

### Benodigdheden voor deze module

Arduino  
 Breadboard  
 Jumperkabels  
 Weerstanden (220  $\Omega$ ; 330  $\Omega$ ; 1 k $\Omega$ ; 10 k $\Omega$ )  
 Potmeter (1 k $\Omega$ ; 10 k $\Omega$ )  
 Diode  
 2x LDR  
 LED (rood, geel, groen)  
 RGB LED  
 Servo motor  
 DC Motor  
 5x Drukknop  
 2N2222 transistor  
 LCD  
 Actieve piëzo element

### Benodigde voorkennis

Wet van Ohm	$R = \frac{U}{I}$
Serieschakeling	stroomsterkte gelijk
Parallelschakeling	spanning gelijk
Vermogen	$P = U \cdot I = I^2 \cdot R$
Werking dynamo/elektromotor	

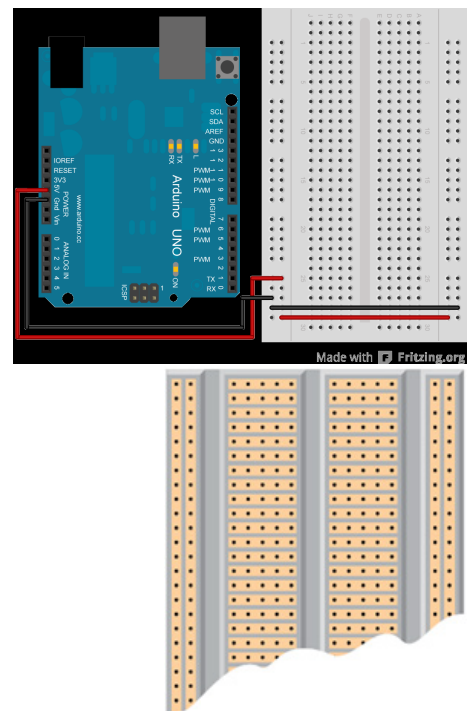
## Introductie

Leuk dat je aan de slag gaat met Arduino! Wellicht heb je nog geen idee wat een Arduino is en wat je er mee kunt. Een Arduino is een soort microcomputer. De Arduino sluit je aan op een computer waarbij je een programma (in Arduino noemen ze dat een sketch) stuurt naar de Arduino. De Arduino voert vervolgens je geschreven script uit en zorgt voor de uitvoer. Zo kun je een koelkast op een bepaalde temperatuur houden, een zelfrijdende robot aansturen, lichtsensoren maken, een lcd scherm op je shirt aansturen en ga zo maar door. Ook is het mogelijk om geen software te gebruiken en alleen met de elektronica te spelen. De Arduino is dan de spanningsbron.

In deze module gaan we aan het werk met de Arduino en leren we de basismogelijkheden van een Arduino. Bij het werken met een Arduino heb je twee belangrijke onderdelen: De Arduino en een breadboard. De Arduino is de computer, met invoer en uitvoer mogelijkheden. Op het breadboard sluit je de elektronica aan die aangestuurd wordt door de Arduino.



## Breadboard



Het breadboard heeft aan beide zijdes twee kolommen die verbonden worden met de voeding (+ en -). De + kant sluit je aan op de 5 V uitgang of op een uitvoerpoort van de Arduino. De – kant sluit je aan op de GND (ground) van de Arduino. Alhoewel je de constante output niet altijd gebruikt, is het wel verstandig om deze altijd aan te sluiten.

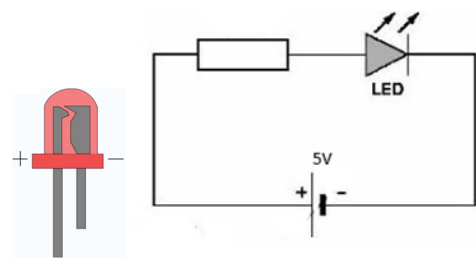
Het breadboard heeft rijen en kolommen, zie hiernaast en hieronder voor een opgewerkte breadboard. De punten in een rij zijn met elkaar verbonden. Maar laten we hier niet te lang bij stil staan...we gaan aan de slag!

