIGH);</i>	Hiermee wordt pin 13 'Hoog' gemaakt. Dat is hetzelfde als 'met plus verbonden'. De Led gaat aan. Let op! ELKE regel wordt in Arduino met <i>;</i> afgesloten!
<i>delay(1000);</i>	De opdracht om 1000 milliseconden helemaal niets te doen
<i>digitalWrite(13, LOW);</i>	Hiermee wordt pin 13 'Laag' gemaakt. Dat is hetzelfde als 'met min verbonden'. De Led gaat uit.
<i>delay(1000);</i>	De opdracht om 1000 milliseconden helemaal niets te doen

We hebben dus een programma waarmee een Led die verbonden is met pin 13 gedurende 1 sec. aan gaat en 1 sec. uit! En bij een Arduino is er altijd een Led verbonden met pin 13. Die zit óp de Arduino.

Een *Knipperled* met het voorbeeld programma *Blink*

```
/*  
  Zet een LED aan voor 1 seconde en daarna uit voor 1 seconde. En herhaal dit. */  
  
// De Setup functie wordt één keer uitgevoerd als je het board aanzet.  
  
void setup() {  
  // Maak van pin 13 een Output pin.  
  pinMode(13, OUTPUT);  
}  
  
// De Loop functie wordt eindeloos lang doorlopen.  
void loop() { digitalWrite(13, HIGH); // Zet de Led aan. (HIGH is hetzelfde als  
verbinden met +) delay(1000);           // Wacht 1 seconde digitalWrite(13,  
LOW); // Zet de Led uit. (LOW is hetzelfde als verbinden met -) delay(1000);  
// Wacht 1 seconde }
```

We gaan programma Blink “*Uploaden*” naar de Arduino.

Maar eerst moet Arduino controleren of het programma in correcte programmeertaal is geschreven. Die controle heet *Compileren*.

Onder de menubalk (Bestand Bewerken Schets Hulpmiddelen enz. staan twee ‘knopjes’.

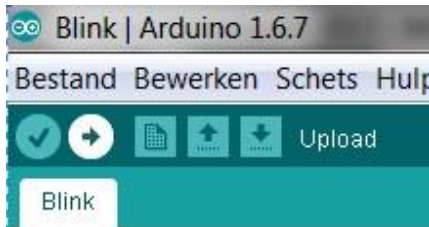
De meest linker is voor “Compileren”.



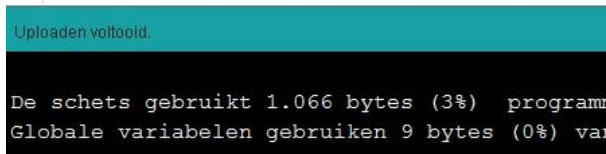
Na het compileren zie je onderin het scherm de melding Compileren voltooid.



Als er geen foutmeldingen worden gegeven kunnen we nu gaan *Uploaden* (het tweede knopje)



Tijdens het *Uploaden* wordt het programma nogmaals gecompileerd en daarna naar de Arduino gestuurd. Als alles goed gaat verschijnt onderin je scherm de melding Uploaden voltooid.



Het Ledje op de Arduino knippert 1x per seconde.

### Variaties:

Verander de delay waarden in de loop. Maak er bijvoorbeeld een flitslicht van.

```
void loop() { digitalWrite(13, HIGH); // Zet de Led aan. (HIGH is hetzelfde als  
verbinden met +) delay(50); // Wacht 50 mseconde  
digitalWrite(13, LOW); // Zet de Led uit. (LOW is hetzelfde als verbinden met -)  
delay(1000); // Wacht 1 seconde }
```

## Opslaan van Arduino bestanden

Als we het flitslicht, (de aangepaste ‘*Blink*’) willen opslaan dan lukt dat niet direct.

Want *Blink* was een voorbeeld, dat kunnen we niet veranderen.

Kies voor:

Bestand, Opslaan als, blader naar je USB stick en type als naam in: *Mijn Blink*

En nu gebeuren er verschillende dingen.

