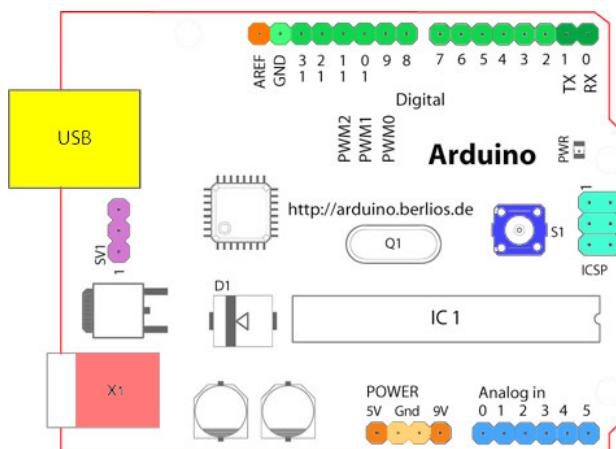


# ABC GIDS

**Jouw gids voor de Arduino Beginners Cursus met Arduino Uno**



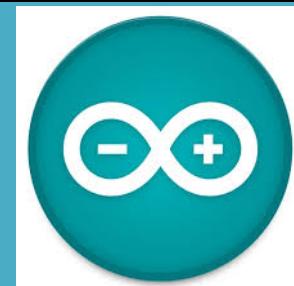
## Inhoudsopgave

### *Welkom bij de Arduino Beginners Cursus*



De Arduino Beginners Cursus is een begin om kennis te maken met electronica samen met de Arduino Uno. Deze gids bevat alle informatie, die je nodig hebt om 15 projecten te onderzoeken en te bouwen. De achterliggende filosofie is - dat iedereen met electronica kan en zou moeten spelen. Als je deze gids hebt doorgewerkt, heb je voldoende know-how om je eigen projecten en experimenten te creëren. Genoeg gezegd, laten we beginnen!

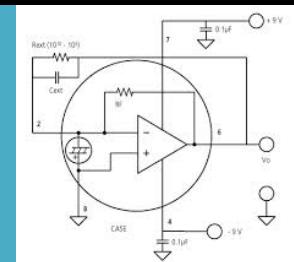
## Sectie 1:



## Begin

Wat is de Arduino Uno platform?	4
Arduino IDE downloaden	6
Drivers installeren	7
Hardware identificeren	9
"ABC Gids Code" downloaden	11

## Sectie 2:



## Projecten

De wereld draait op circuits	12
Inventaris van onderdelen	14
Arduino Uno board	16
Breadboard	18
Circuit #1 - Het eerste circuit: een knipperende LED	21
Circuit #2 - Potentiometer	27
Circuit #3 - RGB LED	31
Circuit #4 - Meerdere LEDs	35
Circuit #5 - Drukknoppen	39
Circuit #6 - Foto-weerstand	43
Circuit #7 - Temperatuur sensor	47
Circuit #8 - Een enkele servo-motor	51
Circuit #9 - Flex sensor	55
Circuit #10 - Zachte potentiometer	59
Circuit #11 - Piëzo element	63
Circuit #12 - Een motor laten draaien	67
Circuit #13 - Relais	71
Circuit #14 - Shift register	75
Circuit #15 - Liquid crystal display (LCD)	79
Circuit #16 - Simon Says	83

## Wat is de Arduino Uno platform?



### De DIY Revolutie

We leven in een unieke tijd waarin we toegang hebben tot allerlei bronnen, waarmee we onze eigen oplossingen en ontdekkingen kunnen creëren. De DIY revolutie bestaat uit hobbyisten, denkers en uitvinders, die liever zelf hun projecten ontwikkelen dan dat aan andere overlaten.

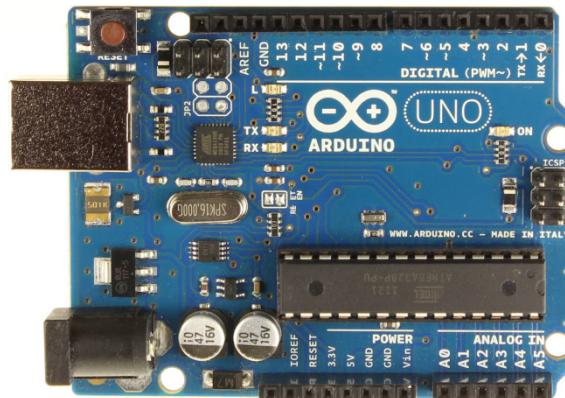
### Een computer voor de fysieke wereld

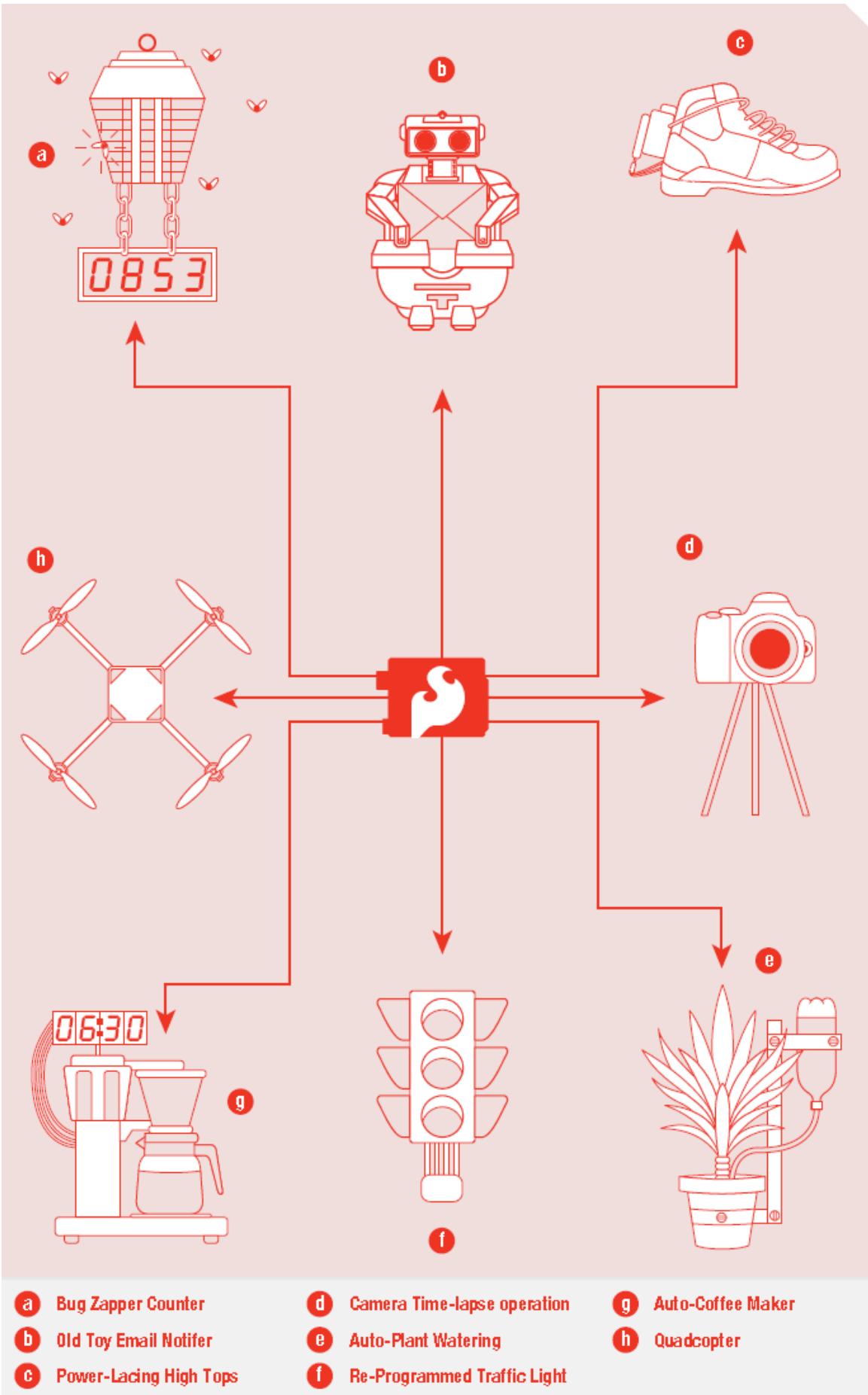
De Arduino Uno is jouw ontwikkelplatform. Je zou kunnen zeggen, dat het een kleine, draagbare computer is. Het kan worden geprogrammeerd, maar accepteert ook **inputs** (zoals het indrukken van een knop of het uitlezen van een lichtsensor), en interpreteert de informatie om verschillende **outputs** te activeren (knipperen van een LED of het laten draaien van een motor).



### Het Arduino Uno board

Arduino Uno is één van de vele ontwikkelboards, dat gebaseerd is op de ATmega328. Het heeft 14 digitale input/output pins (6 als PWM outputs), 6 analoge inputs, een 16 MHz kristal-oscillator, een USB connectie, een power-plug, een ISP-header en een reset-knop. Wees gerust, later leer je hier alles over.





## Download de Arduino IDE (Integrated Development Environment)



### Toegang tot internet

Om met de Arduino Uno te kunnen werken, heb je de nieuwste versie van de Arduino software nodig van [www.arduino.cc](http://www.arduino.cc) (het is gratis!). Deze software, bekend als Arduino IDE, maakt het mogelijk om de Arduino te programmeren en te laten doen wat jij wilt. Het is net een word-processor om programma's te schrijven. Maak een verbinding met het internet met jouw favoriete browser en typ het volgende URL adres in:



1

#### Download

Kies het geschikte operating system van de computer.

- + Windows
- + Mac OS X
- + Linux 32 bit, 64 bit
- + source

 Buy Download Products ▾ Learning ▾ Reference Support ▾ Blog LOG IN SIGN UP

## Arduino 1.5.8 BETA (with support for Arduino Yún and Arduino Due boards)

If you have the Arduino Yún or Due you must download the 1.5.8 version. Refer to the [Yun getting started page](#), or [Due getting started page](#) for specific details about those boards. Please discuss any issues in the [Yún forum](#) or [Due forum](#).

**Download**

Arduino 1.5.8 ([release notes](#)):

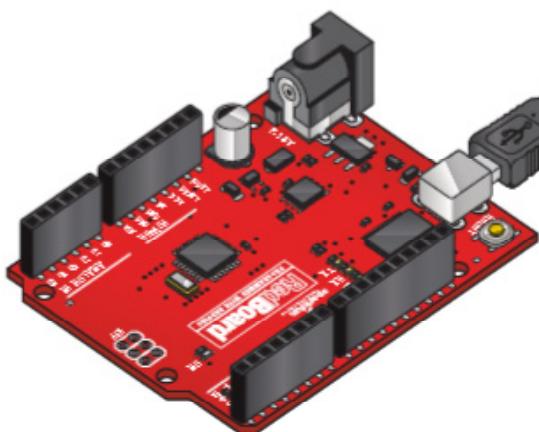
- Windows: [Installer](#)
- Windows: [ZIP file](#) (for non-administrator install)
- Mac OS X: [ZIP file for Java 6](#)
- Mac OS X: [ZIP file for Java 7](#). NOTE: the ZIP for Java 7 is experimental and it works only on OSX 10.7 or greater. If you experience problems running it please use the Java 6 version.
- Linux: [32 bit, 64 bit](#)
- [source](#)



Kies het geschikte operating systeem voor jouw computer.

## Verbind de Arduino Uno met jouw computer

Gebruik de geleverde USB kabel in het ABC-pakket om de Arduino te verbinden met één van de USB poorten van jouw computer.



2

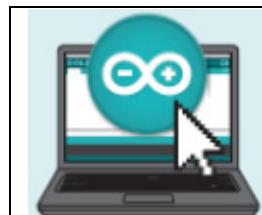
3

## Installeer de Arduino drivers

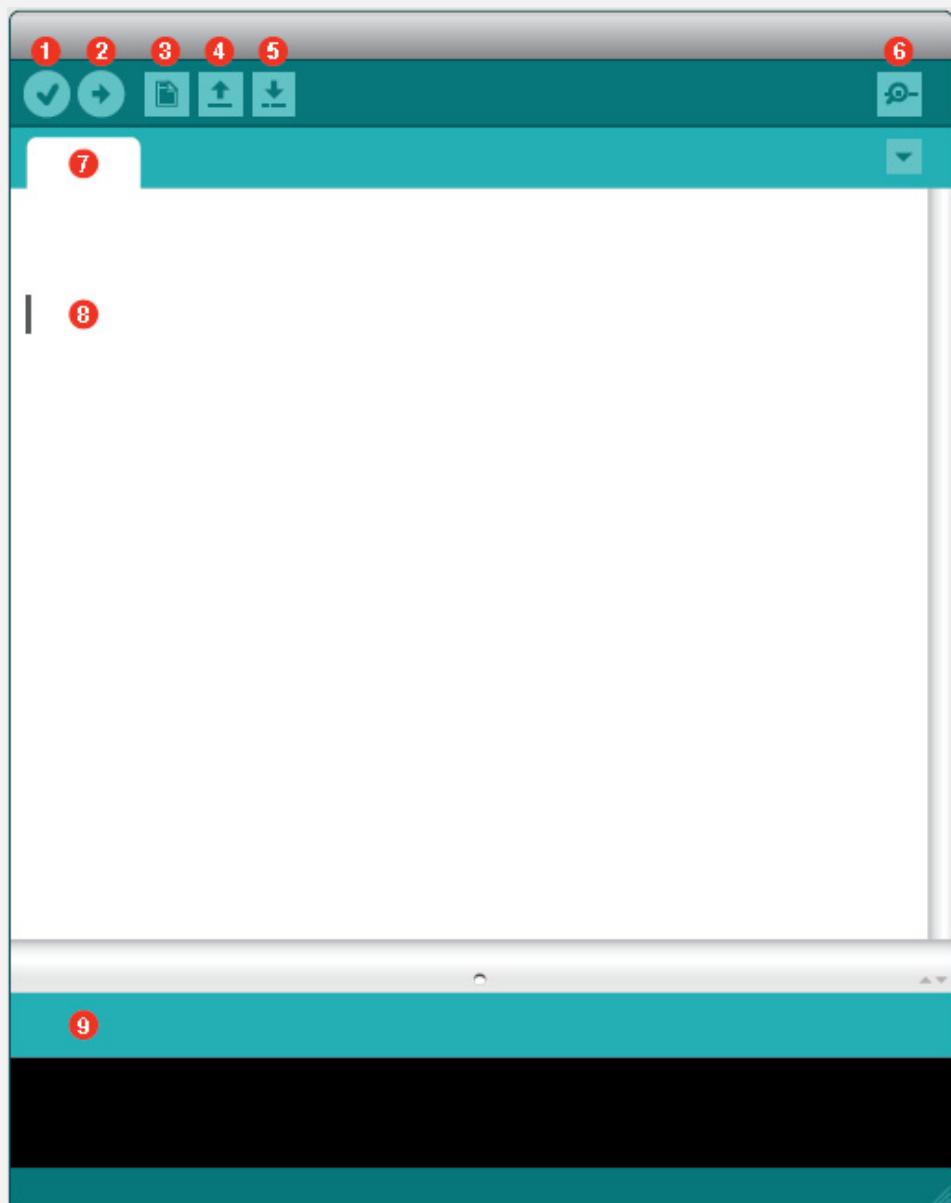
Dit is afhankelijk van jouw computer systeem. Windows installeert FTDI drivers niet automatisch. De drivers zijn nodig om met Virtual COM ports te werken. Ga naar de [FTDI website](#), selecteer 'Drivers' en 'VCP Drivers'. Dit staat voor Virtual Com Port.



## Open de Arduino IDE



Open de Arduinoo IDE software op jouw computer. Bekijk de menu's om bekend te raken met deze interface. We gaan nog geen programma's schrijven, dit is een introductie. Hier gaan we de IDE instellen, zodat het Arduino Uno board kan worden herkend.



## GUI (Graphical User Interface)

- 1** Verifieer: compileert en test de code op fouten (zoals vergeten puntkomma's, haakjes etc.)
- 2** Uploaden: compileert de code en zend het naar de Arduino. LEDs gaan snel knipperen.
- 3** Nieuw: dit opent een nieuw venster tab.
- 4** Open: een bestaande sketch openen.
- 5** Opslaan: bewaren van de actieve sketch in het Arduino IDE venster.
- 6** Seriële monitor: opent een nieuw communicatie-venster; gemakkelijk bij debugging.
- 7** Sketch naam: dit is de naam van de sketch waarmee je aan het werk bent.
- 8** Code venster: hier wordt de code van de sketch weergegeven.
- 9** Bericht venster: dit venster toont eventueel de compilatie en foutmeldingen.

**De drie meest gebruikte commando's voor deze gids staan hieronder:**

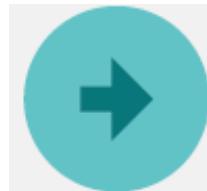
*Open*



*Verifieer*

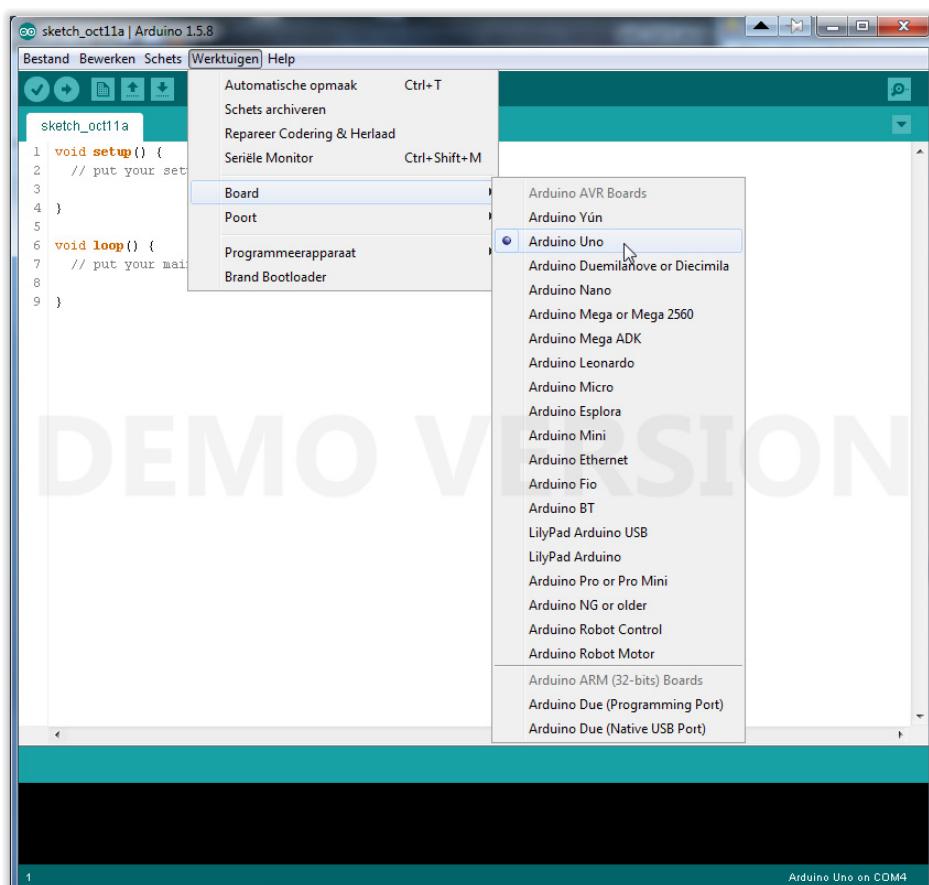


*Uploaden*

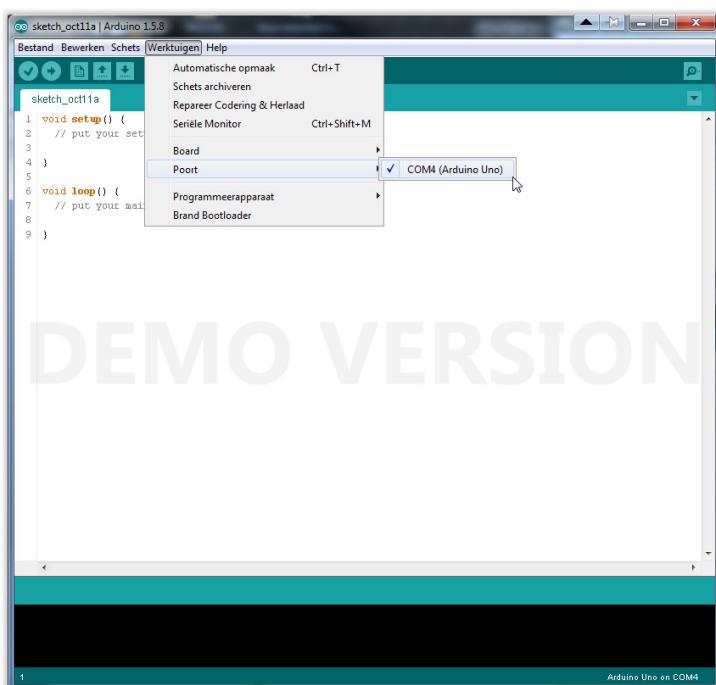


# 4

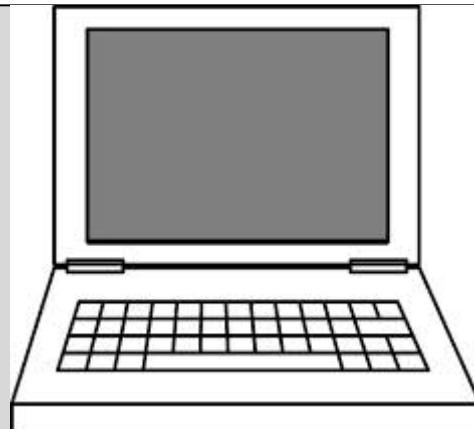
Selecteer jouw board: Arduino Uno



Selecteer de poort via  
**Werktuigen | Poort** menu. Dit  
zal waarschijnlijk **COM3 of hoger**  
zijn. COM1 en COM2 zijn meestal  
gereserveerd voor andere  
hardware op de computer.