**NB:** Vaak moet je even aangeven in welke USB-Poort je Arduino zit. Dit doe je door in het programma te gaan naar: Hulpmiddelen/Poort. Daar kun je de juiste USB poort aanklikken.

## Opdracht 1 Het aansluiten van een LED

Een LED is een diode die licht uitzendt als er stroom door de LED gaat. De LED laat stroom maar door in één richting. De lange poot van de LED moet altijd op de + kant aangesloten worden, de korte poot op de – kant, zie onderstaand figuur.

De stroom door een LED mag meestal maar 20 mA zijn. De spanning over de LED is dan ongeveer 2.0 V, deze waarden verschillen een beetje per soort LED, zie onderstaande tabel. De Arduino levert een 5.0 V spanning. De LED moet dus in serie geschakeld worden met een weerstand. De weerstand moet dus minimaal  $150\ \Omega$  zijn ( $R = \frac{U}{I} = \frac{3.0\text{ V}}{0.020\text{ A}} = 150\ \Omega$ ).



KLEURCODE VAN WEERSTANDEN

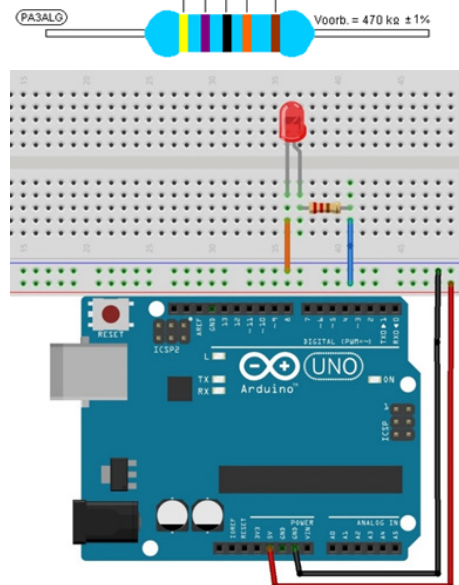
KLEUR	1e RING	2e RING	3e RING	MULTIPL.	TOL.
ZWART	0	0	0	1	
BRUIN	1	1	1	10	± 1%
ROOD	2	2	2	100	± 2%
ORANJE	3	3	3	1k	
GEEL	4	4	4	10k	
GROEN	5	5	5	100k	± 0,5%
BLAUW	6	6	6	1M	± 0,25%
VIOLET	7	7	7	10M	± 0,10%
GRIS	8	8	8		± 0,05%
WIT	9	9	9		
GOUD				0,1	± 5%
ZILVER				0,01	± 10%
BLANK					± 20%

- Er is geen weerstand aanwezig van  $150\ \Omega$  maar wel een van  $220\ \Omega$ . Zoek de weerstand op met behulp van de kleurcode: de eerste ring moet rood zijn, de tweede ring ook en de derde ring bruin ( $22 \cdot 10$ ). Een andere mogelijkheid is rood, rood, zwart, zwart ( $220 \cdot 1$ ).
- Sluit nu de 5 V uitgang van de Arduino aan op de + kolom en de GND (ground) uitgang van de Arduino aan op de – kolom.
- Sluit de LED en de weerstand in serie aan, zie de tekening.
- Verbind de Arduino via de usb met de computer. Als je het goed hebt gedaan brandt de LED!

Doordat de rode led bij lagere spanning licht uit zendt dan bijvoorbeeld een groene led, kun je een kleinere weerstand bij groene en blauwe led's gebruiken. Zo branden ze uiteindelijk toch nog even fel!