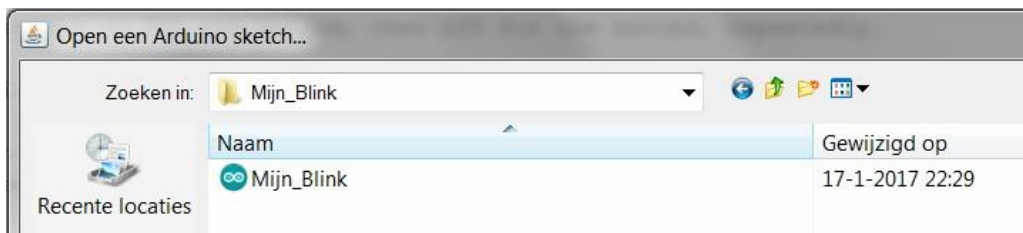
- Arduino accepteert geen naam met een spatie en maakt er *Mijn\_Blink* van.
- Arduino maakt een *map* aan met de naam *Mijn\_Blink*.
- Arduino slaat in die map het bestand op als *Mijn\_Blink.ino* Een programma in Arduino heet een sketch.

## Openen van Arduino bestanden

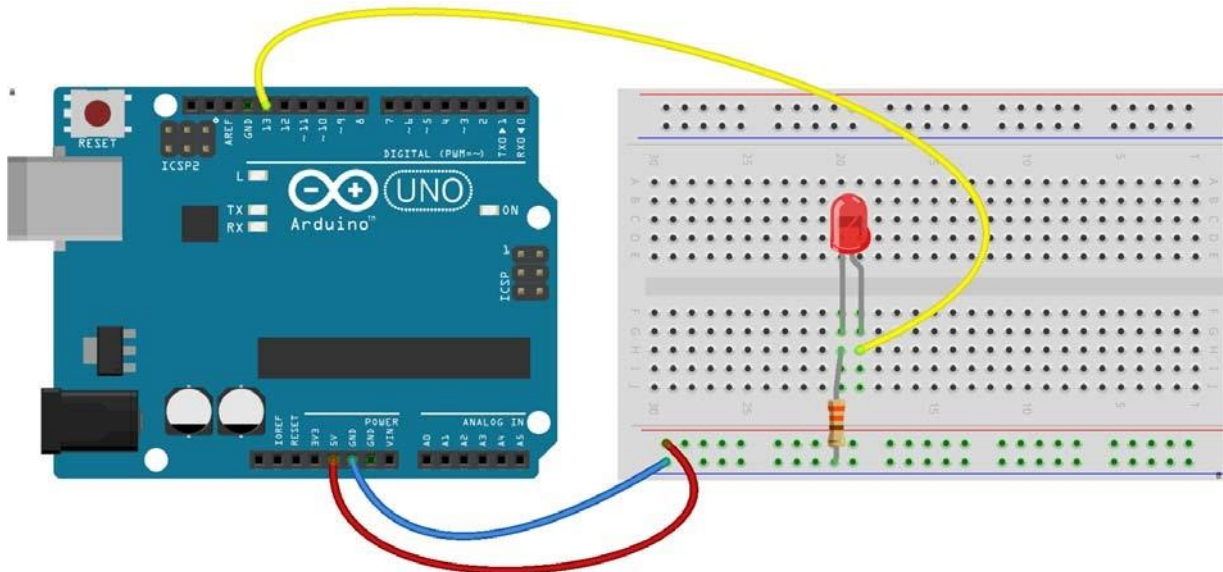
Kies voor:

Bestand, Openen, blader naar je USB stick en daar zie je een map *Mijn\_Blink*. Klik er op en je ziet in die map het programma *Mijn\_Blink* staan. Door er op te klikken open je het bestand.

(Een handige manier om één van je laatste programma's te openen is Open Recent)



Nu gaan we het programma zo aanpassen dat de Led op het breadboard gaat knipperen. Verbind de + kant van de Led met pin 13.



Er is niets veranderd aan het programma. Pin 13 wordt “*even hoog*”, wordt daarna weer “*even laag*”. De Led knippert mee met de Led op de Arduino.

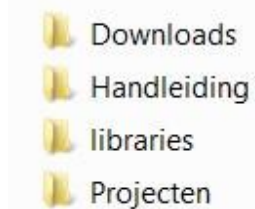
## Libraries

In veel Arduino programma's wordt gebruik gemaakt van een bibliotheek. Een *library*.

Bij deze cursus hoort een USB met bestanden en alle benodigde libraries.

Alle bestanden kunnen ook online worden gevonden op <https://tinyurl.com/y88ugf2o>

De inhoud van die USB stick en de map op internet is:



Libraries moeten geplaatst worden in C:\Program Files\Arduino\libraries óf, -beter eigenlijk- in C:\users\usernaam\Documents\Arduino

Na het wijzigen of plaatsen van een library moet Arduino opnieuw gestart worden!

Zie ook <https://www.arduino.cc/en/Guide/Libraries>