# 5



Gebruik de volgende URL om de codes te downloaden:



[www.quillby.nl/vindigo/electronics/abc-codes](http://www.quillby.nl/vindigo/electronics/abc-codes)

	Computer  C:  Program Files (x86)  Arduino1.5.6  examples
Unzip het bestand "ABC-codes". Dit bevindt zich in de 'Download' folder van jouw browser.	Kopieer de "ABC-gids Code" in de Arduino's folder "examples"

## Beginnen met circuits



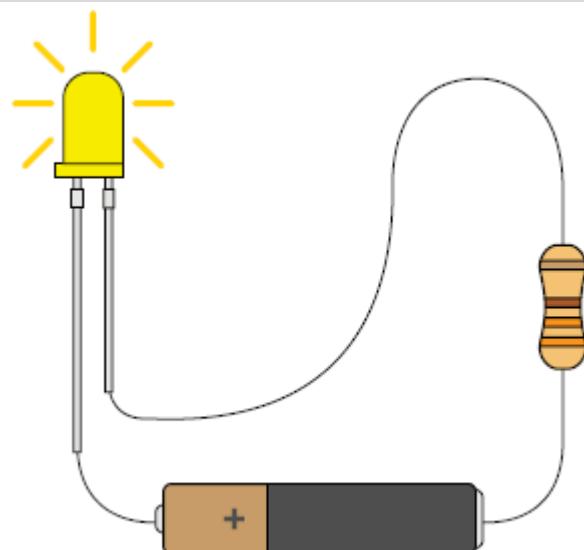
## *Wat is een elektronisch circuit?*

Een circuit is een elektrische loop met een start- en eindpunt en een aantal componenten daartussen. Circuits kunnen weerstanden, diodes, inductoren, transistors, sensors, motoren en andere componenten bevatten.

Circuits kunnen worden opgedeeld in verschillende categorieën - analoge circuits, digitale circuits of een mix van beide. In deze gids worden alle drie types behandeld.

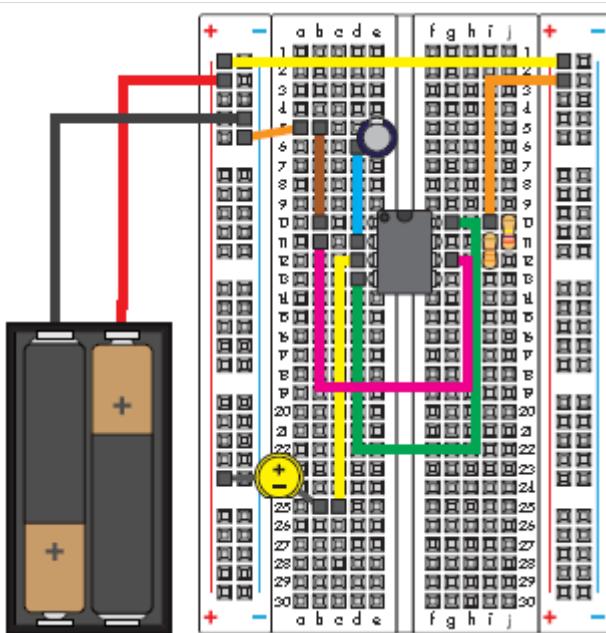
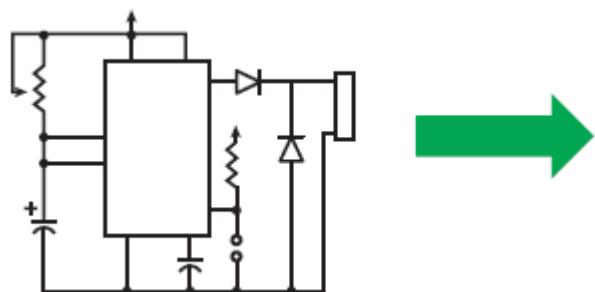
## ***De wereld draait om circuits:***

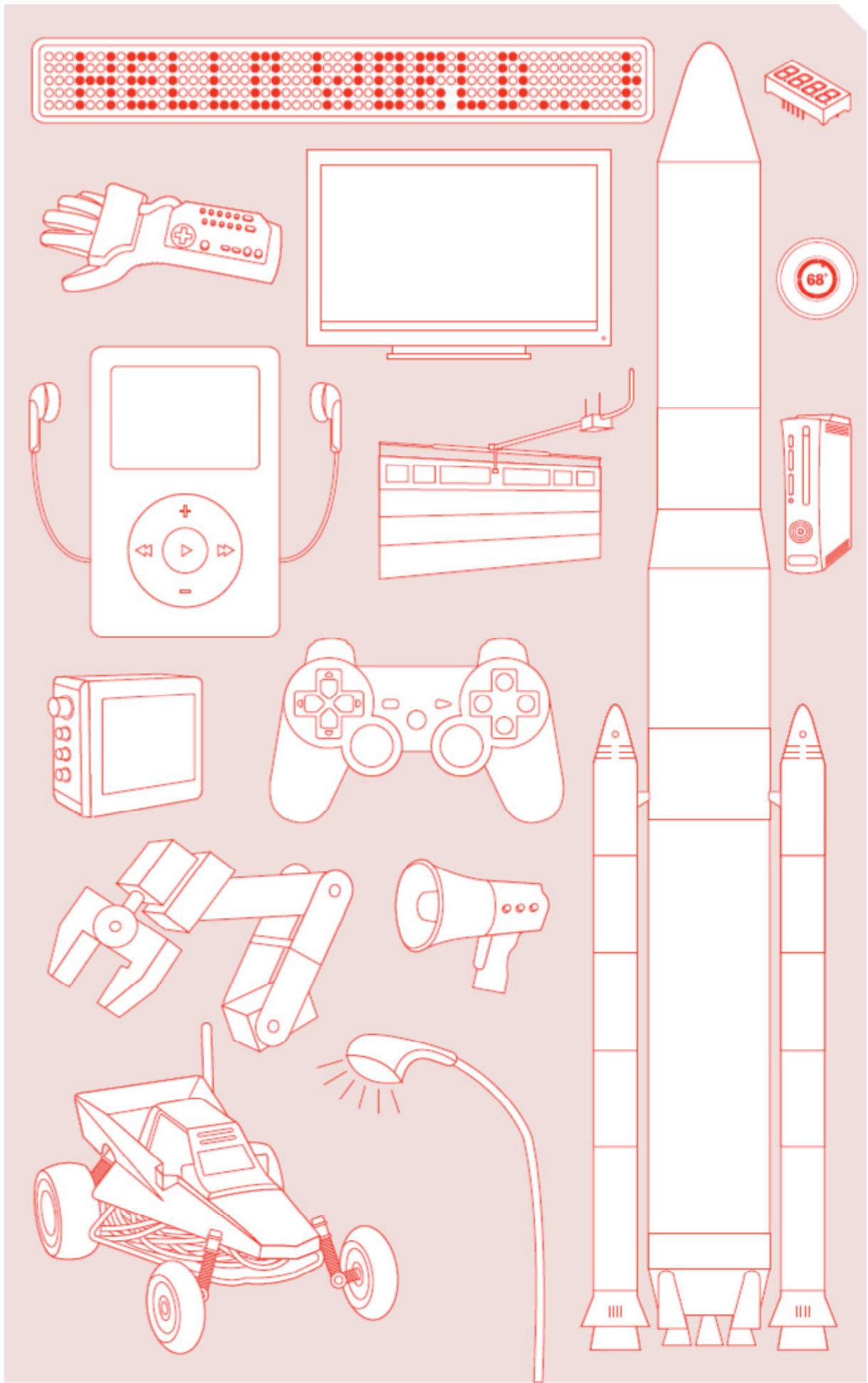
Overal waar je kijkt, vind je circuits. In jouw mobiele telefoon, de computer, auto's, video game consoles - ze zitten tjokvol electronica. In deze gids ga je experimenteren met enkele simpele circuits en leer je de wereld van de 'embedded electronics' kennen.



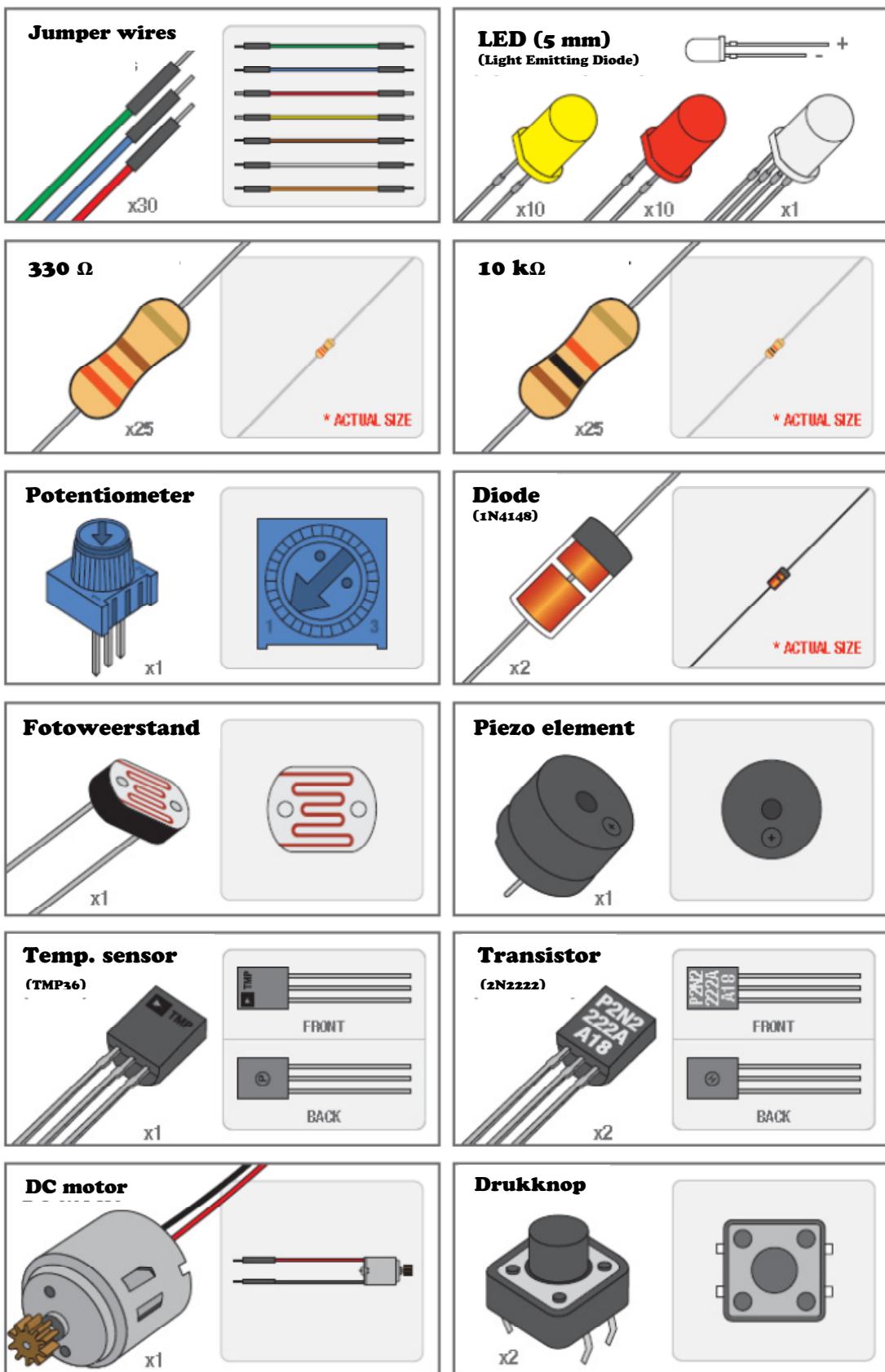
## *Simpele en complexe circuits*

In deze gids gebruiken we uitsluitend simple circuits, maar dat betekent niet, dat je geen leuke dingen kunt doen met deze tools. Als je alle experimenten hebt uitgevoerd en kennis hebt opgebouwd omtrent circuits, kun je verdere stappen nemen met meer complexere circuits en je fantasie de loop laten.