### Kleur Drempelspanning

Blauw	2.3 V
Groen	2.0 V
Rood	2.0 V




## Opdracht 2 Een knipperende LED

Nu kun je je led wel laten branden, maar daar heb je nog niet veel aan...Een volgende stap is de LED aansturen met behulp van een stukje code. Om de LED aan te sturen met behulp van code moet je een uitvoerpoort (bijvoorbeeld pin 13) van de Arduino verbinden met je LED. De 5 V output heb je nu niet nodig, pin 13 levert nu de spanning. In de tekening zie je dat deze wel verbonden is, bij grotere projecten vergeet je snel de 5 V, vandaar!

- Bouw de opstelling die je ziet in de tekening en sluit de Arduino aan op de computer.

We willen de Arduino aansturen, daarvoor gebruiken we het Arduino programma.

- Open het programma en open het script blink via: bestand/voorbeelden/basis/blink.
- Controleer het script met het vinkje. Mocht het programma niet werken, dan geeft het onderaan een foutmelding weer (het kan zijn dat je de COM-poort moet toewijzen, dit doe je via hulpmiddelen/poort).
- Upload het script naar je Arduino met het teken  (snelcode: ctrl + u)
- Beschrijf kort wat je ziet. Probeer wat je ziet te verklaren met behulp van de code.
- Verander het script zodat de LED sneller knippert.
- Sluit drie verschillende LED lampjes aan, verander het script zodat ze om de beurt aan staan.
- Verander het script en maak een verkeerslicht waarbij oranje maar even aan staat.

## Programmeren deel 1

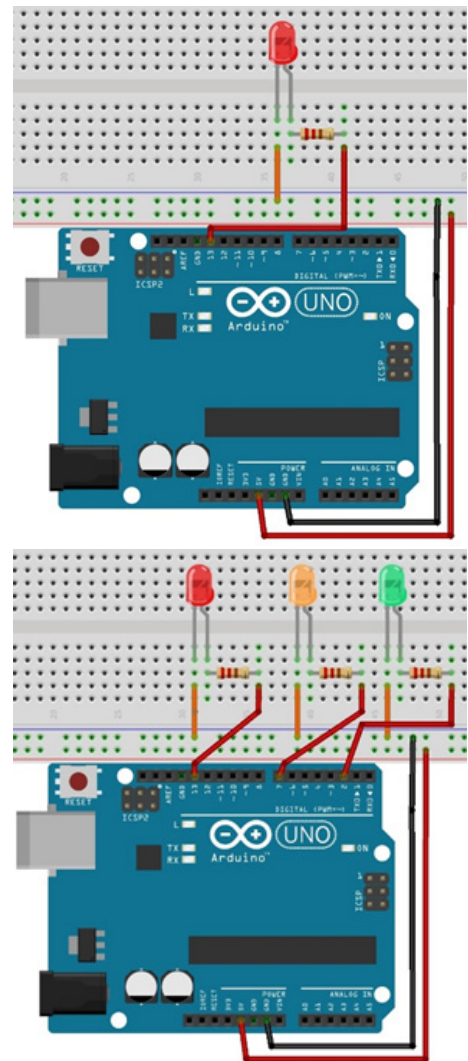
Arduino heeft een eigen programma dat gebaseerd is op C++. Heb je al ervaring met programmeren en/ of met C++ (of java), dan is het programmeren heel makkelijk. Heb je geen ervaring met programmeren? Het is veel minder moeilijk dan je misschien wel denkt!

Bovenaan het script Blink staat `/*` dit betekent dat alles na dit beginteken en voor het afsluitteken `*/` commentaar is. Hier zet je neer hoe het programma heet, wat het doet, wie het gemaakt heeft en wanneer je het voor het laatst hebt veranderd.

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

Een tweede mogelijkheid om commentaar toe te voegen is met behulp van `//`. Alles op dezelfde regel na `//` is commentaar. Zo kun je bijhouden wat de code op die regel doet. Met behulp van de snel code ctrl / kun je code snel omzetten naar commentaar.