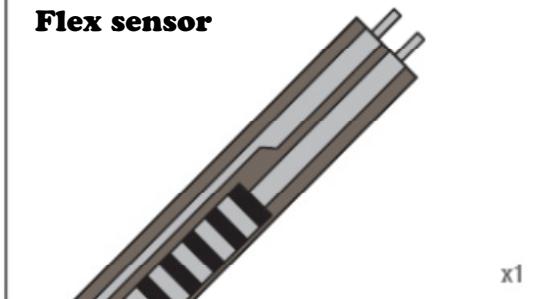


## Onderdelenlijst

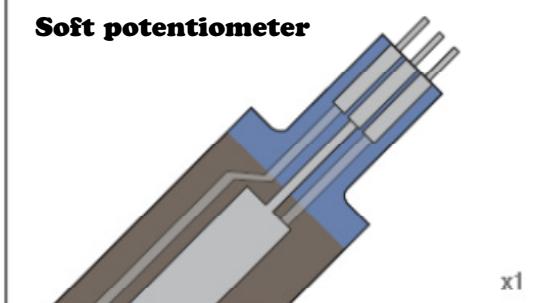


## Onderdelenlijst (vervolg)

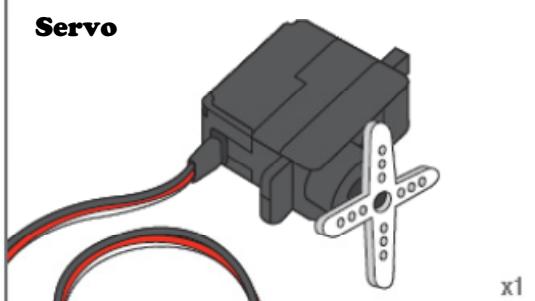
**Flex sensor**



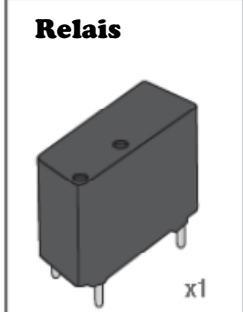
**Soft potentiometer**



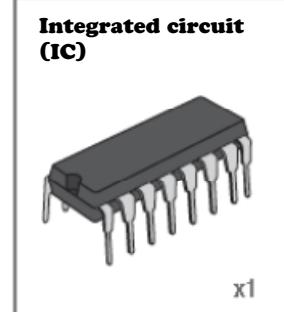
**Servo**



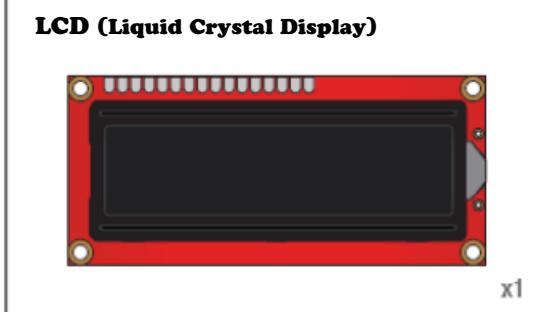
**Relais**



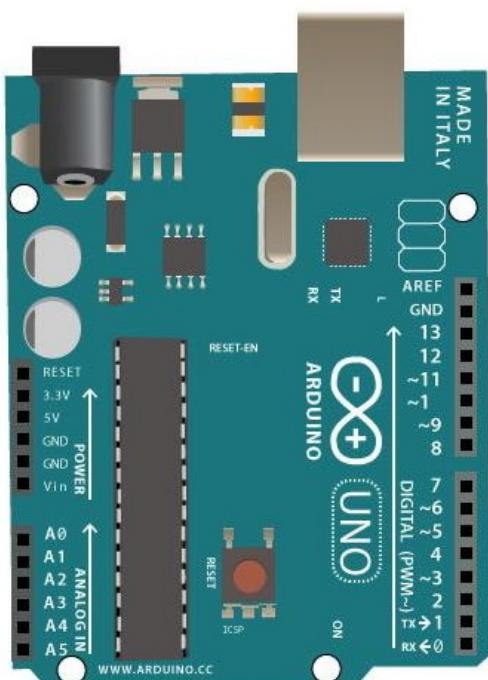
**Integrated circuit (IC)**



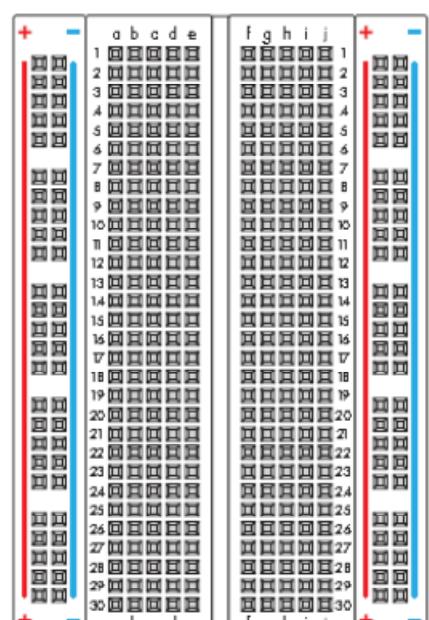
**LCD (Liquid Crystal Display)**



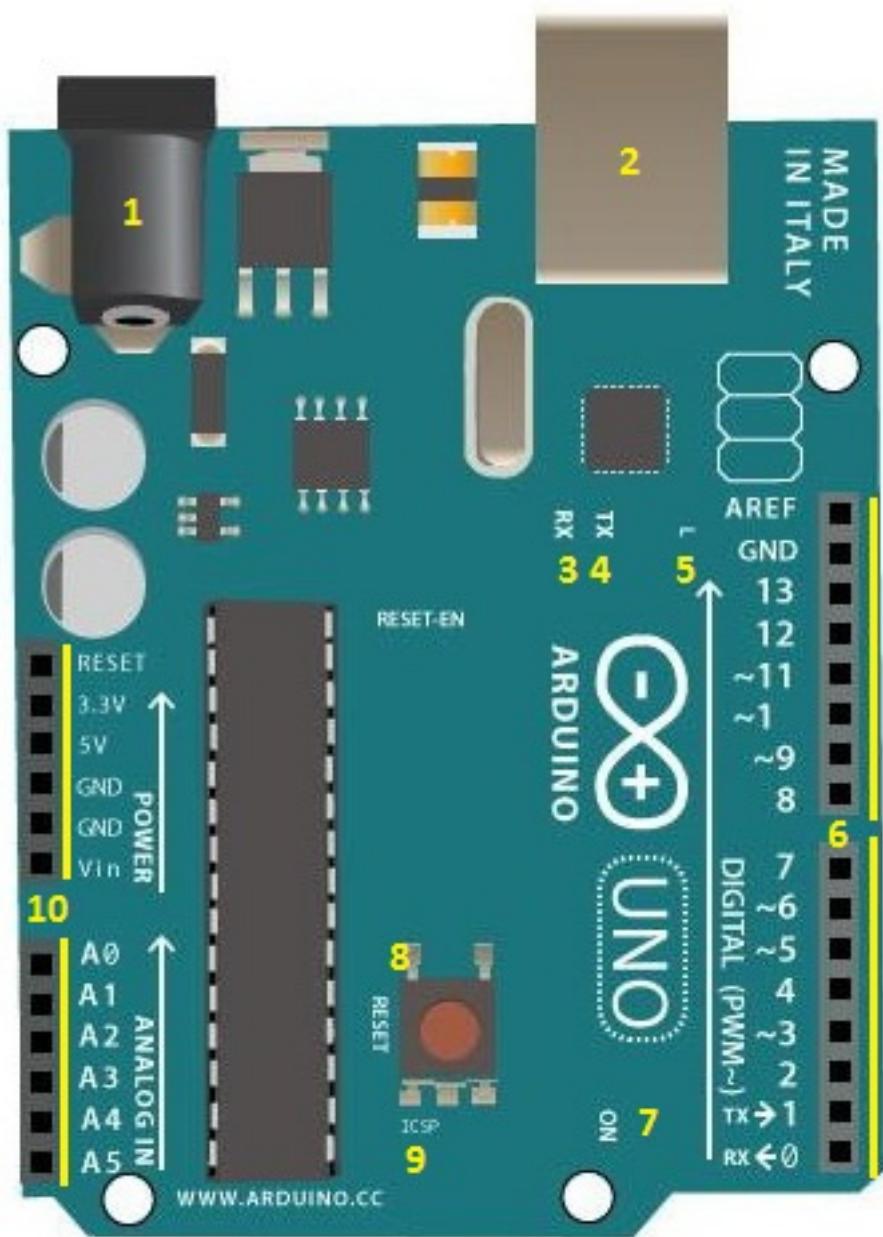
**Arduino Uno Board**



**Breadboard  
(Standaard soldeerloos)**

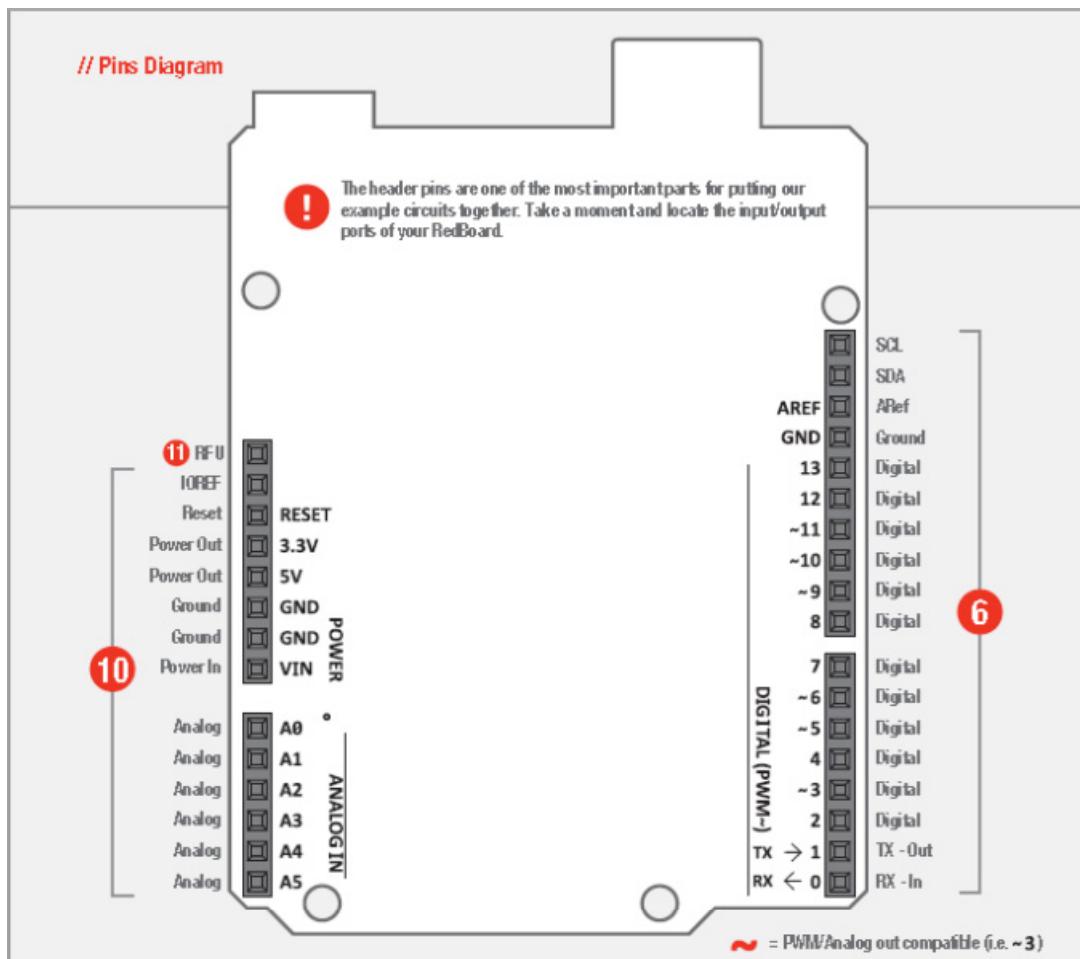


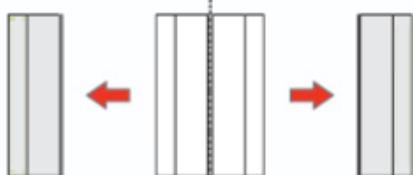
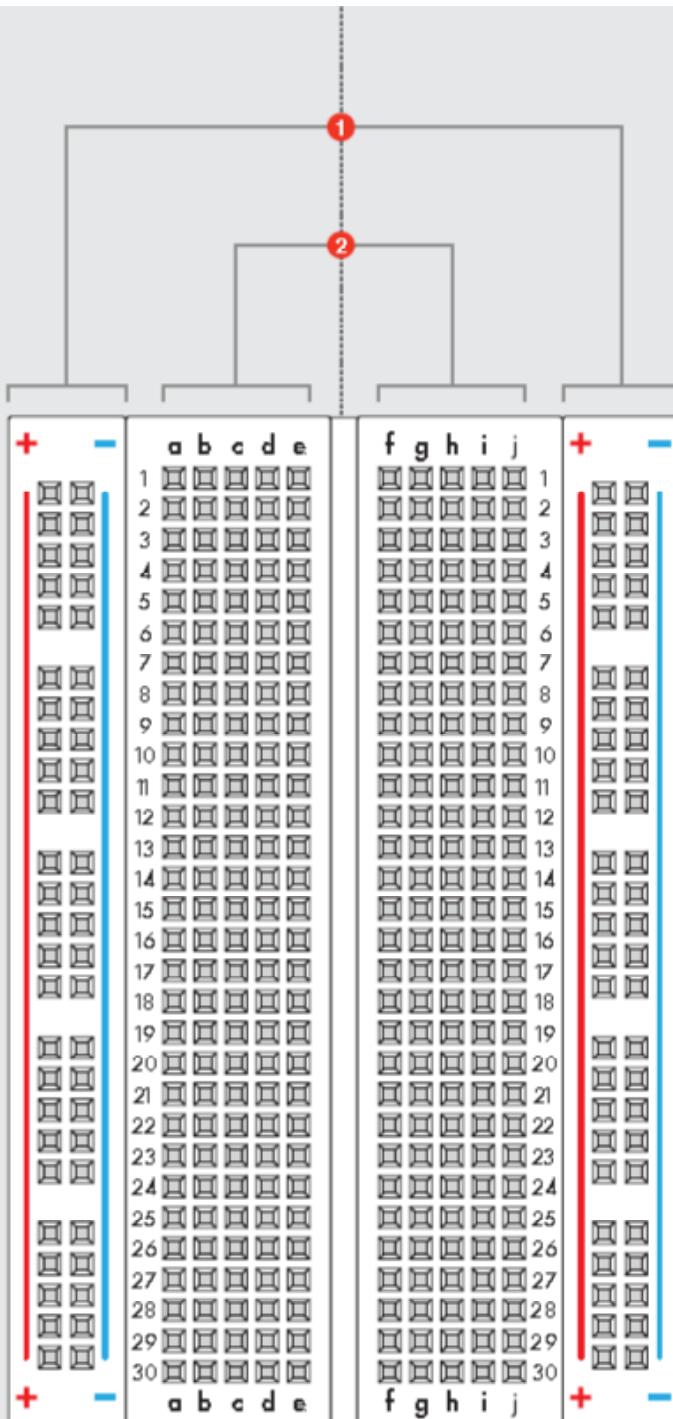
x1



## ABC Arduino Uno board

- 1 Voedingsplug: kan worden verbonden met een 9-12V DC-voeding of batterijen.
- 2 USB poort: verzorgt de voeding als ook de communicatie tussen computer en board via USB.
- 3 LED (RX-receiving): dit knippert wanneer de FTDI chip data ontvangt.
- 4 LED (TX-transmitting): dit knippert wanneer de FTDI chip data stuurt.
- 5 LED (Pin 13-troubleshooting): dit geeft aan of het programma goed werkt (ingebouwd).
- 6 Pins (ARef, GND, digitaal, Rx, Tx: pins voor inputs en outputs en aarde.
- 7 LED: geeft aan of het board voeding krijgt.
- 8 Reset knop: hiermee kan het board handmatig worden gereset en start de sketch opnieuw.
- 9 ICSP pins: dit is voor "In-Circuit-Serial-Programming" en omzeilt de bootloader.
- 10 Pins (Analog In, Voeding In, GND, Voeding Uit, Reset).
- 11 RFU: deze pin is gereserveerd voor toekomstig gebruik.





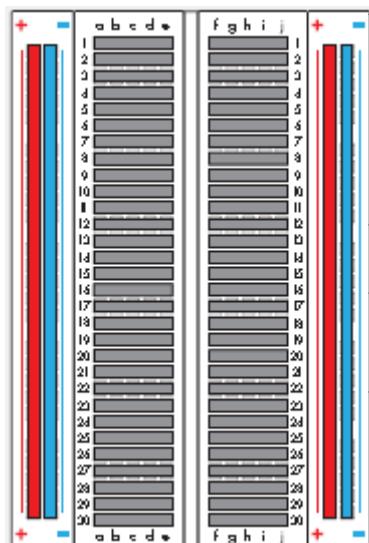
Deze lijn verdeelt het board in tweeën en scheidt de electriciteit voor de ene of de andere zijde.

## Breadboard

**1** Verticale verbinding (+ voeding en - ground) voeding lijnen.

**2** Horizontale verbinding (a-e & f-j)

Hoe wordt alles verbonden?



**Voeding:**

Bij iedere + teken loopt de voeding door de verticale kolommen.



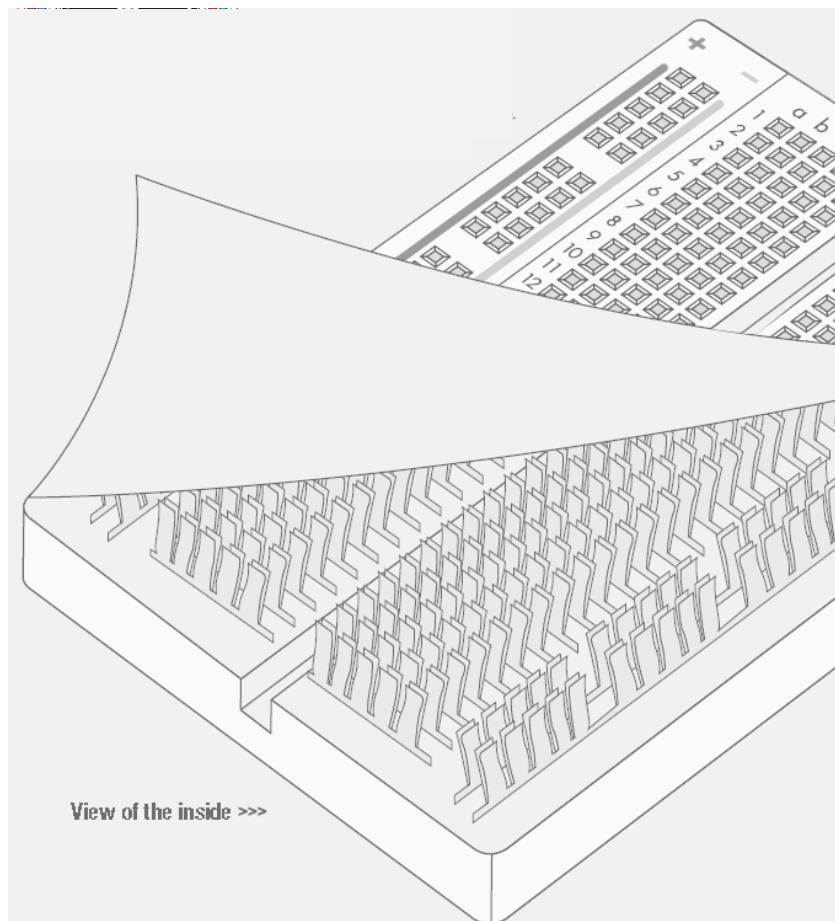
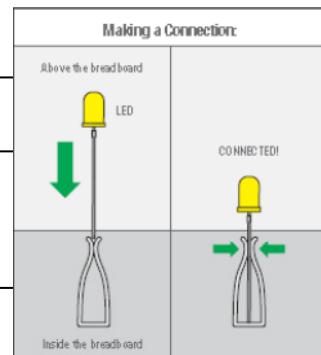
**Ground:**

Bij ieder - teken loopt aarde door de verticale kolommen.



**Horizontale rijen:**

Elke rij bestaat uit met elkaar verbonden verbindingen.  
Componenten in dezelfde rij worden met elkaar verbonden.



View of the inside >>>

## Circuit #1 - Jouw eerste circuit

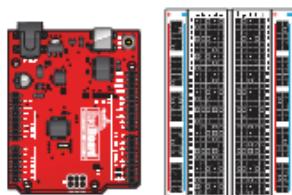
Hoe werkt het:

### 1 ASSEMBLE

### 2 WRITE

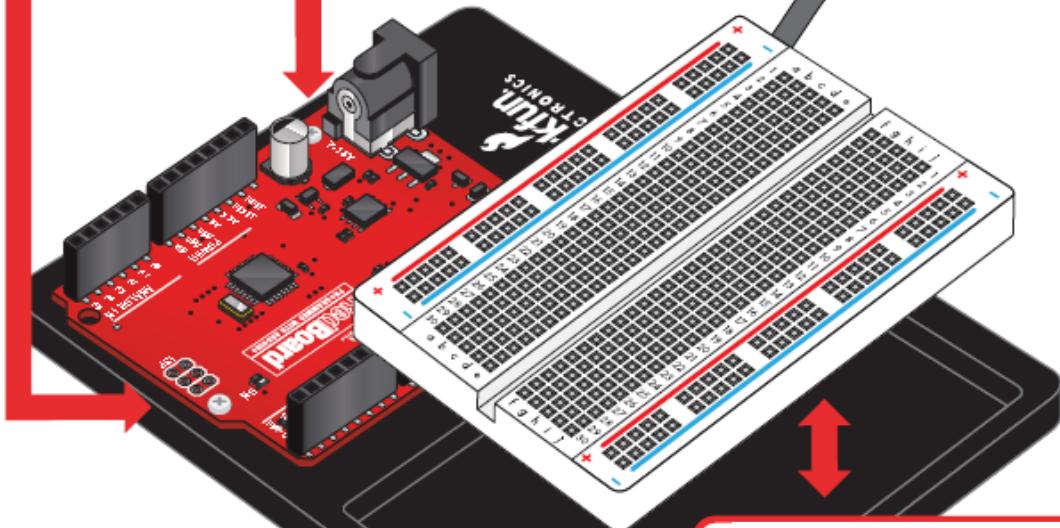
### 3 UPLOAD

- + Zorg ervoor dat de tekst leesbaar is op de Arduino en breadboard.



- + Verbind USB kabel.

- + Schroef het Arduino board vast.



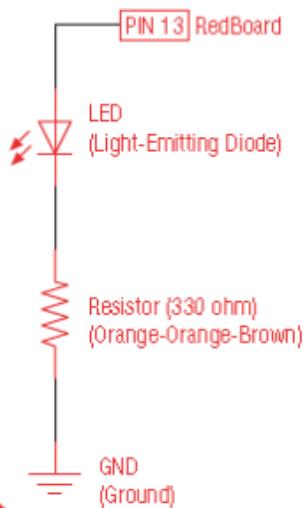
- + Verwijder het folie van de achterzijde van het breadboard en plak het vast.



Het Arduino board runt op 5V. Dit is de voeding, dat via de USB poort van de computer wordt geleverd en alle componenten van voeding voorziet. Door de Arduino aan je computer te koppelen, weet je dat de juiste spanning aanwezig is. De circuits kun je gewoon aanraken, omdat 5V niet gevaarlijk voor je is. De Arduino kan ook met een externe DC-voeding worden gevoed. De spanning moet liggen tussen 7-15 VDC.

## Circuit #1

1



+ Schema van het circuit.

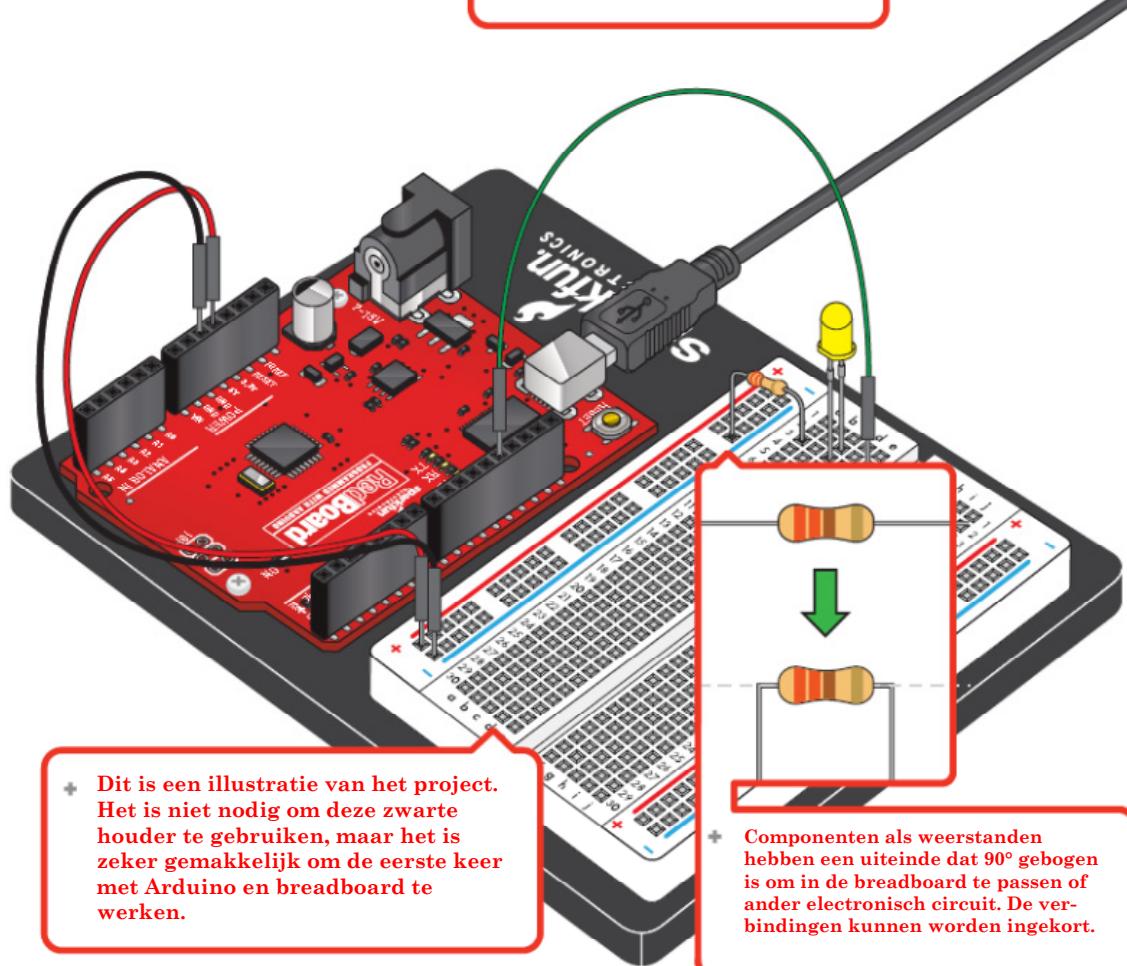
### Knipper een LED

LEDs (light-emitting diode) zijn kleine, krachtige lampjes, die in verschillende applicaties worden gebruikt. Als start met ABC, beginnen we met het knipperen van een lamp. Het is net zo simpel als het in- en uitschakelen van een lamp. Het lijkt wellicht niet veel, maar het geeft wel een goede start om kennis te maken met electronica, breadboard en programmering van het Arduino board. Deze fundering zal je later helpen om complexere experimenten om te gaan.

+ Elk circuit begint met een korte beschrijving over wat in elkaar wordt gezet en het verwachte resultaat.



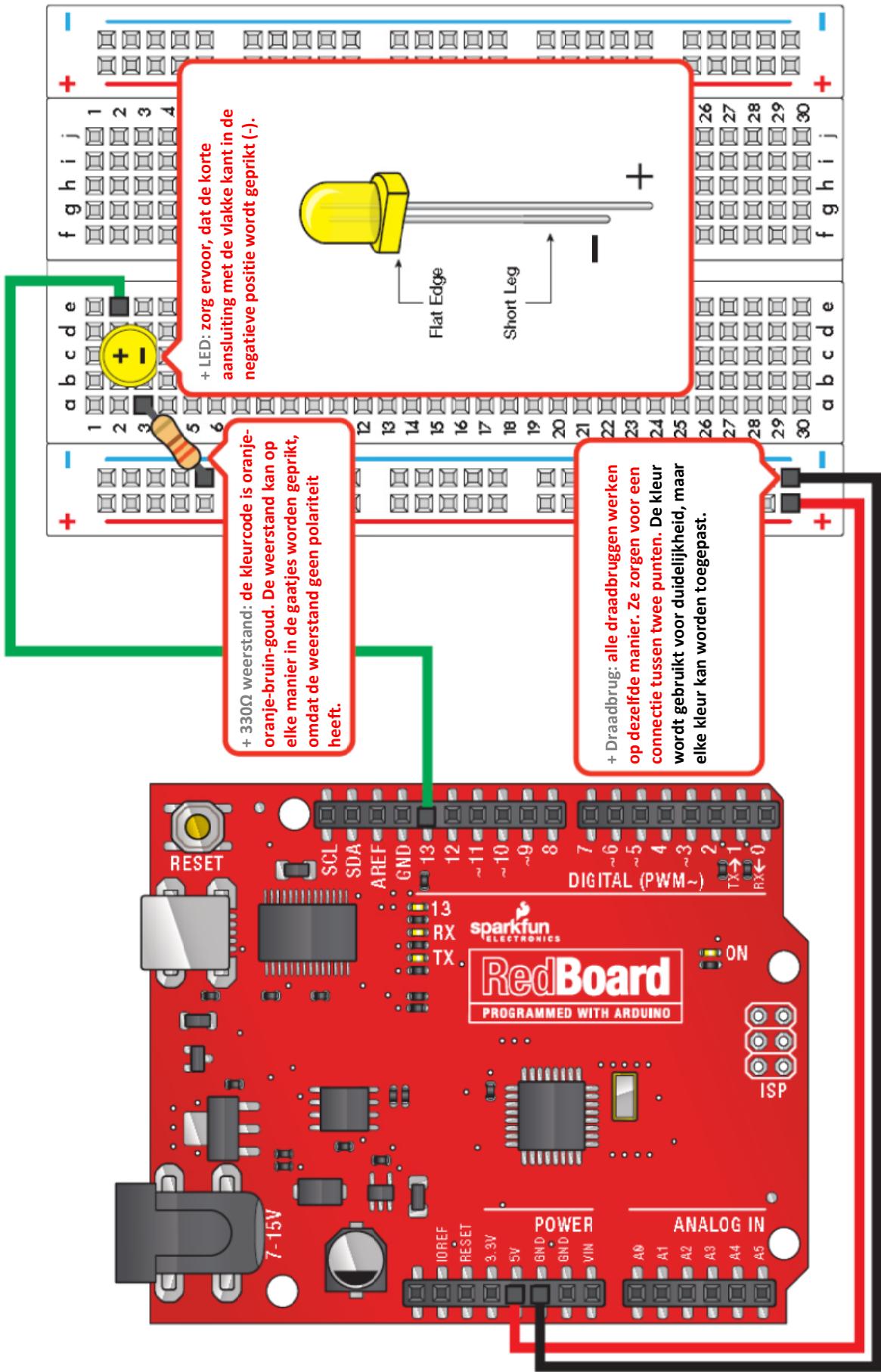
+ Lijst met componenten voor dit project.



+ Dit is een illustratie van het project. Het is niet nodig om deze zwarte houder te gebruiken, maar het is zeker gemakkelijk om de eerste keer met Arduino en breadboard te werken.

+ Componenten als weerstanden hebben een uiteinde dat 90° gebogen is om in de breadboard te passen of ander elektronisch circuit. De verbindingen kunnen worden ingekort.

Circuit 1: Knipperen van een LED



Component:	Afbeelding:						
LED (5 mm)			+ -	c2 c3	+ Componenten als LEDs worden in de breadboard gestoken. Let op de polarisatie van het component. Dit wordt met een waarschuwingsschijnhoek aangegeven.		
330Ω weerstand			a3 -		+ Weerstanden worden in de gaatjes van het breadboard gestoken. Het "-" symbool geeft verticale voedingslijn aan.		
Draadbrug			GND		+ "GND" op het Arduino board wordt verbonden met de "-" op het breadboard.		
Draadbrug			5V		+ "5V" op het Arduino board wordt verbonden met de "+" op het breadboard.		
Draadbrug			Pin 13	e2	+ "Pin 13" op het Arduino board wordt verbonden met "e2" op het breadboard.		
					+ <b>Breadboard:</b> de witte achtergrond geeft de verbindingen op het breadboard aan. De positie wordt aangegeven met een letter-cijfer combinatie, zoals e2. Dit is slechts een suggestie, dat overeenkomt met de grafische suggestie.		
					+ <b>Arduino:</b> de zeeblauwe achtergrond geeft een verbinding aan met een header pin op de Arduino.		



## Open jouw eerste Sketch:



Open de Arduino IDE software op jouw computer. De code in de Arduino taal geeft controle over de circuit. Open de code voor Circuit #1 uit de ABC-gids Code, dat je hebt gedownload en geplaatst hebt in de "examples" Arduino folder.

Bestand → Voorbeelden → ABC-gids Code → Circuit\_01

// Circuit #1

```
void setup()
{
pinMode(13, OUTPUT);
}

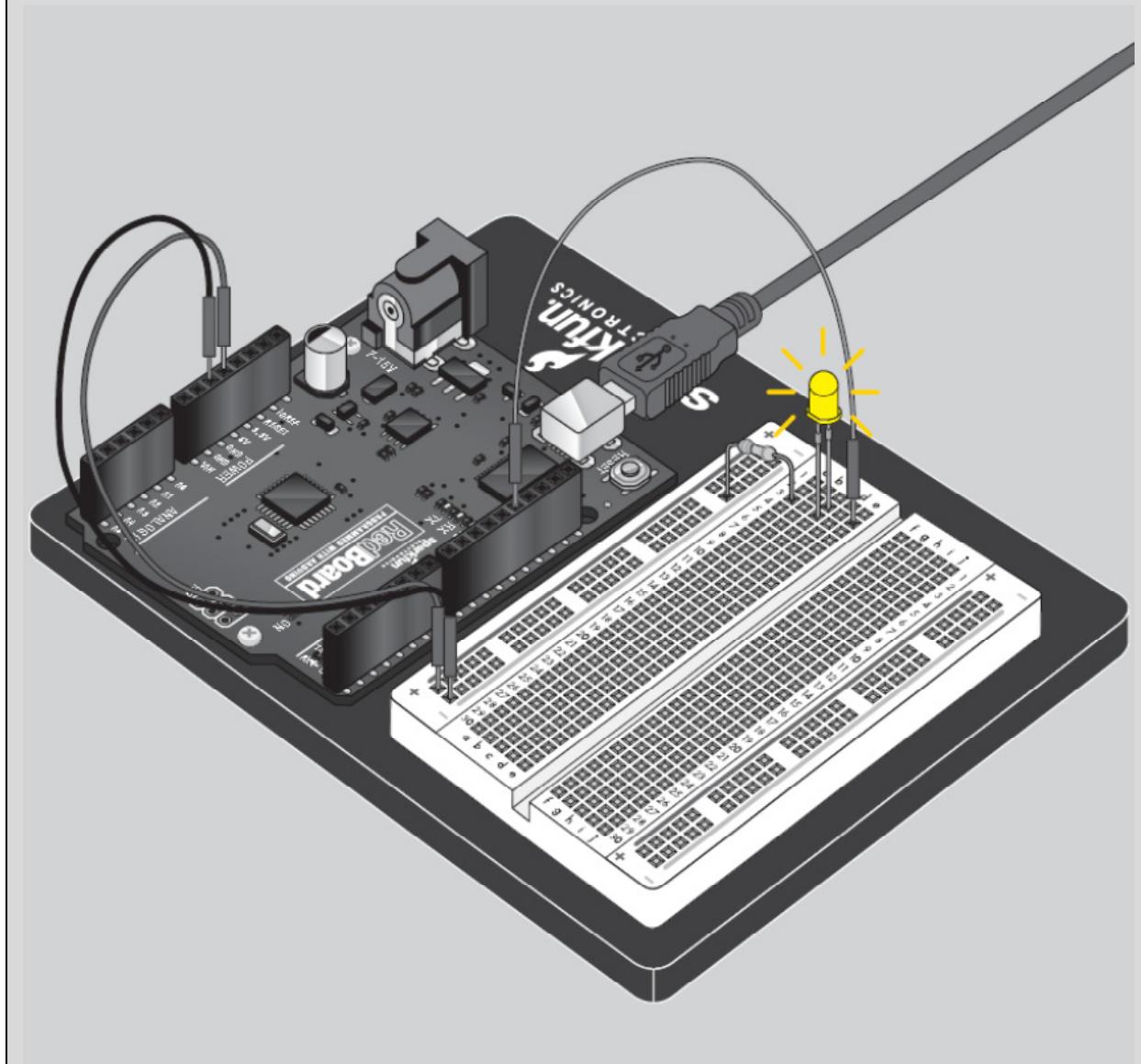
void loop()
{
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
}
```

Circuit\_01 §

```
1 /*
2 ABC Componenten Kit
3 Voorbeeld sketch 01
4
5 Knipperen van een LED
6
7 Schakelt een LED in gedurende 1 seconde, en uit gedurende 1 seconde;
8 dit wordt constant herhaald.
9 */
10
11 void setup()
12 {
13 // initialiseer de digitale pin als output
14 // Pin 13 zit bij de meeste Arduino's vast aan een LED.
15
16 pinMode(13, OUTPUT);
17
18 // By the way, the Arduino offers many useful built-in functions
19 // like this one. You can find information on all of them at the
20 // Arduino website: http://arduino.cc/en/Reference
21 }
22
23 void loop()
24 {
25 digitalWrite(13, HIGH); // Turn on the LED
26 delay(1000); // Wait for one second
27 digitalWrite(13, LOW); // Turn off the LED
28 delay(1000); // Wait for one second
29 }
30
```

 Verify	Dit compileert de code. De IDE verandert het van tekst naar instructies, die de computer begrijpt.
 Upload	Dit zendt de instructies via de USB kabel naar de computer chip op het Arduino board. De code wordt automatisch gestart.

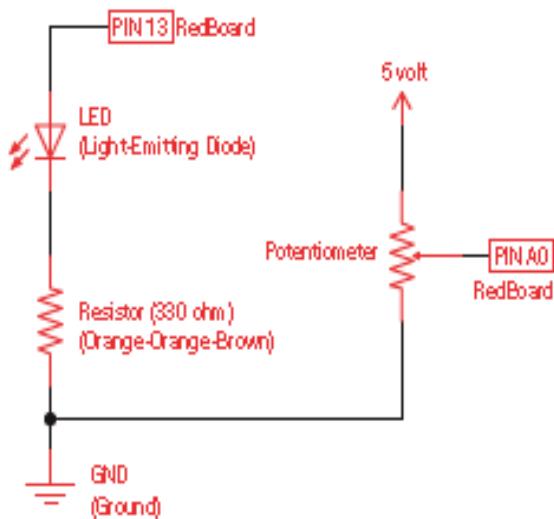
// Het resultaat van een compleet circuit met correcte code na verificatie en upload.



<p><b>1</b>  Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → <a href="#">Circuit_01</a></p> <p>Notitie code: + Hoe werkt de Arduino code.</p>	
	<p>+ Hier vind je de Arduino code voor elk circuit.</p>
	<p>+ Vergeet niet de code te verifiëren en te uploaden.</p>
<p><b>pinMode(13, OUTPUT);</b></p> <p><b>digitalWrite(13, HIGH);</b></p> <p><b>Wat gebeurt er nu:</b></p>	<p>Voordat je een pin van de Arduino board kunt gebruiken, moet je eerst aangeven of het een INPUT of OUTPUT pin is. We gebruiken hiervoor een ingebouwde "functie" pinMode() om dat te doen.</p> <p>Als een pin als OUTPUT wordt gebruikt, dan is het mogelijk de pin HIGH (5 volts) of LOW (0 volts) te schakelen.</p> <p>+ Controleer of het circuit compleet is en werkt.</p>
<p>Je zou nu de LED moeten zien knipperen als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven.</p>	 <p>+ Hier kun je een oplossing vinden voor de meest voorkomende fouten bij het in elkaar zetten van het circuit.</p> <p>+ Hier vind je voorbeelden van applicaties in de 'real world' van het circuit. Theorieën gebruikt in deze circuits vindt men in <a href="#">alledeasele dingen</a>.</p>
<p><b>Tips:</b></p> <p>LED licht niet op: LEDs werken slechts in één richting. Het is een component met polariteit. Verwijder de LED en draai het 180°. Je hoeft niet bang te zijn, dat de LED stuk gaat.</p> <p>Programma upload lukt niet: Dit kan weleens gebeuren; de meest voor de hand liggend probleem is de foute COM-poort. Kies de correcte COM-poort: Werk具gen → Poort.</p> <p>Nog steeds niet succesvol: Dit zou kunnen betekenen, dat het Arduino board stuk is of dat er iets anders aan de hand is. Hier moet een specialist naar kijken en kan niet 1, 2, 3 worden opgelost.</p>	<p><b>Applicatie in de echte wereld:</b></p> <p>Bijna alle moderne flatscreen TV's en monitors gebruiken LED indicatie-lampjes om aan te geven, dat het toestel aan of uit is.</p> 

## Circuit #2

2

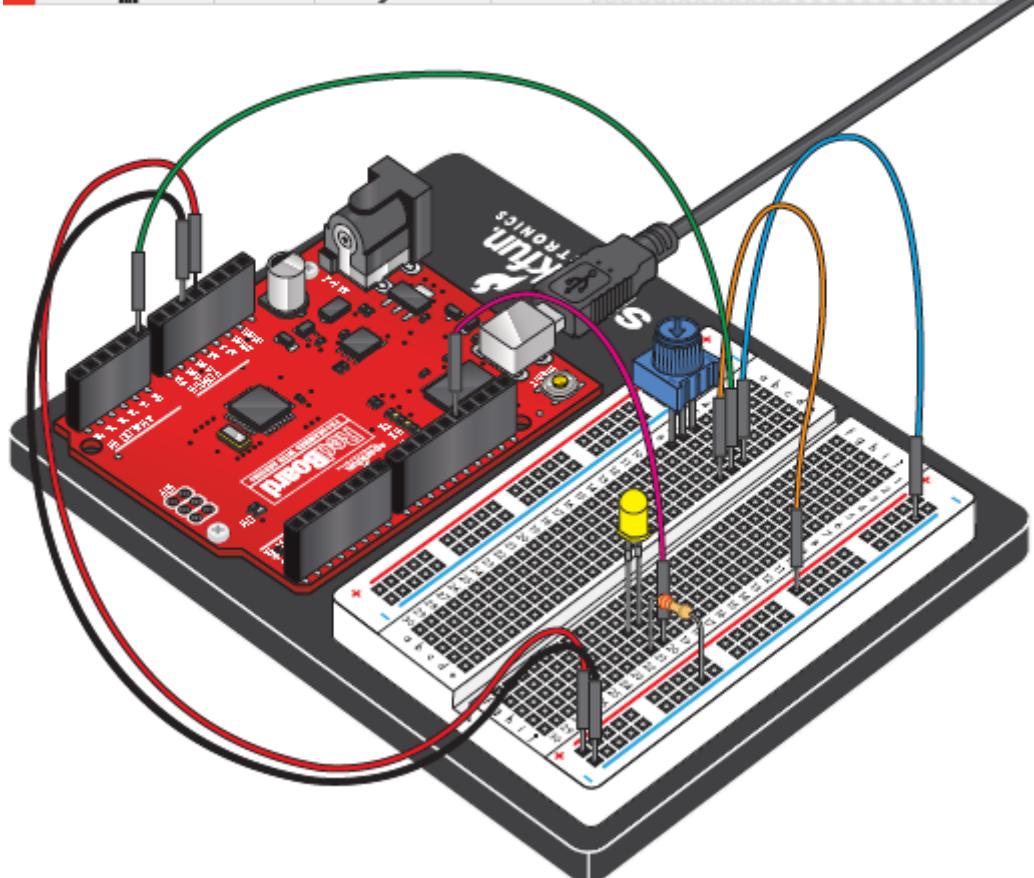


### Potentiometer

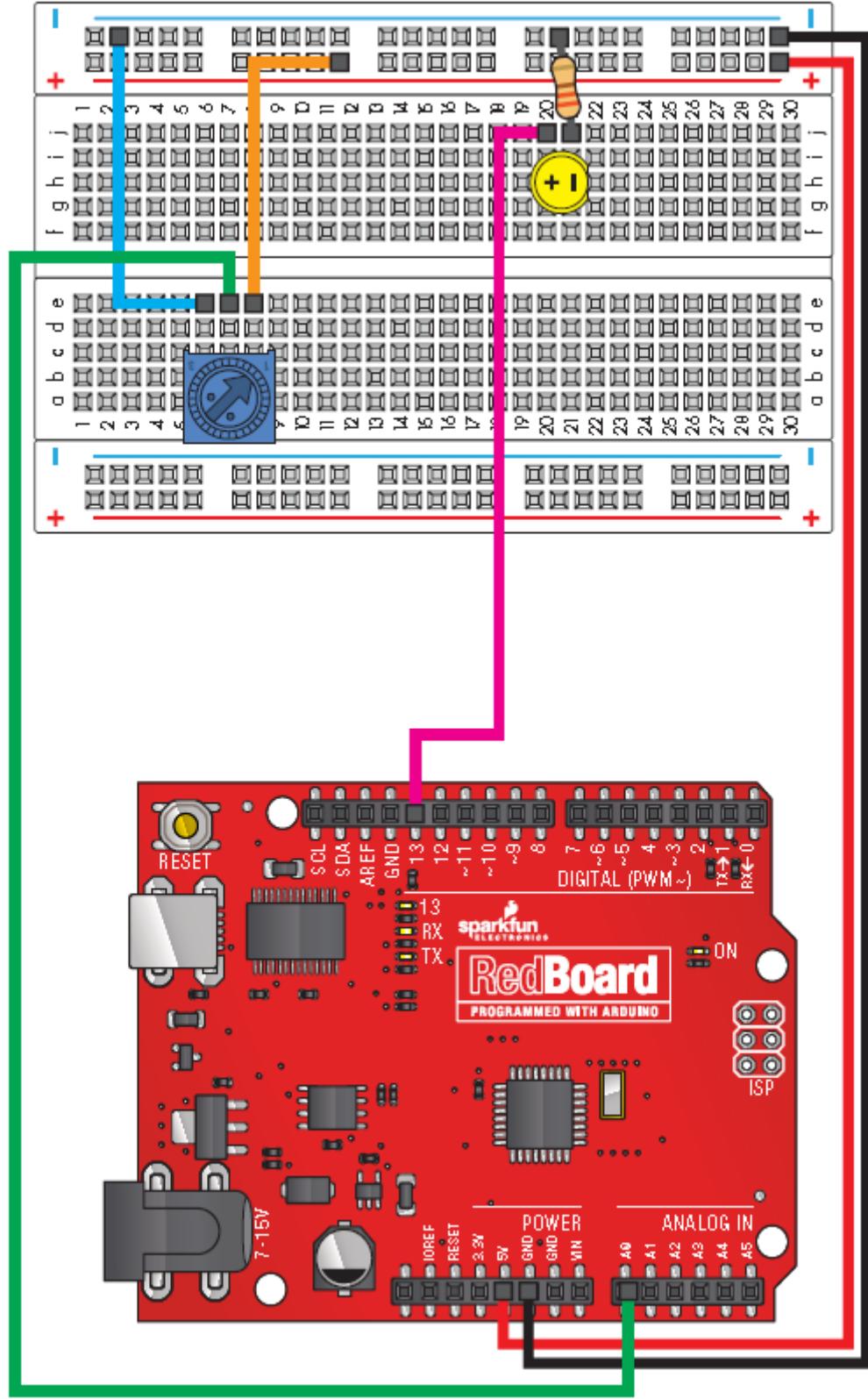
In dit circuit ga je werken met een potentiometer. De potentiometer wordt ook wel een variabele weerstand genoemd.

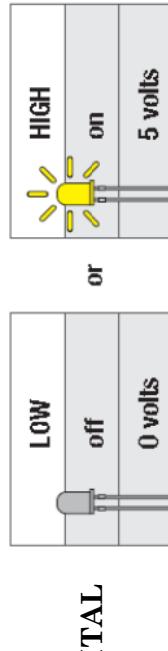
Als de potentiometer met de 2 buitenste aansluitingen aan 5 volt wordt verbonden, dan zal de middelste aansluiting een output geven tussen 0 en 5 volt afhankelijk van de stand van de draaiknop. De potentiometer is een prima voorbeeld van een variabele voltage verdeler. De voltage wordt proportioneel verdeeld tussen de middelste aansluiting en grond (0 volt). Met dit circuit leer je hoe je een potentiometer kunt gebruiken om de helderheid van een LED te regelen.

PARTS:	Potentiometer	LED	330Ω Resistor	Wire
	x1	x1	x1	x6



Circuit 2: Potentiometer



Digitaal versus Analoog:	<p>Als je goed naar het Arduino board kijkt, zie je pinnen die 'DIGITAL' of 'ANALOG' zijn gelabeld. Wat is het verschil?</p> <p>Vele van de componenten waar je mee werkt, zoals LEDs en drukknoppen, hebben maar 2 waarden: aan of uit, oftewel 'HIGH' (5 volts) en 'LOW' (0 volts). Deze pinnen van de Arduino zijn fantastisch om signalen te sturen naar en te ontvangen van de buitenwereld. Je kunt er trouwens mee uithalen zoals dimmen (zeer snel knipperen), of seriële communicatie.</p>	 <table border="1" data-bbox="673 480 841 835"> <tr> <td data-bbox="673 480 765 597"><b>DIGITAL</b></td><td data-bbox="765 480 841 597"> <b>HIGH</b>    <b>on</b>  <b>5 volts</b> </td></tr> <tr> <td data-bbox="673 597 765 835"></td><td data-bbox="765 597 841 835"> <b>or</b>    <b>LOW</b>    <b>off</b>  <b>0 volts</b> </td></tr> </table>	<b>DIGITAL</b>	<b>HIGH</b>  <b>on</b> <b>5 volts</b>		<b>or</b>  <b>LOW</b>  <b>off</b> <b>0 volts</b>	<p>Maar er zijn ook zaken, die niet 'aan' of 'uit' gaan. Temperatuur, controle knoppen, geluid, etc. hebben een continue range van waarden tussen HIGH en LOW. Voor deze situaties zijn er op het Arduino board <b>zes</b> analoge inputs, die de input voltage vertalen naar een nummer in de range van 0 (0 volts) tot 1023 (5 volts). Analog pinnen kunnen deze waarden meten.</p> <table border="1" data-bbox="1179 480 1298 835"> <tr> <td data-bbox="1179 480 1298 597"><b>ANALOG</b></td><td data-bbox="1298 480 1298 597"></td><td data-bbox="1179 597 1298 835"> <b>0 volts</b>  <b>to</b>  <b>1023</b> </td></tr> </table>	<b>ANALOG</b>		<b>0 volts</b> <b>to</b> <b>1023</b>
<b>DIGITAL</b>	<b>HIGH</b>  <b>on</b> <b>5 volts</b>									
	<b>or</b>  <b>LOW</b>  <b>off</b> <b>0 volts</b>									
<b>ANALOG</b>		<b>0 volts</b> <b>to</b> <b>1023</b>								

2



*Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → Circuit\_02*



Notitie code:

**int sensorValue;**

Een 'variabele' bewaart een waarde met een door jouw gegeven naam. Deze moet eerst worden geïntroduceerd of 'declared', voordat je die kunt gebruiken. Hier wordt een variabele aangemaakt met de naam `sensorValue` van het type 'int' (integer). Variabele zijn hoofdlettergevoelig.