Voordat de belangrijkste code wordt uitgevoerd moet je zeggen wat er precies aangestuurd wordt. We geven aan dat in pin 13 iets zit en dat de Arduino daar een output (spanning) geeft als jij dat wilt. Iets netter zou zijn om nog voor de setup aan te geven hoe pin 13 heet: je geeft de pin een naam (`int LEDrood = 13;`). Wil je pin 13 vervolgens aansturen, dan kan je dat doen met de naam LEDrood.



Alles wat tussen de accolades van de loop staat wordt continu herhaald. Eerst wordt pin 13 hoog (5.0 V) gemaakt met behulp van de code `digitalWrite` (`digitalWrite` kan alleen aan of uit). Daarna moet het programma 1000 ms wachten (delay) voordat de volgende regel code wordt uitgevoerd. De volgende regel code maakt pin 13 weer laag (0.0 V).

De pin wordt aangestuurd met een hoog of met een laag signaal. Zit daar nog iets tussen? Ja en nee...De output is altijd 0.0 V of 5.0 V. Maar je kunt de LED wel dimmen door maar een bepaalde tijd de LED aan te zetten. Als de LED snel genoeg knippert zie je niet dat de LED knippert, het lijkt er alleen op dat de LED minder fel brandt. Wanneer je een LED wilt dimmen gebruik je een output met het symbool  $\sim$ . Dit is een Puls Width Modulation (PWM). De waarde van de PWM zit tussen de 0 (geheel uit) en 255 (geheel aan).

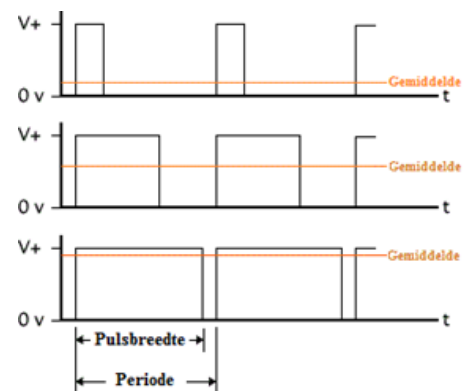
```

Blink$
1  /*
2   Blink
3   Turns on an LED on for one second, then off for one second, repeatedly.
4   Most Arduinos have an on-board LED you can control. On the Uno and
5   Leonardo, it is attached to digital pin 13. If you're unsure what
6   pin the on-board LED is connected to on your Arduino model, check
7   the documentation at http://www.arduino.cc
8
9   This example code is in the public domain.
10  */
11
12
13  // the setup function runs once when you press reset or power the board
14  void setup() {
15    // initialize digital pin 13 as an output.
16    pinMode(13, OUTPUT);
17  }
18
19  // the loop function runs over and over again forever
20  void loop() {
21    digitalWrite(13, HIGH);  // turn the LED on (HIGH is the voltage level)
22    delay(1000);             // wait for a second
23    digitalWrite(13, LOW);   // turn the LED off by making the voltage LOW
24    delay(1000);             // wait for a second
25  }