**sensorValue =  
analogRead(sensorPin);**

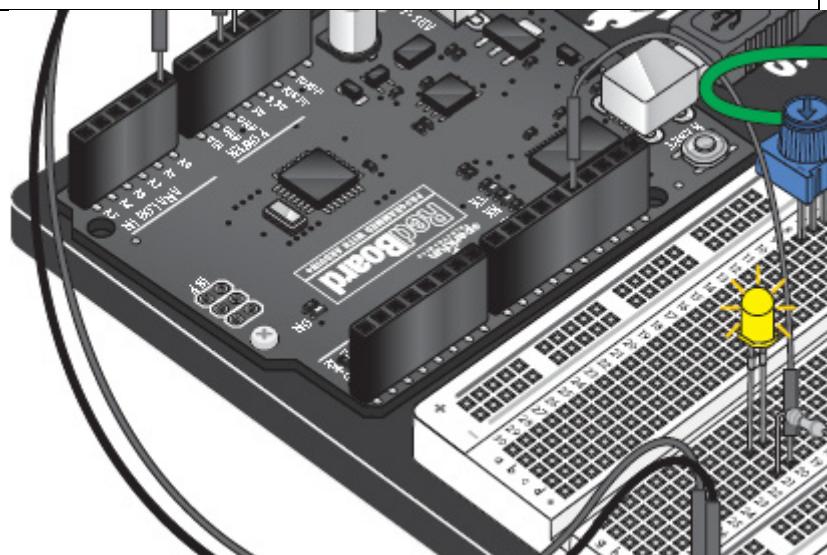
We gebruiken `analogRead()` functie om een waarde te lezen van een analoge pin. `analogRead()` heeft één parameter, nl. de analoge pin ('`sensorPin`'), die je wilt gebruiken, en geeft een nummer ('`sensorValue`') terug tussen 0 en 1023.

**delay(sensorValue);**

De Arduino is heel snel, in staat om duizend regels code elke seconde uit te voeren. Om het te vertragen, zodat we kunnen zien wat er gebeurt, voeren we een vertraging in. `delay()` telt in milliseconden; er gaan 1000 milliseconden in één seconde.

### Wat gebeurt er nu:

Je zou nu de LED sneller of langzamer moeten zien knipperen afhankelijk van de stand van de potentiometer, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven.



### Tips:

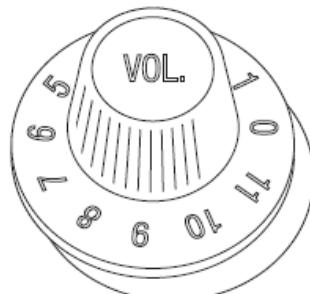
Werkert sporadisch:  
Dit komt waarschijnlijk door een slechte verbinding met de potentiometer-pinnen. Dit kan verholpen worden door de potentiometer iets naar beneden te drukken.

Werkert niet:  
Zorg ervoor, dat de middelste pin van de potentiometer met analoge pin 0 is verbonden en niet de digitale pin 0.

LED licht niet op:  
LEDs werken slechts in één richting. Het is een component met polariteit. Verwijder de LED en draai het 180°. Je hoeft niet bang te zijn, dat de LED stuk gaat.

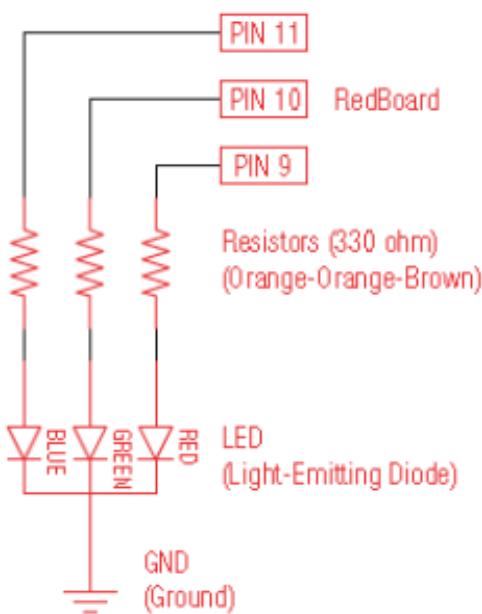
### Applicatie in de echte wereld:

Het meest traditionele waar een potentiometer voor gebruikt wordt, is de volume-knop.



### Circuit #3

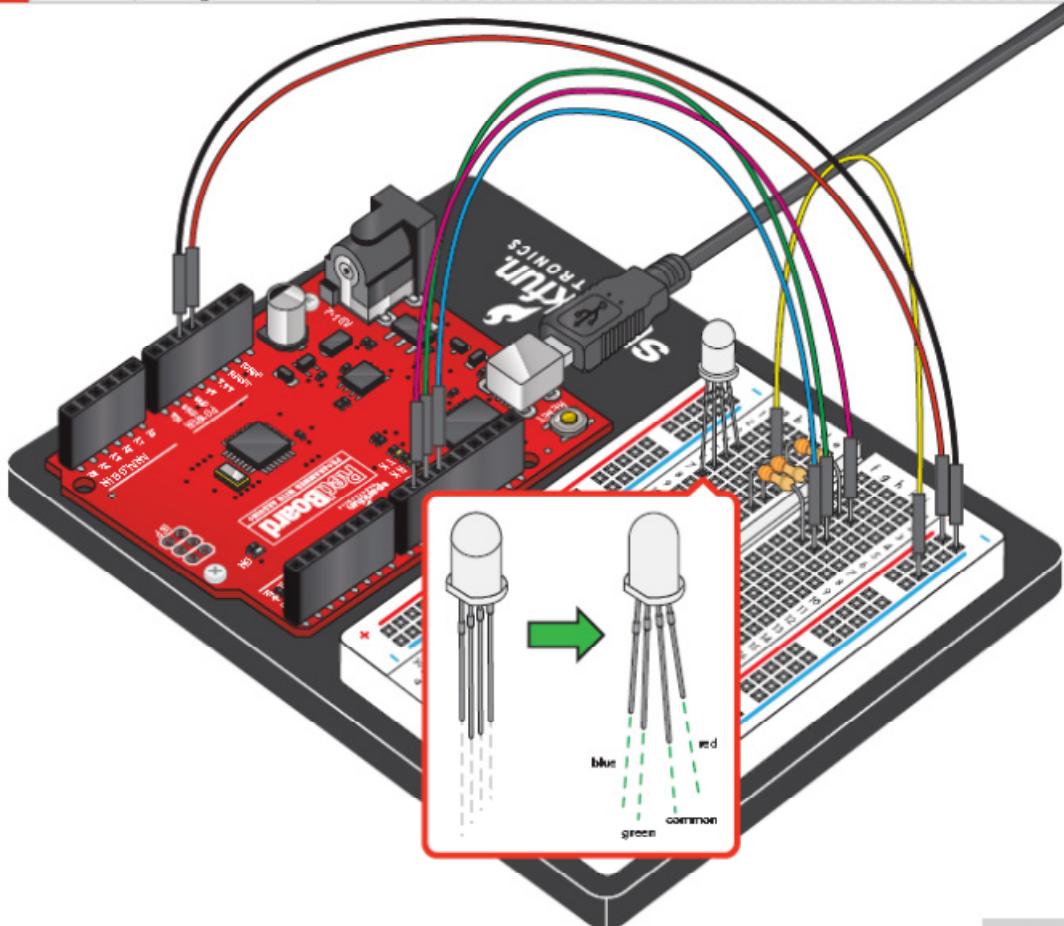
3



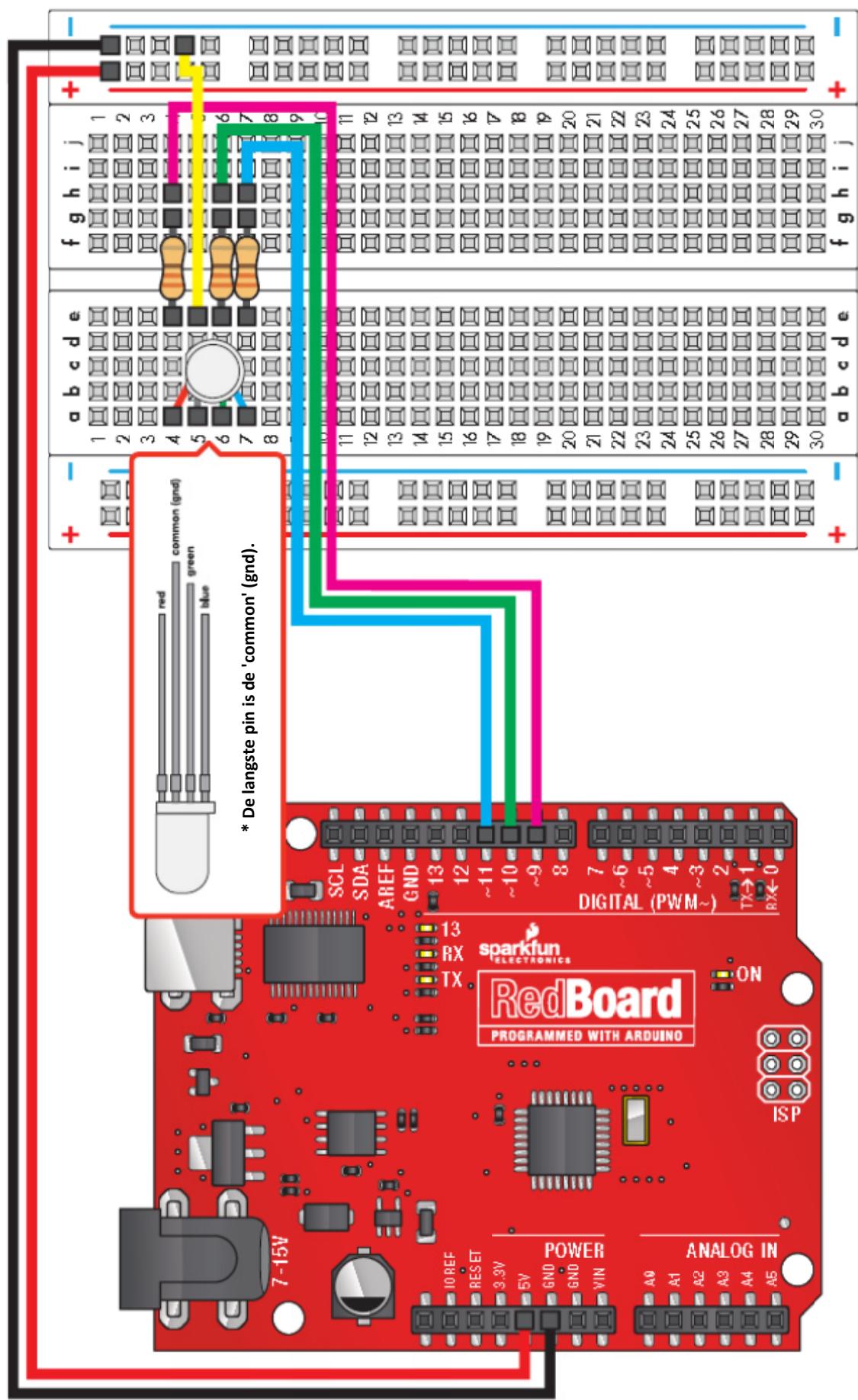
### RGB LED

Weet je wat nog leuker is dan een knipperende LED? Kleuren veranderen met één LED. RGB, of rood-groen-blauw, LEDs hebben drie verschillende kleurdiodes in één en kunnen worden gecombineerd om allerlei kleuren te maken.

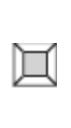
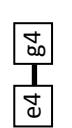
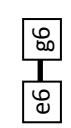
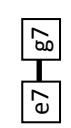
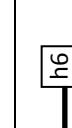
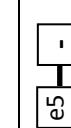
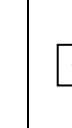
In dit circuit leer je hoe je een RGB LED moet gebruiken om verschillende kleuren te creëren. Bijna elke kleur is mogelijk afhankelijk van de helderheid van elk van de kleurdiodes.



## Circuit 3: RGB LED



De schockerende waarheid achter analogWrite():	
<p>We hebben gezien, dat de Arduino analoge voltage (voltages tussen 0 en 5 volts) kan lezen m.b.v. de <b>analogRead()</b> functie. Is er een manier om Arduino ook analoge output voltages te laten leveren?</p> <p>Het antwoord is nee... en ja. De Arduino heeft geen echte analoge voltage output. Maar, omdat de Arduino zo snel is, kan dit wel worden gesimuleerd met wat men noemt <b>PWM</b> ("Pulse-Width Modulation"). De pinnen op de Arduino met een "~-" ernaast zijn PWM/Analog output kompatibel.</p> <p>Arduino kan een pin bijna 1000x per seconde aan- en uitschakelen. PWM gaat een stap verder door de tijd dat een pin HIGH of LOW is te variëren. Als een LED langere tijd HIGH is, dan schijnt hij helder. Bij een kortere tijd lijkt de LED te dimmen. De snelle verandering kan niet door ons oog worden waargenomen.</p> 	

Component:	Afbeelding:		
RGB LED (5 mm)			
330Ω weerstand		e4	g4
330Ω weerstand		e6	g6
330Ω weerstand		e7	g7
Draadbrug		Pin 11	h7
Draadbrug		Pin 10	h6
Draadbrug		-	e5
Draadbrug		h4	+
Draadbrug		5V	GND
Draadbrug			

# 3



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → **Circuit\_03**



Notitie code:

```
for (x=0; x<768; x++)  
{ }
```

Een for( ) loop wordt gebruikt om stapsgewijs door nummers te gaan waarbij herhaaldelijk de code tussen de accolade's {} wordt uitgevoerd. Hier wordt variabele 'x' gebruikt, beginnend bij 0 en eindigt bij 767 en wordt bij elke run met 1 verhoogd ('x++').

```
for (x<=255)  
{ }  
else  
{ }
```

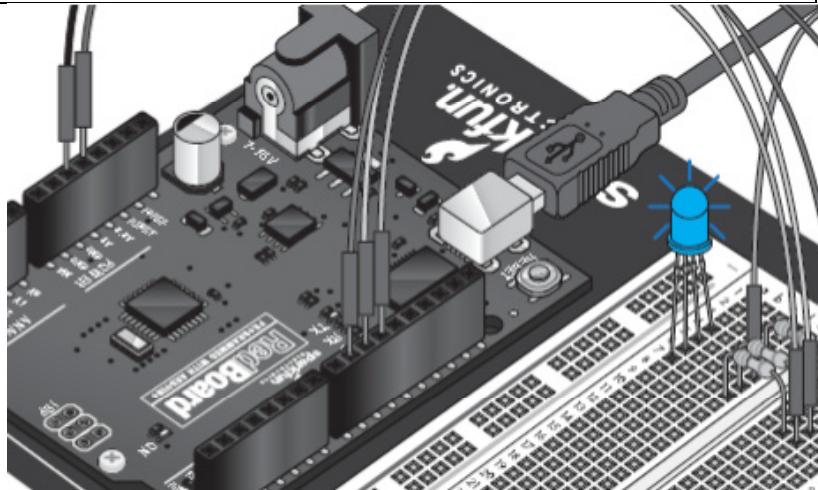
"If/else" wordt gebruikt om een beslissing te nemen. De voorwaarde tussen de haakjes () wordt geëvalueerd; indien die waar is, wordt de code tussen de eerste set accolade's {} uitgevoerd. Als het niet waar is, wordt de code tussen de tweede set accolade's {} uitgevoerd.

```
delay(sensorValue);
```

De Arduino is heel snel, in staat om duizend regels code elke seconde uit te voeren. Om het te vertragen, zodat we kunnen zien wat er gebeurt, voeren we een vertraging in. delay() telt in milliseconden; er gaan 1000 milliseconden in één seconde.

## Wat gebeurt er nu:

Je zou nu de LED moeten zien oplichten, maar nu in nieuwe, gekke kleuren, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven.



## Tips:

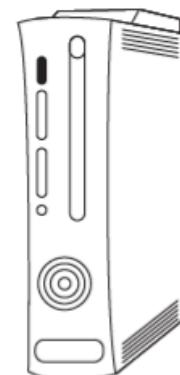
LED blijft uit of geeft de verkeerde kleur:  
Omdat de pinnen zo dicht bij elkaar zitten, kan het zijn dat ze verkeerd in de breadboard zijn geplaatst. Check dit nauwkeurig.

Alleen rood zichtbaar:  
De rode diode in de RGB LED is veel helderder dan de andere twee. Om een meer gebalanceerde kleurschakering te krijgen, kun je een hogere weerstand voor rood gebruiken of de code aanpassen:

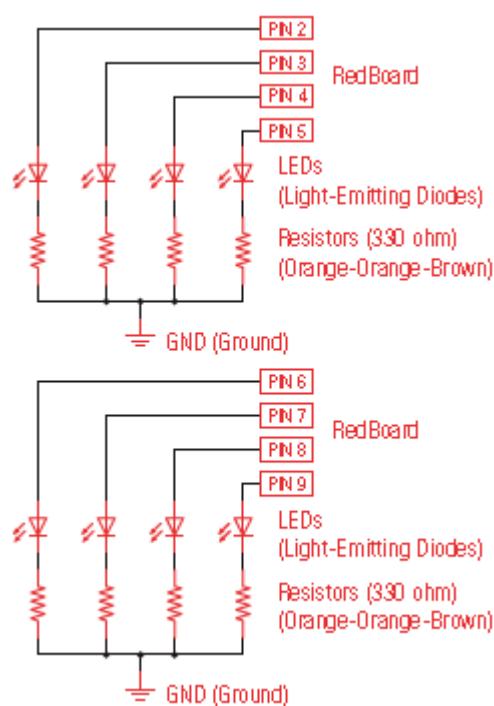
```
analogWrite(RED_PIN, redIntensity);  
naar  
analogWrite(RED_PIN, redIntensity/3);
```

## Applicatie in de echte wereld:

Veel electronica zoals videogame consoles gebruiken RGB LED om verschillende kleuren te tonen met één lampje. Vaak geeft de kleur van de LED aan wat de conditie is van het apparaat of welke applicatie wordt uitgevoerd.



## Circuit #4



# 4

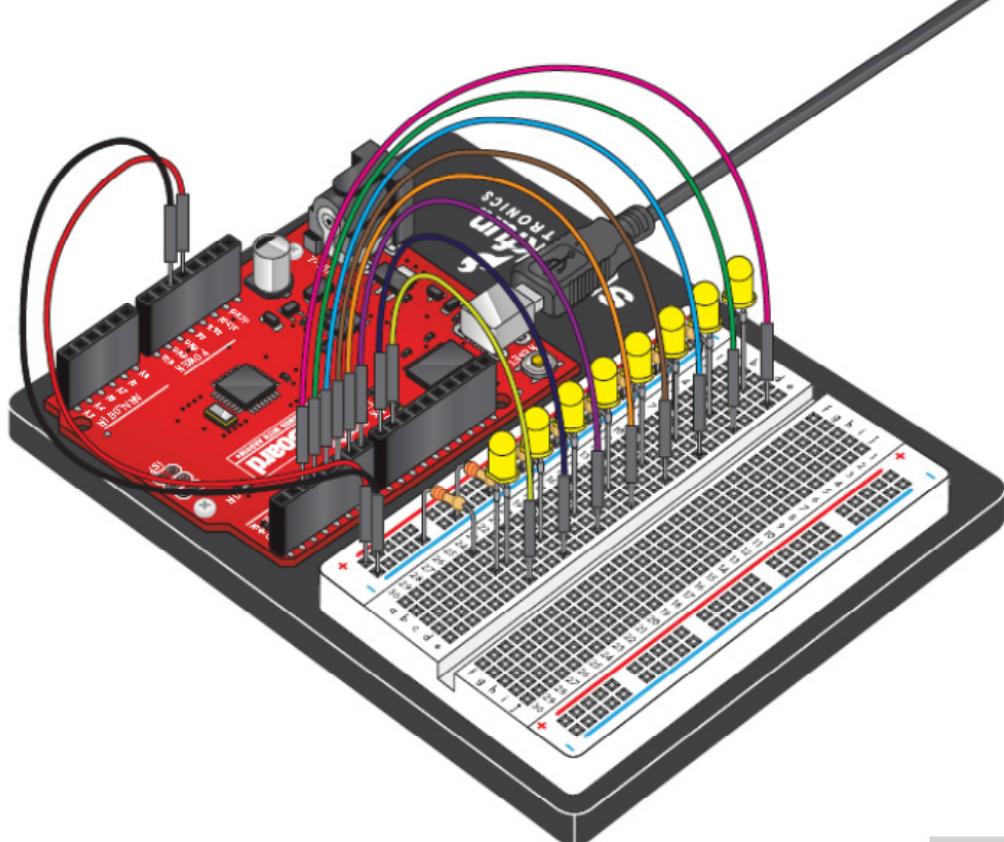
## Meerdere LEDs

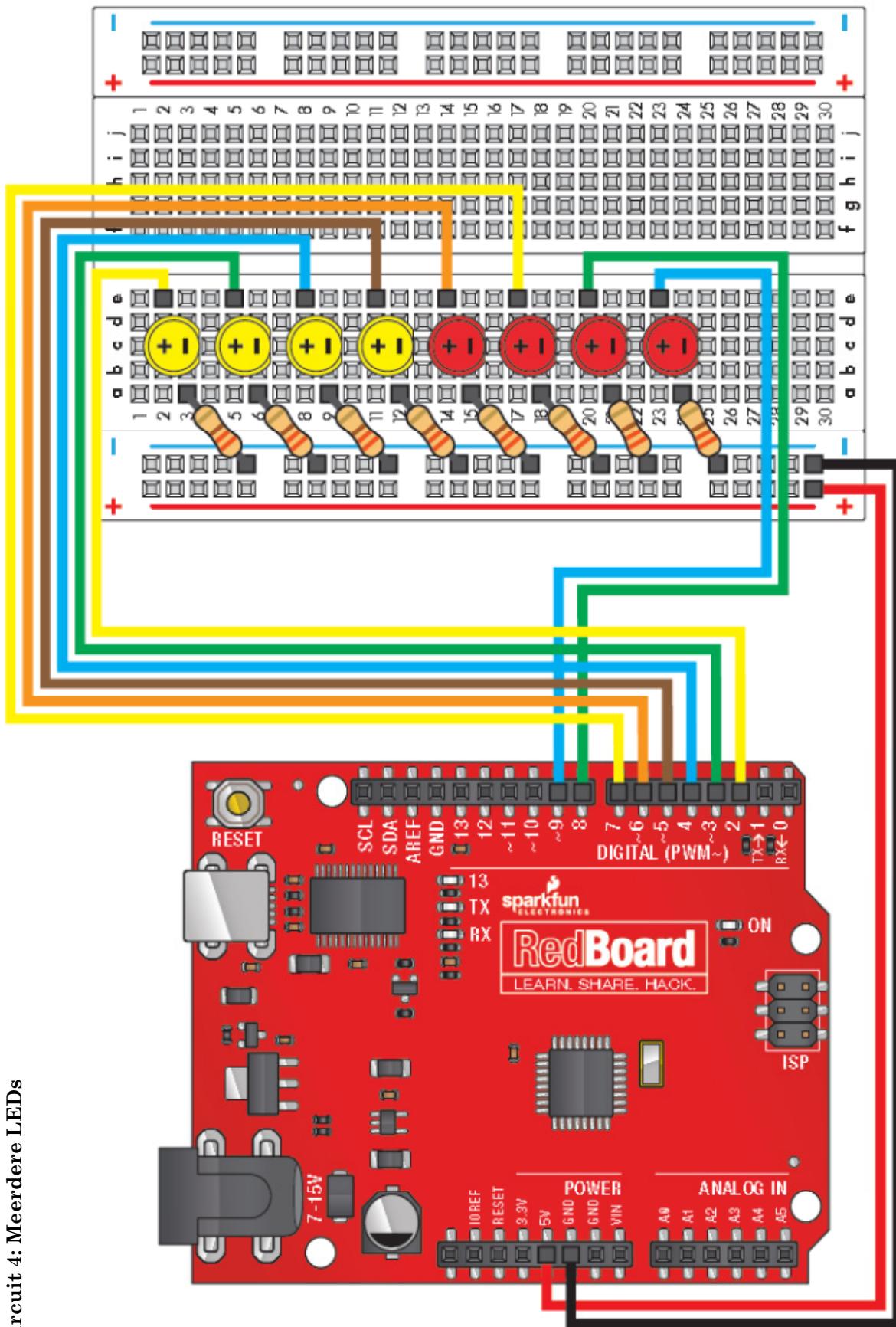
Zo, het is dus gelukt om één LED te laten knipperen, fantastisch! Nu is het tijd om het wat groter aan te pakken - een connectie maken met ACHT LEDs tegelijkertijd. We gaan ook de Arduino testen door verschillende licht sequenties uit te voeren. Dit circuit is een goed begin om programma's te leren schrijven en inzicht te krijgen hoe het Arduino board werkt.

Naast het controleren van 8 LEDs leer je ook enkele programmeertips, zodat je code netjes en overzichtelijk blijft.

**for() loops** - wordt gebruikt wanneer je bepaalde code enkele keren wilt laten uitvoeren

**arrays[]** - wordt gebruikt om variabelen makkelijker te onderhouden door ze te groeperen





Component:	Afbeelding:	Component:	Afbeelding:
LED (5 mm)		-	
LED (5 mm)		c2 - c3	330Ω weerstand
LED (5 mm)		c5 - c6	330Ω weerstand
LED (5 mm)		c8 - c9	330Ω weerstand
LED (5 mm)		c11 - c12	Draadbrug
LED (5 mm)		c14 - c15	Draadbrug
LED (5 mm)		c17 - c18	Draadbrug
LED (5 mm)		c20 - c21	Draadbrug
LED (5 mm)		c23 - c24	Draadbrug
330Ω weerstand		a3 -	Draadbrug
330Ω weerstand		a6 -	Draadbrug
330Ω weerstand		a9 -	Draadbrug
330Ω weerstand		a12 -	Draadbrug
330Ω weerstand		a15 -	Draadbrug
		e18 -	
		e21 -	
		e24 -	
		e2	
		e5	
		Pin 2	
		Pin 3	
		Pin 4	
		Pin 5	
		Pin 6	
		Pin 7	
		Pin 8	
		Pin 9	
		5V	
		GND	

# 4



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → **Circuit\_04**



Notitie code:

**int ledPins[]={2,3,4,5,6,7,8,9};**

Als je veel variabelen moet controleren, dan is een "array" een gemakkelijke manier om die te groeperen. Hier maken we een array van integers aan, genaamd ledPins met 8 elementen.

**digitalWrite(ledPins[0], HIGH);**

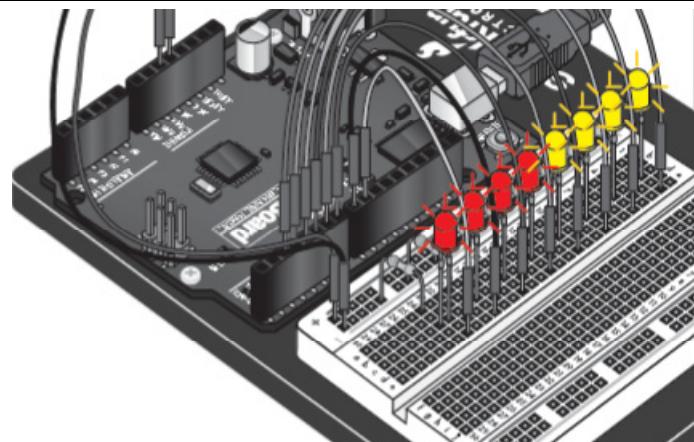
Je refereert naar een element in een array door de positie in de array aan te geven. Het eerste element bevindt zich op positie 0, de tweede op positie 1, etc. De referentie wordt uitgevoerd met "ledPins[x]" waarbij x de positie aangeeft. Hier maken we digital pin 2 HIGH, omdat het element op positie 0 een "2" is.

**index=random(8);**

Computers zijn handig om een handeling meerdere keren te kunnen uitvoeren. Maar soms wil je handelingen willekeurig uitvoeren. Een willekeurige handeling is bv. het rollen van een dobbelsteen simuleren. De random() functie kan dit doen. Lees er meer over op <http://arduino.cc/en/reference/random>.

## Wat gebeurt er nu:

Dit circuit lijkt op circuit #1, maar in plaats van één LED, zie je nu alle LEDs knipperen, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven.



## Tips:

### Sommige LEDs knipperen niet

Het is makkelijk de LED verkeerd in het breadboard te plaatsen. Check dit nauwkeurig.

### Sequentie loopt verkeerd

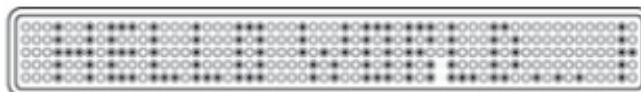
Door de vele draden kan het zijn, dat de volgorde verkeerd is. De eerste LED zit op pin 2 en elke volgende LED op de openeenvolgende pins.

### Opnieuw starten

Soms is het moeilijk om te controleren welke draad verkeerd zit. Haal alle draden weg en begin gewoon opnieuw. Dat is vaak eenvoudiger dan proberen het probleem te achterhalen.

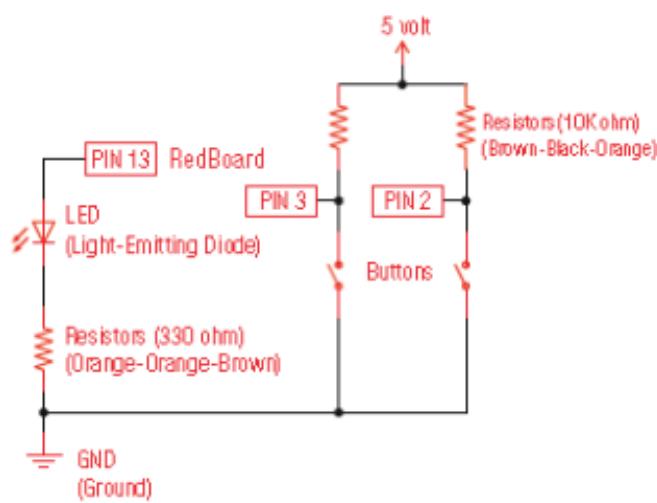
## Applicatie in de echte wereld:

Een zogenaamde scrolling bericht toont korte fragmenten van belangrijke informatie. Deze zijn opgebouwd uit veel LEDs.



## Circuit #5

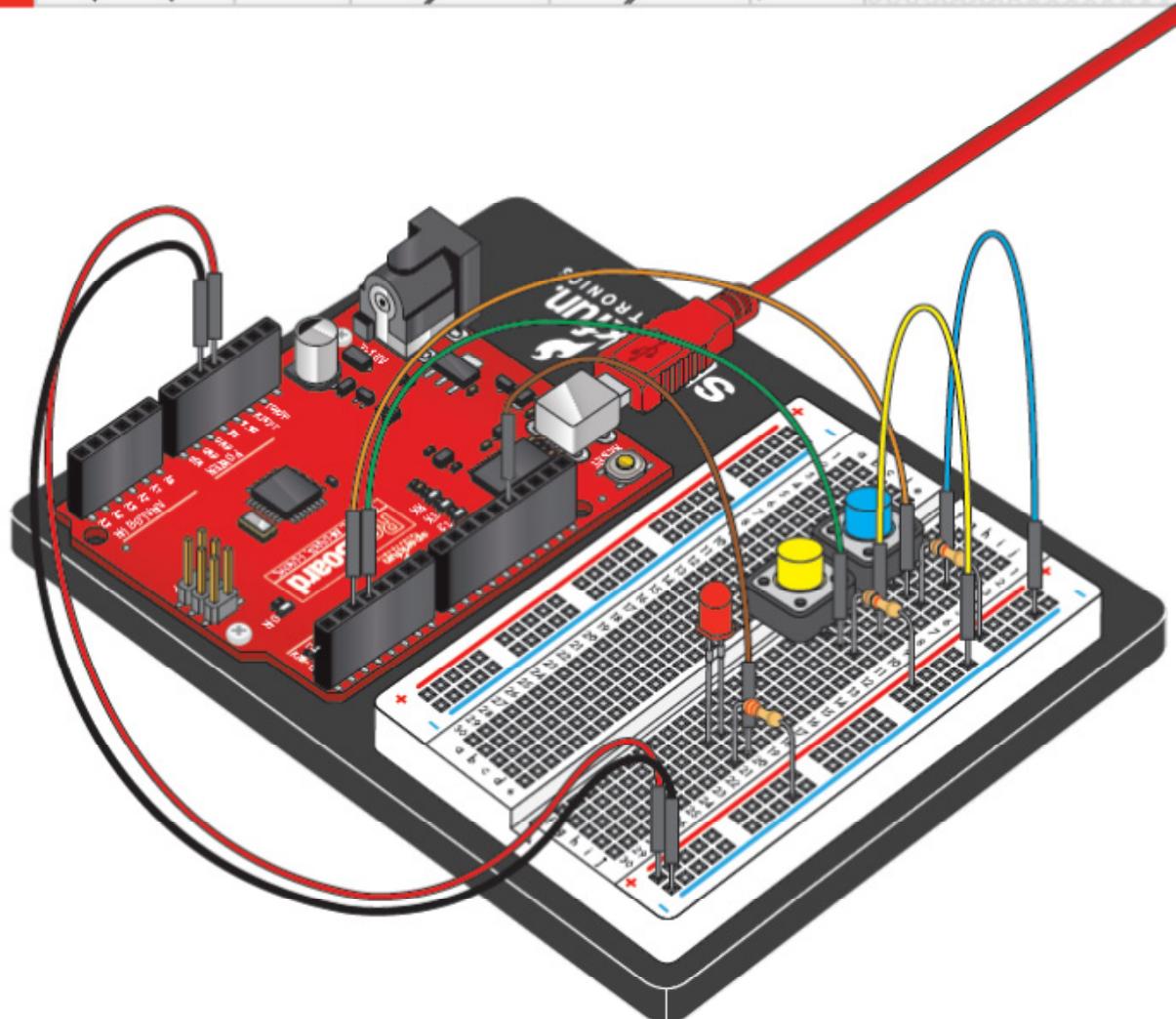
5

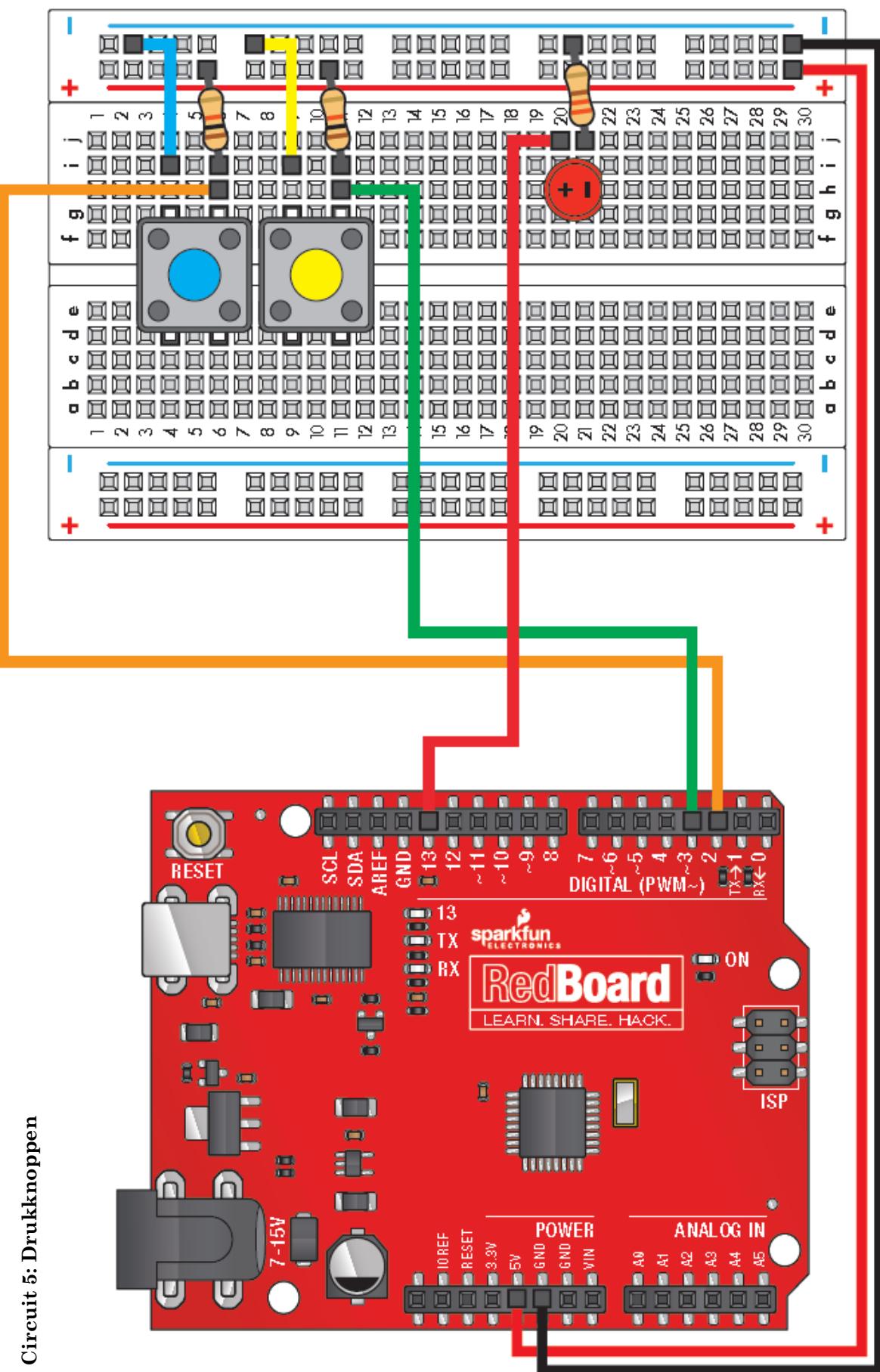


### Drukknoppen

Tot nu toe hebben we ons alleen bezig gehouden met outputs. Nu bekijken we het andere einde van het spectrum en gaan we spelen met inputs. In dit circuit gaan we kijken naar de meest algemene en simpele input, een drukknop. De manier waarop een drukknop werkt met de Arduino is, dat het voltage LOW gaat als de knop wordt ingedrukt. De Arduino neemt dit waar en reageert erop.

In dit circuit maken we ook gebruik van een pull-up weerstand, die het voltage HIGH houdt als de knop niet wordt ingedrukt.





Hoe gebruik je logische parameters:		
Eén van de zaken, die het Arduino board zo bruikbaar maken, is het vermogen om complexe beslissingen te nemen op basis van de inputs, die het Arduino board krijgt. Bijvoorbeeld, je kunt een thermoslaat maken, die de verwarming inschakelt als het te koud is, een ventilator als het te warm is, de planten water geeft als deze droog staan.		
Om zulke beslissingen te kunnen maken, biedt de Arduino omgeving een set van logische operatoren, waarmee je complexe "If" beslissingen kunt bouwen. De volgende logische operatoren zijn aanwezig:		
<b>==</b>	<b>EQUivalence</b>	<b>A == B</b> is waar als A en B <b>gelijk</b> zijn
<b>!=</b>	<b>DIFFERENCE</b>	<b>A != B</b> is waar als A en B <b>niet gelijk</b> zijn
<b>&amp;&amp;</b>	<b>AND</b>	<b>A &amp;&amp; B</b> is waar als zowel A als B <b>WAAR</b> zijn
<b>  </b>	<b>OR</b>	<b>A    B</b> is waar als A of B <b>beide WAAR</b> zijn
<b>!</b>	<b>NOT</b>	<b>!A</b> is <b>WAAR</b> als A <b>NIET WAAR</b> is <b>!A</b> is <b>NIET WAAR</b> als A <b>WAAR</b> is
Je kunt deze logica combineren om complexe if() functie te bouwen.		
<i>Bijvoorbeeld</i>		
<pre>if(mode==heat) &amp;&amp; ((temperature&lt;threshold)    (override==true)) {   digitalWrite(HEATER, HIGH); }</pre>		
...schakelt de verwarming aan als de schakeling in verwarming modus is <b>AND</b> de temperatuur laag is, <b>OR</b> handmatige schakeling wordt bediend. Met gebruik van deze logische operatoren kan de Arduino intelligente beslissingen maken en kan de buitenwereld worden gecontroleerd.		

# 5



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → **Circuit\_05**



Notitie code:

**pinMode button2Pin, INPUT);**

De digitale pinnen kunnen zowel als inputs als outputs worden gebruikt. Voordat je een pin gaat gebruiken, zal het Arduino board moeten worden verteld hoe de pinnen worden gebruikt.

**button1State =  
digitalRead(button1Pin);**

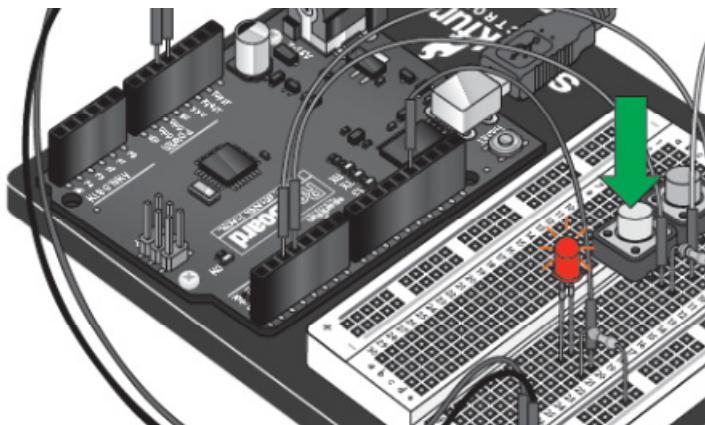
Om een digitale input te lezen, gebruik je de digitalRead() functie. Het geeft een HIGH terug als er 5V op de pin staat, of LOW als er 0V op staat.

**if (button1State == LOW)**

Omdat de drukknoppen verbonden zijn met GND, zal het LOW schakelen als deze wordt ingedrukt. Hier gebruiken we de "equivalence" operator ("==") om te zien of de knop is ingedrukt.

## Wat gebeurt er nu:

Je zult de LED zien oplichten als één van de knoppen wordt ingedrukt, en gaan uit als beide knoppen tegelijkertijd worden ingedrukt, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven.



## Tips:

### LED gaat niet aan

De drukknoppen zijn vierkant en het kan gebeuren dat ze verkeerd zijn geplaatst. Draai de knoppen 90° en kijk of ze dan wel werken.

### Veel te simpel

Natuurlijk is dit een primitieve schakeling, maar als je eenmaal allerlei componenten met elkaar verbindt, dan is er veel mogelijk.

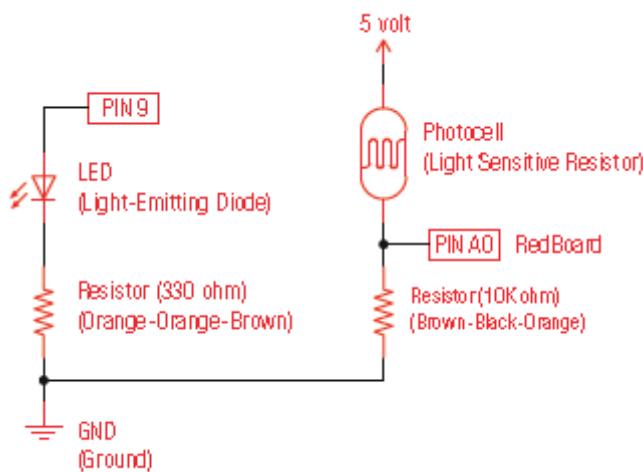
## Applicatie in de echte wereld:

Knoppen zie je overal verschijnen waar iets moet worden in- of uitgeschakeld of om een keuze te maken.