```

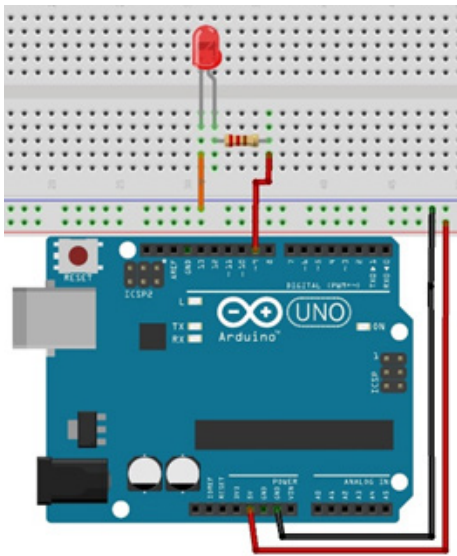
### Opdracht 3 Een LED dimmen

- Bouw de opstelling die hiernaast staat. Gebruik bij de output een PWM pin, bijvoorbeeld Pin 9 (let op, de LED hoeft niet aangesloten te worden aan de constante spanning, de spanning wordt nu geleverd door pin 9).
- Open het script Fade in de voorbeelden/basis en upload het script.
- Beschrijf wat je ziet en probeer met behulp van de code een te verklaren wat er gebeurt.
- Verander de code zodat de LED sneller volledig brandt en sneller uit is. Let op, er zijn twee manieren! Probeer ze allebei uit.
- In de code staat `analogWrite`. Voorheen hebben we `digitalWrite` gebruikt. Leg uit waarom dat nu niet kan.





## Opdracht 4 Een aan en uit knop voor de LED



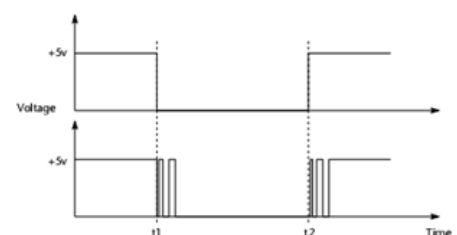
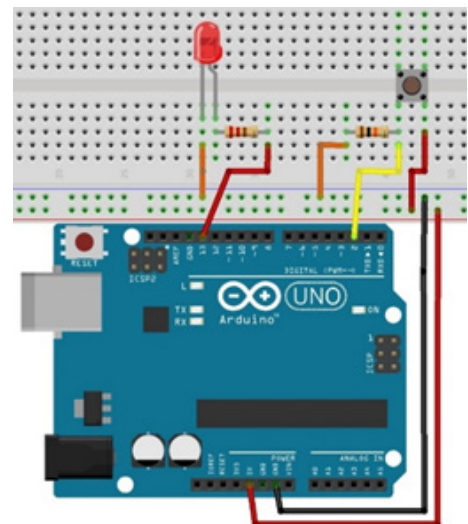
We kunnen nu de LED met behulp van de code aan en uit zetten en zelfs dimmen. Maar vaak wil je ook een schakeling handmatig aan en uit kunnen zetten. Daarvoor hebben we een drukknop nodig.

- Bouw de schakeling die hiernaast staat. Let op dat je een grote weerstand gebruikt zodat de te leveren stroom niet te groot is, een Arduino is namelijk slecht in het leveren van grote stroomsterktes.

Het idee is nu dat we pin 2 gebruiken als INPUT. Pin 2 meet de spanning op dat punt (vergelijkt deze met 0.0 V). Dit gebeurt met de code `digitalRead()`. Deze kan nu een hoog (5.0 V) of laag (0.0 V) signaal meten (met een analoge pin, A0 t/m A5, kunnen ook tussen gelegen waardes (10 bits) gemeten worden). Pin 2 meet alleen een spanning als de knop (button) ingedrukt wordt.

- Open het script Button via voorbeelden/digitaal en upload het script.
- Druk op de knop en controleer wanneer de LED uit is en wanneer deze aan is.
- Pas het script aan zodat de functie van de knop precies omgedraaid wordt.

**NB:** Met de button is er iets bijzonders aan de hand. Deze heeft namelijk een zogenaamde Bounce. Dit betekent dat de spanning niet direct van 0 V naar 5 V gaat maar, nog een keer op en neer gaat. Dit komt omdat er in de button een veer zit die op en neer gaat. Bij het gebruik en uitlezen van de button is het dus handig om een delay in te bouwen...