## Circuit #6

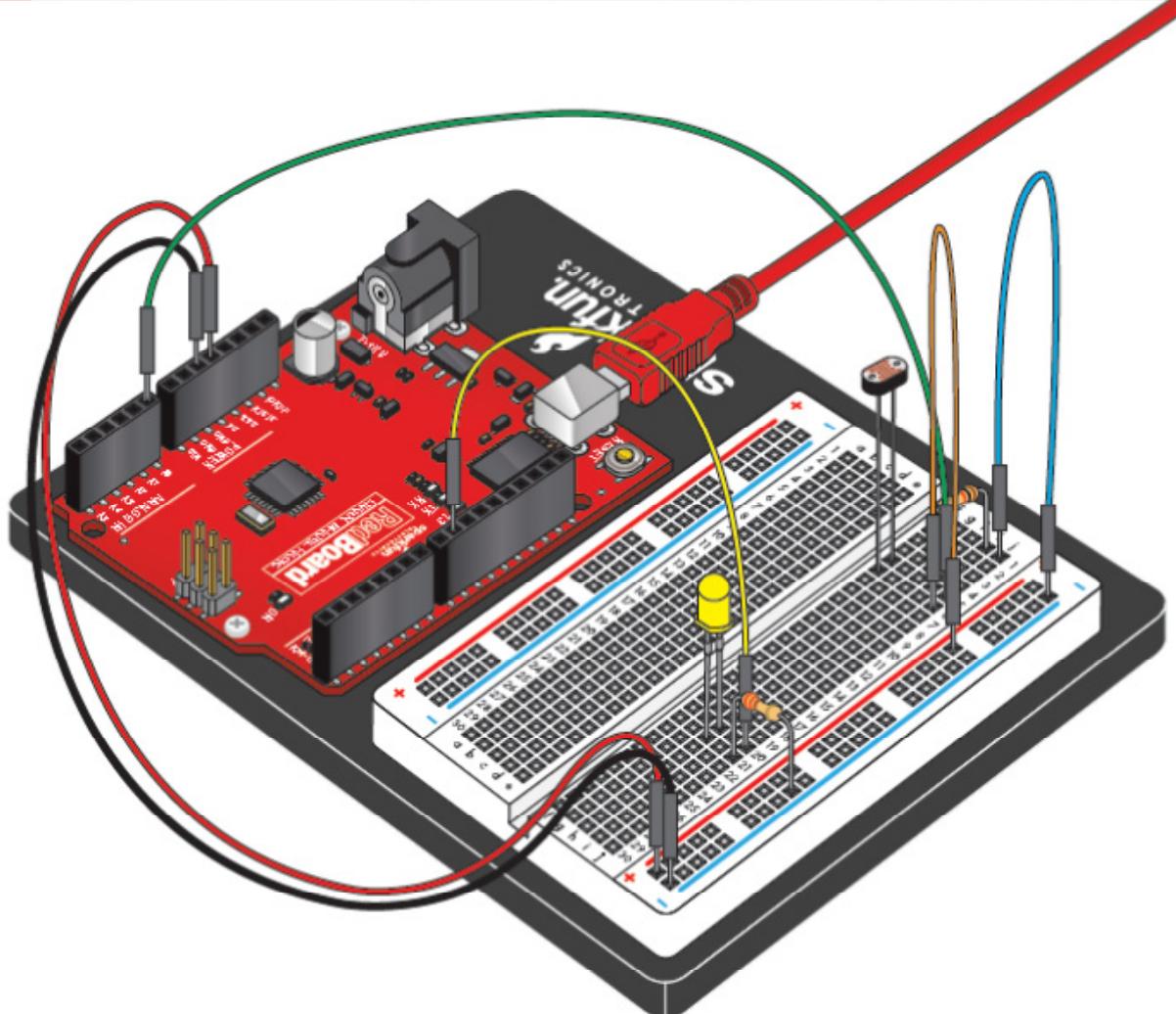
# 6



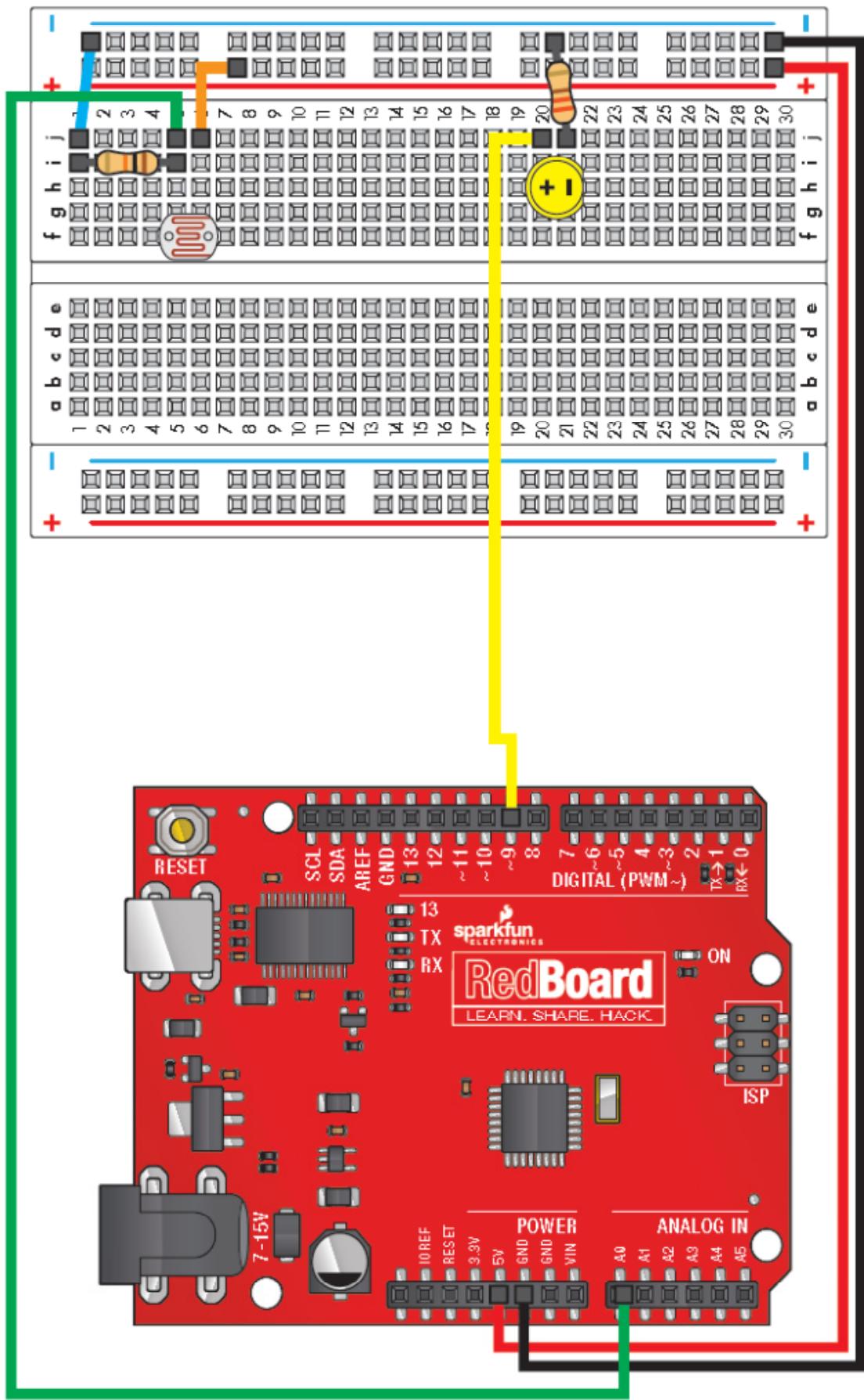
## Fotoweerstand

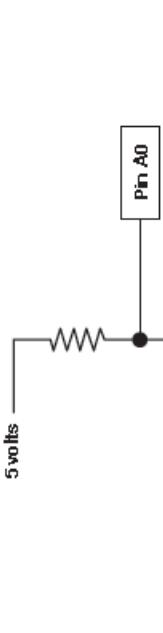
Tot nu toe hebben we ons alleen bezig gehouden met outputs. Nu bekijken we het andere einde van het spectrum en gaan we spelen met inputs. In dit circuit gaan we kijken naar de meest algemene en simpele input, een drukknop. De manier waarop een drukknop werkt met de Arduino is, dat het voltage LOW gaat als de knop wordt ingedrukt. De Arduino neemt dit waar en reageert erop.

In dit circuit maken we ook gebruik van een pull-up weerstand, die het voltage HIGH houdt als de knop niet wordt ingedrukt.



Circuit 6: Fotoweerstand



<h2>Meten van weerstand sensoren:</h2> <p>Veel van de sensoren, die je gebruikt (potentiometers, fotoweerstand, etc.) lijken op een soort weerstand. Die weerstand verandert in verhouding tot wat zij waarnemen (lichtniveau, temperatuur, geluid, etc.).</p>	<p>De Arduino analoge input</p> 	<p>Een voltage verdeler bestaat uit twee weerstanden. De "top" weerstand is de sensor. De "bottom" is een normale, vaste weerstand. Wanneer de bovenste weerstand aan 5V ligt en de onderste aan grond zal het middelste gedeelte een voltage geven dat proportioneel is aan de onderste weerstand relatief aan de totale weerstand (bovenste + onderste weerstand). Als één van de weerstanden verandert (wat gebeurt als de sensor iets waarnemt), zal ook de output voltage veranderen!</p>	<p>Hoewel de sensor-weerstand zal veranderen, de weerstand-sensoren in dit pakket hebben een waarde van ong. 10K ohms. We willen meestal een vaste weerstand, die dicht bij deze waarde ligt en het is dan ook voor de hand liggend om voor de onderste weerstand een 10kΩ te gebruiken. Uiteraard is het geen verplichting, dat de vaste weerstand de onderste weerstand moet zijn. We doen dit, zodat de foto-weerstand (foto-diode) bij meer licht een hogere voltage genereert. Wanneer de beide weerstand worden omgedraaid, krijgen we de tegenovergestelde situatie, meer licht = minder voltage.</p>
--	---	--	--

Component:	Afbeelding:
Fotoweerstand	
LED (5 mm)	
330Ω weerstand	
10kΩ weerstand	
Draadbrug	
Draadbrug	
Draadbrug	
Draadbrug	
Draadbrug	
Draadbrug	

# 6



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → **Circuit\_06**



Notitie code:

**lightLevel=map(lightLevel,0,1023,0,255);**

**Parameters**

map(value,fromLow,fromHigh,toLow,toHigh)

**value:** waarde, die gedetecteerd wordt

**fromLow:** ondergrens waarde voltage sensor

**fromHigh:** bovengrens waarde voltage sensor

**toLow:** ondergrens waarde van doelement

**toHigh:** bovengrens waarde van doelement

Als we met analogRead() een analoog signaal uitlezen, krijgen we een waarde tussen 0 en 1023. Maar we willen een PWM pin besturen met analogWrite(), dat een nummer tussen 0 en 255 verwacht. De grote waarde kunnen we omzetten in de kleine waarde met de map() functie. Zie ook <http://arduino.cc/en/reference/map> voor meer info.

**lightLevel=constraint(lightLevel,0,255);**

**Parameters**

constraint(x, a, b)

**x:** nummer, dat beperkt moet worden

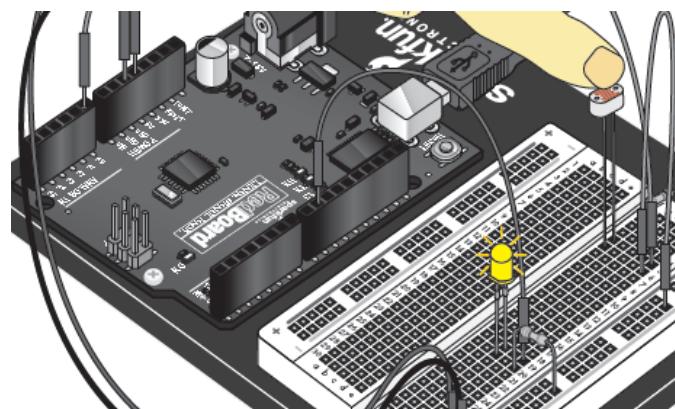
**a:** ondergrens van de waarde

**b:** bovengrens van de waarde

Omdat map() toch een waarde buiten de range kan genereren, bouwen we een beperking (constraint) in, zodat de waarde binnen de range blijft. Wanneer de waarde buiten de range valt, zal constraint() de waarde veranderen in het hoogste of laagste nummer van de range. Binnen de range verandert de waarde niet.

## Wat gebeurt er nu:

De LED zal van lichtsterkte veranderen afhankelijk van hoeveel licht er op de fotodiode valt, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven.



## Tips:

**LED gaat niet aan**

We maken de fout keer op keer. Een LED heeft een polariteit en werkt niet in beide richtingen. Draai de LED en kijk of het werkt.

**Geen verandering bij lichtdosering**

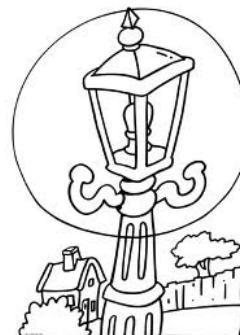
Zorg ervoor dat de fotodiode goed is aangesloten. Soms wordt een draadbrug in het verkeerde gaatje gestoken en is er geen contact met de fotodiode.

**Het werkt nog niet zoals het hoort**

Wellicht is de ruimte waarin de opstelling staat te licht of te donker. Verander het omgevingslicht; lamp dimmen/uit of een zaklamp gebruiken.

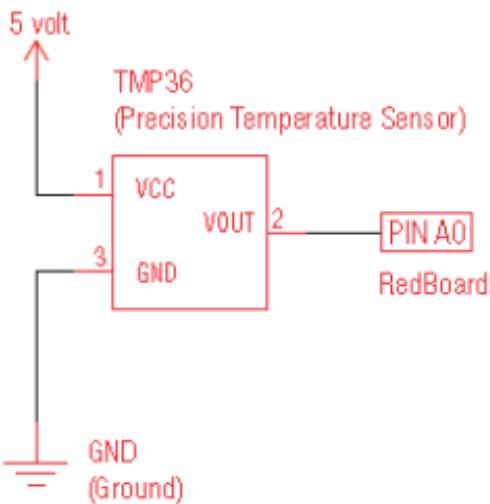
## Applicatie in de echte wereld:

Een straatlantaarn gebruikt een lichtsensor om te detecteren wanneer het licht aan moet in de avond, en weer uit in de ochtend.



## Circuit #7

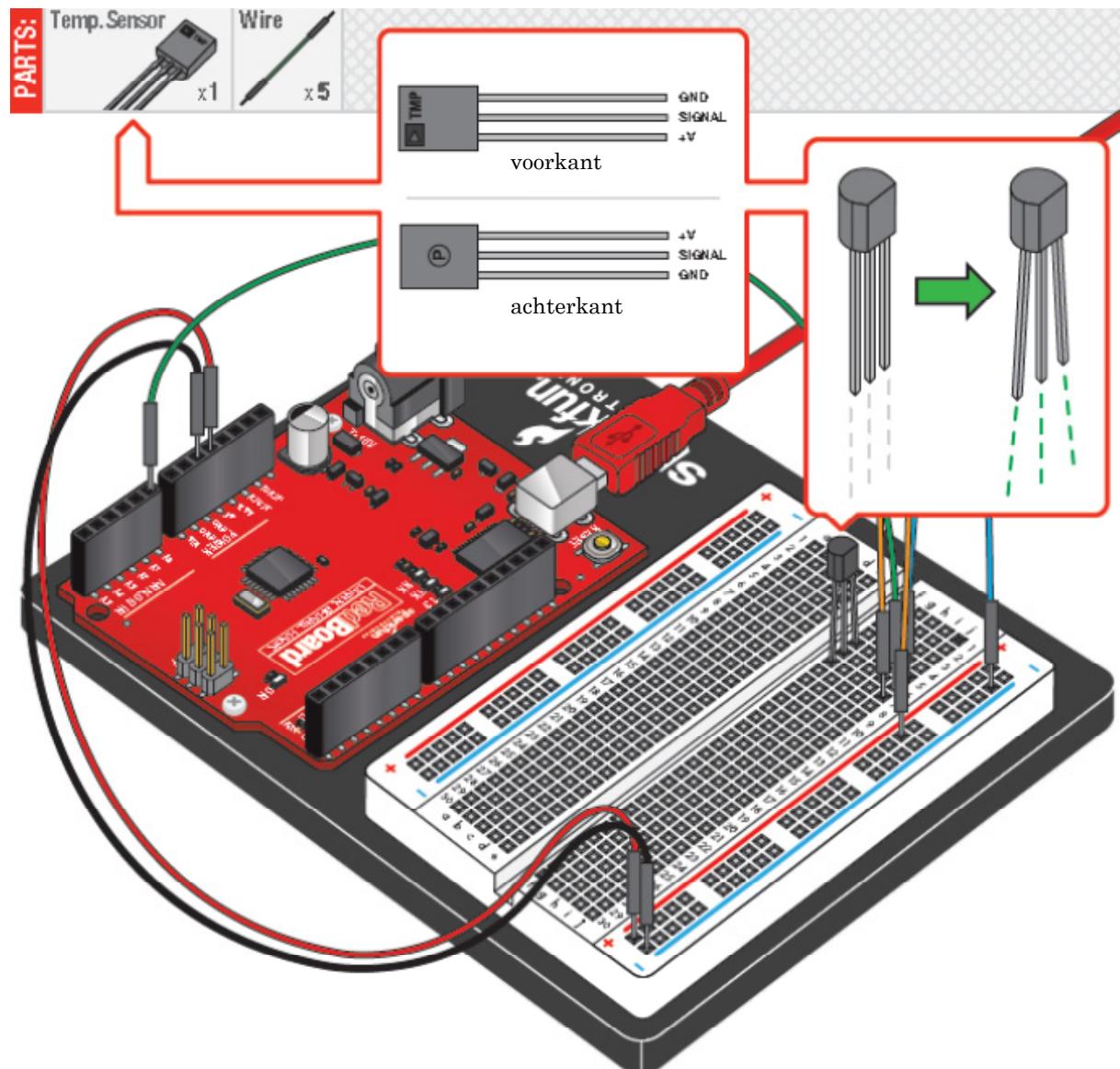
7



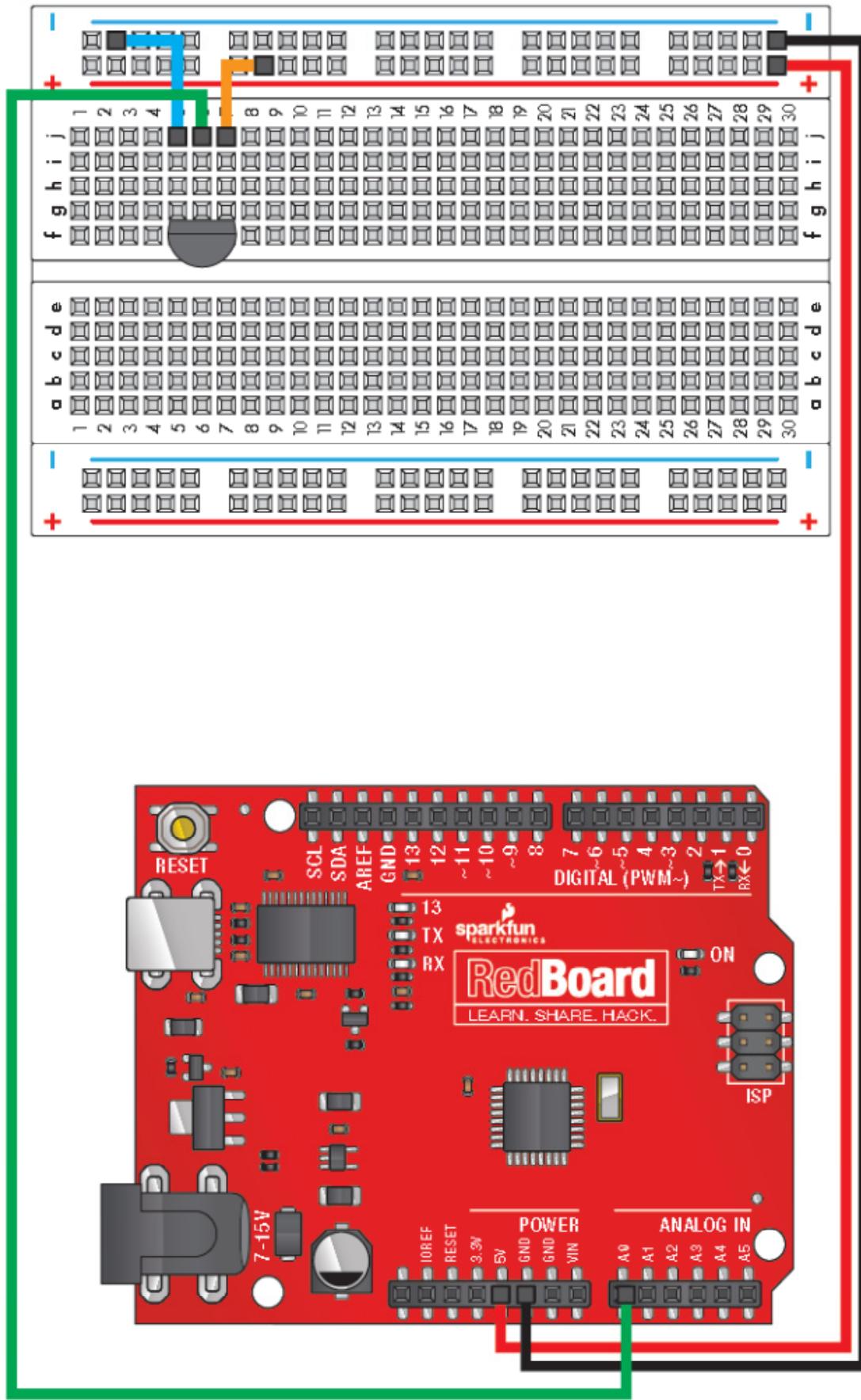
### Temperatuursensor

Een temperatuursensor is precies wat het woord zegt - een sensor om de omgevingstemperatuur te meten. De sensor heeft 3 pinnen, een positieve pin, een grond- en signaalpin. Dit is een lineaire temperatuur sensor. Een verandering in temperatuur van één graad geeft een verandering van 10 millivolts aan de sensor's signaalpin. De TMP36 heeft een nominale waarde van 750 mV bij 25°C. In dit circuit leer je de sensor te gebruiken met de Arduino, waarbij de waarden worden weergegeven in de Arduino's seriële monitor.