## Programmeren deel 2

Het script Button is uitgebreider dan we voorheen hebben gezien. We lopen stap voor stap door het script.

```
const int buttonPin = 2;
const int ledPin = 13;
```

We hebben te maken met twee poorten waar iets moet gebeuren. Pin 2 is een invoer en pin 13 moet een uitvoer zijn. De pin verandert niet, dit is een constante (`const`). Het is een gehele waarde, een integer (`int`). We geven pin 2 een herkenbare naam, pin 2 heet nu `buttonPin`. Pin 13 heeft de naam `ledPin` gekregen.

```
int buttonState = 0;
```

We willen straks weten wat de 'staat' van de knop is (ingedrukt of niet). Deze kan 1 of 0 zijn. We moeten even vertellen dat we de staat van de knop willen weten (aanmaken van een variabele) en deze straks willen vergelijken. Voordat we het script laten draaien is de `buttonState` gelijk aan 0.

```
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}
```

Zoals gezegd, we moeten even vertellen dat de `ledPin` een `OUTPUT` is en de `buttonPin` een `INPUT`. Zo, dat weten ze nu dan ook...

Er moet iets gebeuren als de knop wordt ingedrukt. Aan het begin van de loop wordt de knop uitgelezen (eigenlijk wordt er alleen gecontroleerd of er een 5.0V spanning over de weerstand staat). Daarna volgt een if-statement. Als (`if`) de knop is ingedrukt (`buttonState == HIGH`) dan moet de LED gaan branden (`digitalWrite(ledPin, HIGH);`). In alle andere gevallen (`else`), moet de LED niet branden (`digitalWrite(ledPin, LOW);`).

```
void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
}
```

```
void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH && state == LOW) {
    state2 = HIGH;
  }
  if (buttonState == HIGH && state == HIGH) {
    state2 = LOW;
  }
  state = state2;
  digitalWrite(ledPin, state);
  delay (100);
}
```

```
if(buttonState == HIGH){i++;}
digitalWrite(ledPin,i%2);
```

Het is een makkelijk if-statement. Je kunt ook meerdere voorwaarden stellen. Als je twee knoppen tegelijk in moet drukken voordat de LED gaat branden zet je && tussen de tekens (`if buttonState1 == HIGH && buttonState 2 == HIGH`). Je kunt ook het teken `||` gebruiken. Dan moet of de ene voorwaarde gelden of de andere. Helaas blijft het lampje nu nog niet branden als je de knop hebt ingedrukt. Je kunt er wel zelf voor zorgen dat je met één knop het lampje aan en uit kan zetten.

Vergeet niet de nieuwe variabele (`int`) te definiëren aan het begin! Er zijn nog eenvoudigere manieren, deze zijn alleen getoond als script. Leg uit hoe ze werken!

```
digitalWrite(13,!digitalRead(13));
```

## Opdracht 5 Het beveiligen van een kluis

De kluis van de bank is beveiligd. De kluis gaat pas open (de LED brandt pas) als er twee sleutels worden omgedraaid (knoppen ingedrukt). Een sleutelgat bevindt zich op de kamer van de directeur, de andere zit naast de kluis.

Bouw de bijbehorende schakeling en programmeer het script zodanig dat de kluis pas open gaat als beide sleutels tegelijk worden omgedraaid.

## Opdracht 6 Een wisselschakeling

Als je onderaan de trap staat wil je het licht bovenaan de trap aan zetten. Bovenaan de trap zit nog een lichtknop. Daarmee kan je het licht weer uit of aan zetten. Bouw de schakeling met twee knoppen en schrijf het programma zodat je een werkende wisselschakeling hebt. Lukt dit niet meteen? Probeer het dan eerst met een enkele knop.

## Opdracht 7 Een voetgangersoversteekplaats

Voor voetgangers is een speciale overgangsplek gemaakt. Deze plek ligt aan een drukke weg. Als er geen voetgangers zijn, staat het verkeerslicht voor de auto's op groen. Op het moment dat een voetganger over wil steken, drukt deze op de knop. Het verkeerslicht voor de auto's gaat dan van groen naar geel naar rood. Voetgangers hebben dan 15 seconde om over te steken. Het verkeerslicht voor hen staat op groen! Dan gaat het groene lampje vijf keer knipperen en springt op rood. Een seconde later springt het verkeerslicht voor de auto's weer op groen.

Maak dit systeem.