! Wanneer je het circuit bouwt, zorg er dan voor dat je de sensor niet verwart met de transistor in het pakket; ze lijken heel veel op elkaar. Kijk naar de tekst "TMP" op de sensor.



Circuit 7: Temperatuur sensor



**De schokkerende waarheid achter analogWrite():**

Dit circuit maakt gebruik van de Arduino IDE's seriële monitor. Om dit te openen wordt eerst de sketch ge-upload en klik dan op het vergrootglas aan de rechterbovenkant van de IDE. De seriële monitor werkt alleen goed als de communicatie-snelheid van het programma hetzelfde is als de baud-rate van de seriële monitor. Het programma werkt met een baud-rate van 9600. Als de seriële monitor daar niet op ingesteld is, verander dat dan.

Component:	Afbeelding:						
Temperatuur sensor							
Draadbrug							
Draadbrug							
Draadbrug							
Draadbrug							
Draadbrug							
Draadbrug							

# 7



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → **Circuit\_07**



Notitie code:

**Serial.begin(9600);**

Voordat je de seriële monitor kunt uitlezen, moet deze eerst worden geïnitialiseerd met `Serial.begin()`. 9600 is de 'baud rate', of communicatie-snelheid. Wanneer 2 apparaten met elkaar communiceren moeten deze dezelfde communicatie-snelheid hebben.

**Serial.print(degreesC);**

Het `Serial.print()` commando kan bijna alles printen wat er tussen de haakjes staat, inclusief variabelen van allerlei types, tekst (oftewel "strings"), etc.

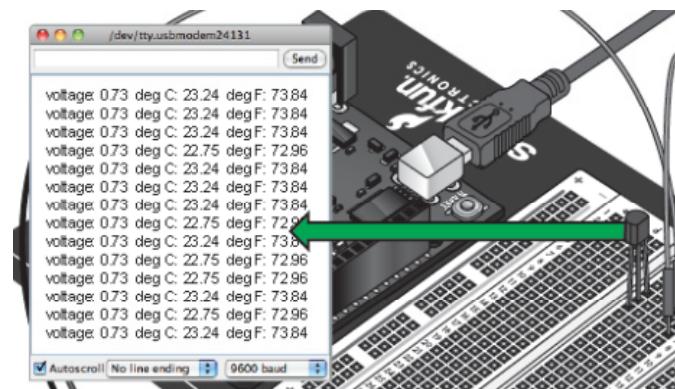
Zie <http://arduino.cc/en/serial/print> voor meer info.

**Serial.println(degreesF);**

`Serial.print()` print alles op dezelfde regel (achter elkaar). `Serial.println()` springt naar een volgende regel. Een combinatie van beide geeft je de mogelijkheid om makkelijk leesbare print-outs te maken van tekst en data.

## Wat gebeurt er nu:

In de seriële monitor zou je nu de temperatuur kunnen aflezen, dat door de temperatuur sensor wordt waargenomen, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hieronder worden beschreven.



### Tips:

#### Er lijkt niets te gebeuren

Dit circuit heeft geen uitwendige indicatie, dat het werkt, zoals een LED. Het resultaat kan alleen worden waargenomen in de seriële monitor van de Arduino IDE.

#### Niet-leesbare output

Dit gebeurt meestal als de communicatie-snelheid van het programma met de seriële monitor niet hetzelfde is. Stel de baud-rate op 9600.

#### Temperatuur verandert niet

Raak de sensor aan met je vingers of houd een zakje met ijs er tegenaan. Temperatuurverandering is niet zichtbaar als de omgevingstemperatuur niet verandert.

### Applicatie in de echte wereld:

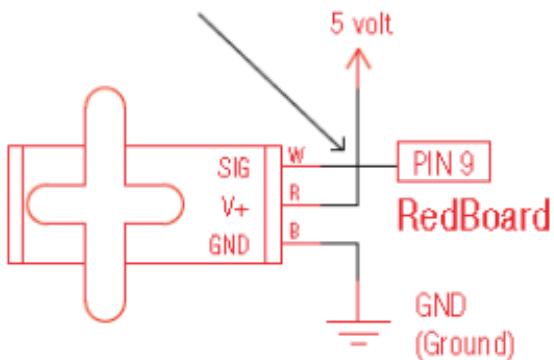
Temperatuur controle systemen gebruiken sensoren om de temperatuur op een ingestelde waarde te houden.



## Circuit #8

8

No junction dot means no connection

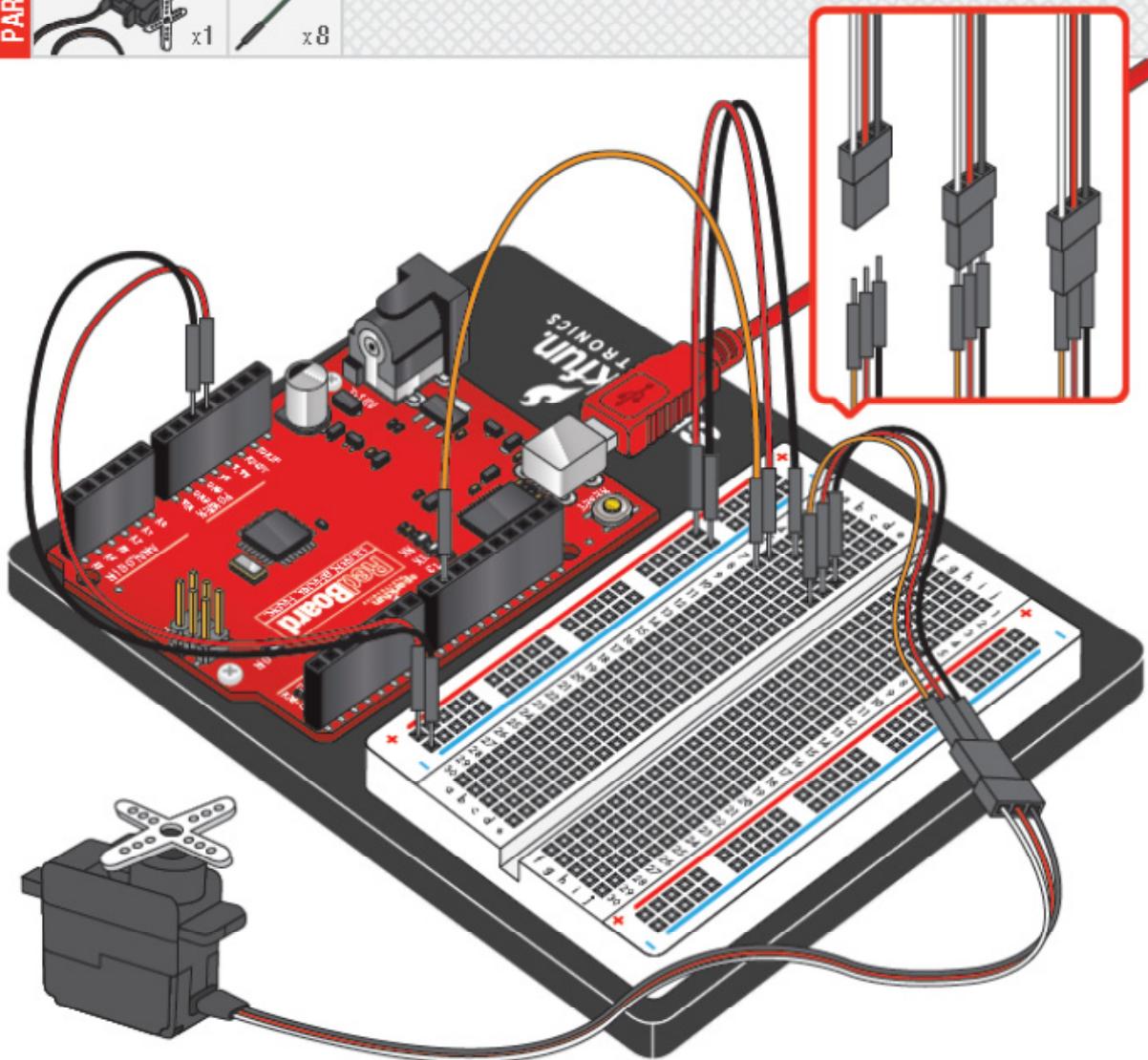


### Een enkele servo

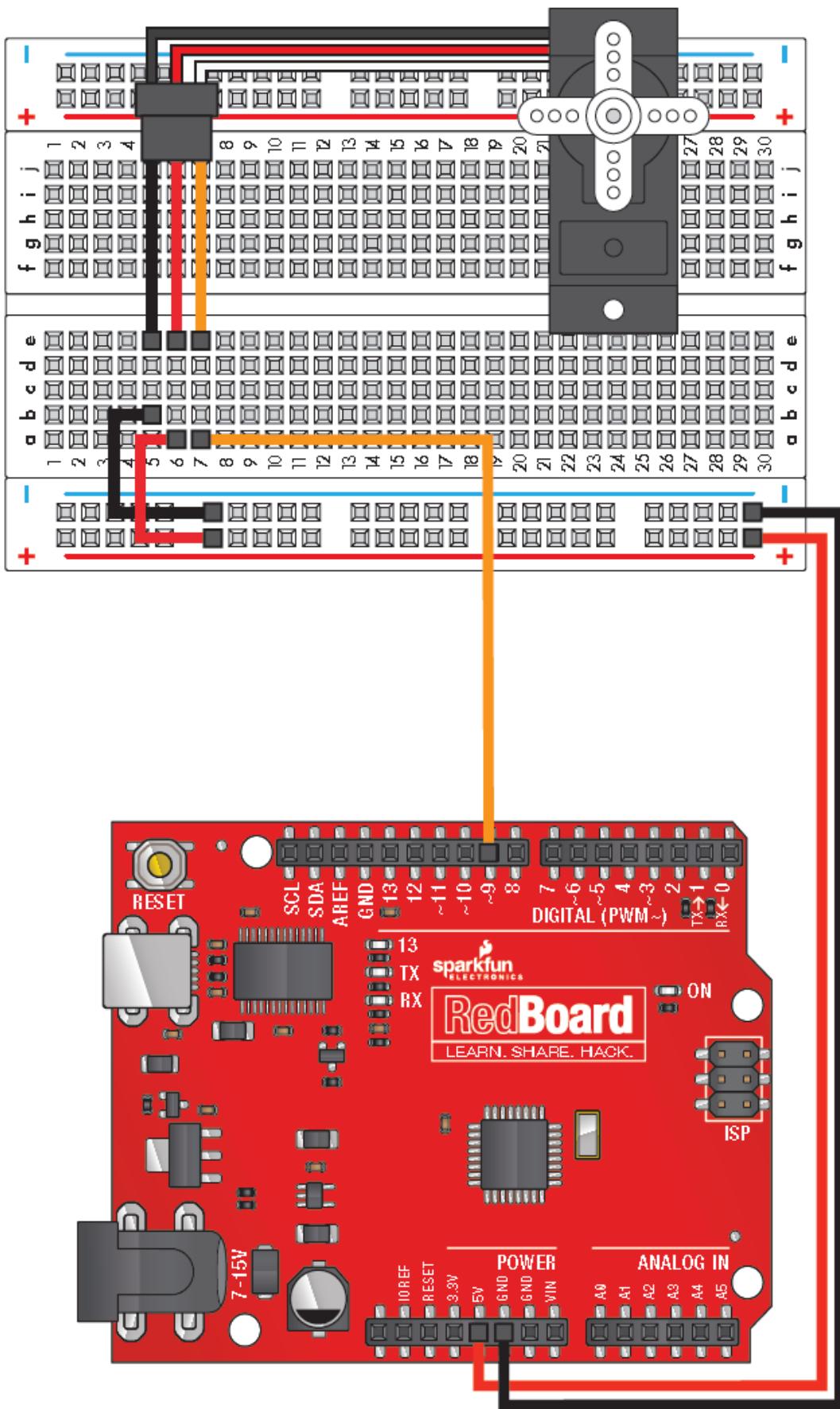
Servo's zijn ideaal voor bepaalde elektronische applicaties, omdat ze iets kunnen dat motoren niet kunnen, de as verplaatsen naar een exact ingestelde positie. Door het wijzigen van de pulse width van het voltage kan de servo naar een specifieke positie bewegen. Bv. een pulse van 1.5 milliseconden draait de servo-koppeling 90°. In dit circuit leer je PWM (pulse width modulation) te gebruiken om een servomotor te controleren en met precisie te roteren.

PARTS:

- Servo x1
- Wire x8



Circuit 8 Een enkele servo



	<b>Afbeelding:</b>			<b>Vergroot je horizon met 'Libraries':</b>
Servo				De Arduino omgeving geeft je een grote set van bruikbare, ingebouwde commando's voor basale input en output, het maken van beslissingen met logica, oplossen van mathematische problemen, etc. Maar de grootste kracht zit in de Arduino community, die graag hun werk met jou willen delen.
Draadbrug				'Libraries' (bibliotheeken) zijn nieuwe commando's die bij elkaar zijn gezet in stukjes code, die makkelijk in jouw sketch kunnen worden geïntegreerd. Arduino heeft al een handvol bibliotheken na de installatie, zoals een Servo bibliothek in dit voorbeeld, dat gecombineerd kan worden met andere devices (LCD displays, stepper motoren, ethernet poorten, etc.)
Draadbrug				Iedereen kan bibliotheken maken, die je dan weer kunt delen met anderen. Er zijn veel bibliotheken te vinden voor allerhand devices. Deze kun je vinden via Google of op de Arduino PlayGround via <a href="http://arduino.cc/playground/">http://arduino.cc/playground/</a> .
Draadbrug				
Draadbrug			<b>Pin 9</b>	
Draadbrug			<b>-</b>	
Draadbrug			<b>+</b>	
Draadbrug			<b>5V</b>	
Draadbrug			<b>GND</b>	



Om een bibliotheek te gebruiken: **Sketch > Importeer Bibliotheek.**



Notitie code:

**#include <Servo.h>**

#include is een speciaal "preprocessor" commando, dat een bibliotheek of andere file aan jouw sketch toevoegt. Je kunt het commando zelf intypen of een bestaande bibliotheek kiezen via menu "Schets > Importeer Bibliotheek".

**Servo servol;**

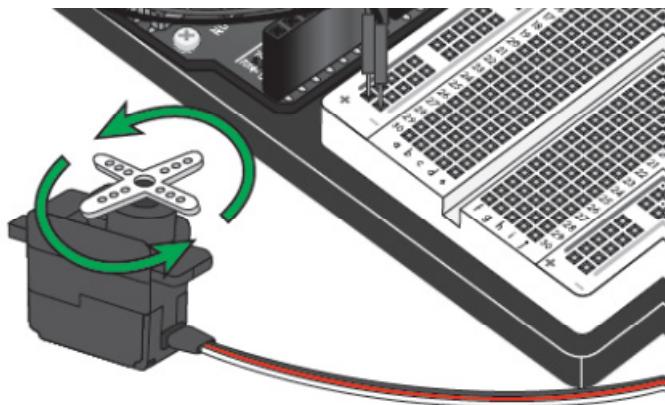
De Servo bibliotheek voegt nieuwe commando's toe om een servo te controleren. Arduino moet geïnformeerd worden over de servo. Eerst wordt een servo "object" gemaakt voor elke servo (hier 'servo1') en wordt dan verbonden met "attach" aan een digitale pin (hier pin 9).

**servo1.write(180);**

De servo in deze kit kan niet helemaal ronddraaien, maar naar een bepaalde positie worden bewogen. We gebruiken de write() in de servo bibliotheek om de servo een bepaalde hoeveelheid graden te laten draaien (0 to 180). Het draaien kost tijd, dus zorg altijd voor een delay() in de code.

### **Wat gebeurt er nu:**

Je zou de servo motor moeten zien draaien naar een ingestelde positie met een bepaalde snelheid, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven.



### **Tips:**

#### **Servo draait niet**

Zelfs met gekleurde draden kan het voorkomen, dat de servo niet goed is verbonden. Controleer de verbindingen.

#### **Het werkt nog steeds niet**

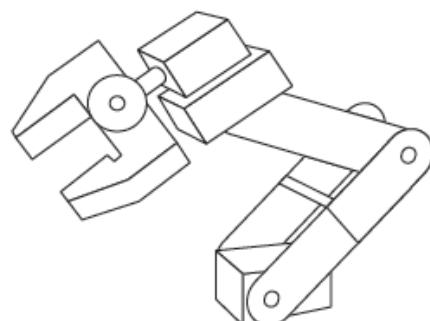
Vergis je ervan, dat ook de voeding goed is aangesloten en überhaupt is aangesloten.

#### **Beweging onregelmatig**

Als de servo niet goed draait of hakkeld draait, dan is de voeding via de USB niet toereikend om de servo goed te laten werken. Gebruik een voedingskabel i.p.v. de USB-voeding om dit probleem op te lossen.

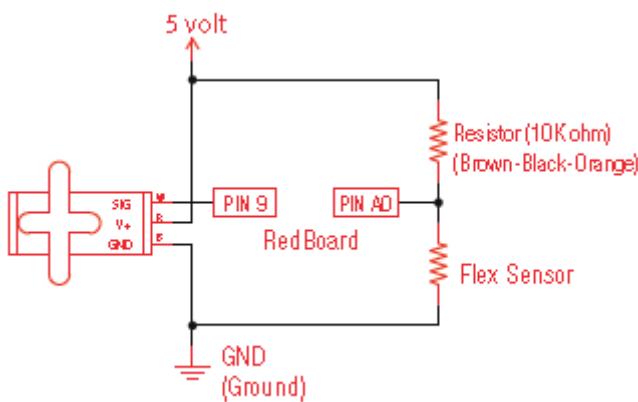
### **Applicatie in de echte wereld:**

Robot armen, die je ziet bij lopende banden of sci-fi films hebben waarschijnlijk servo's ingebouwd.



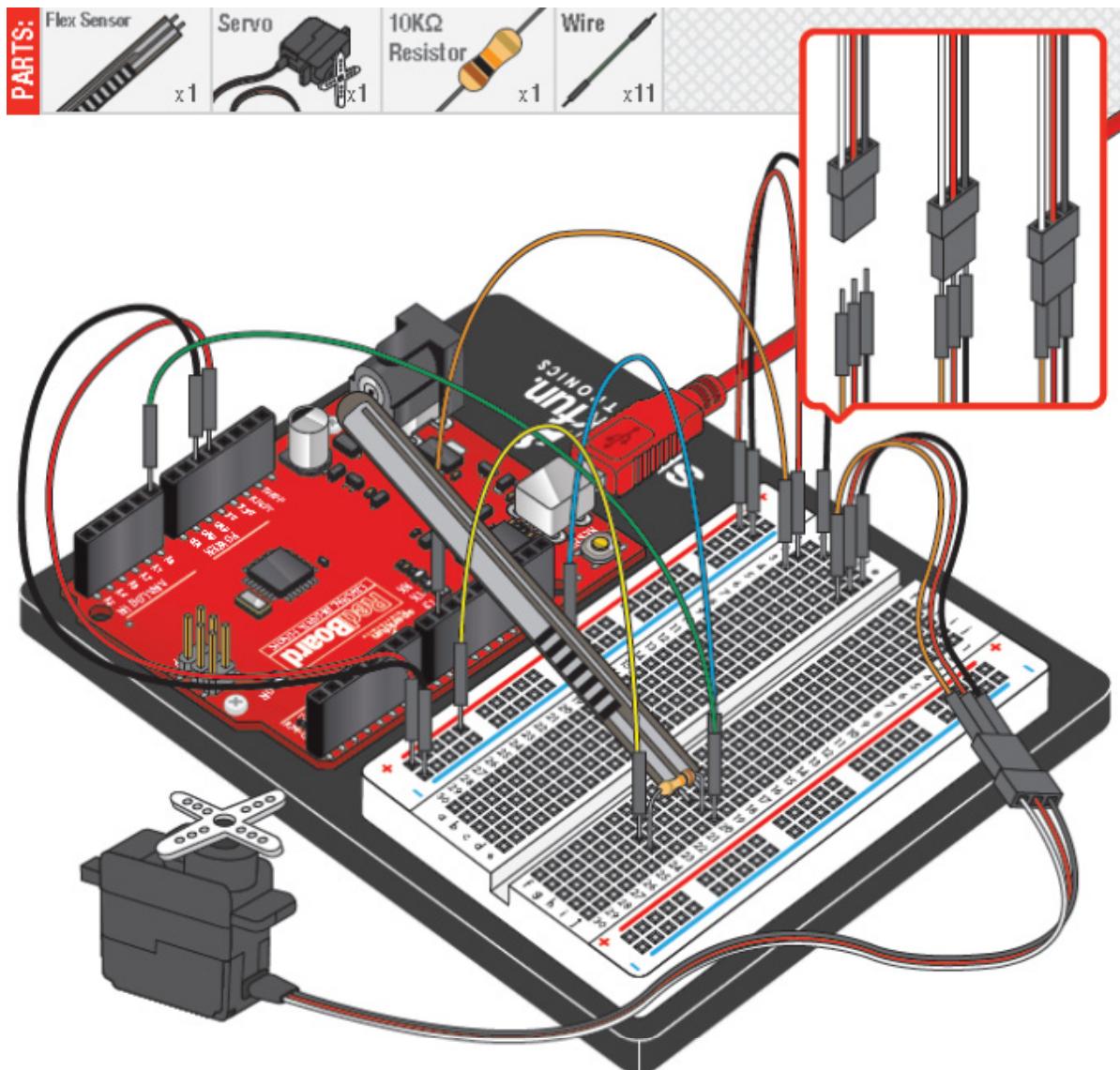
## Circuit #9

9

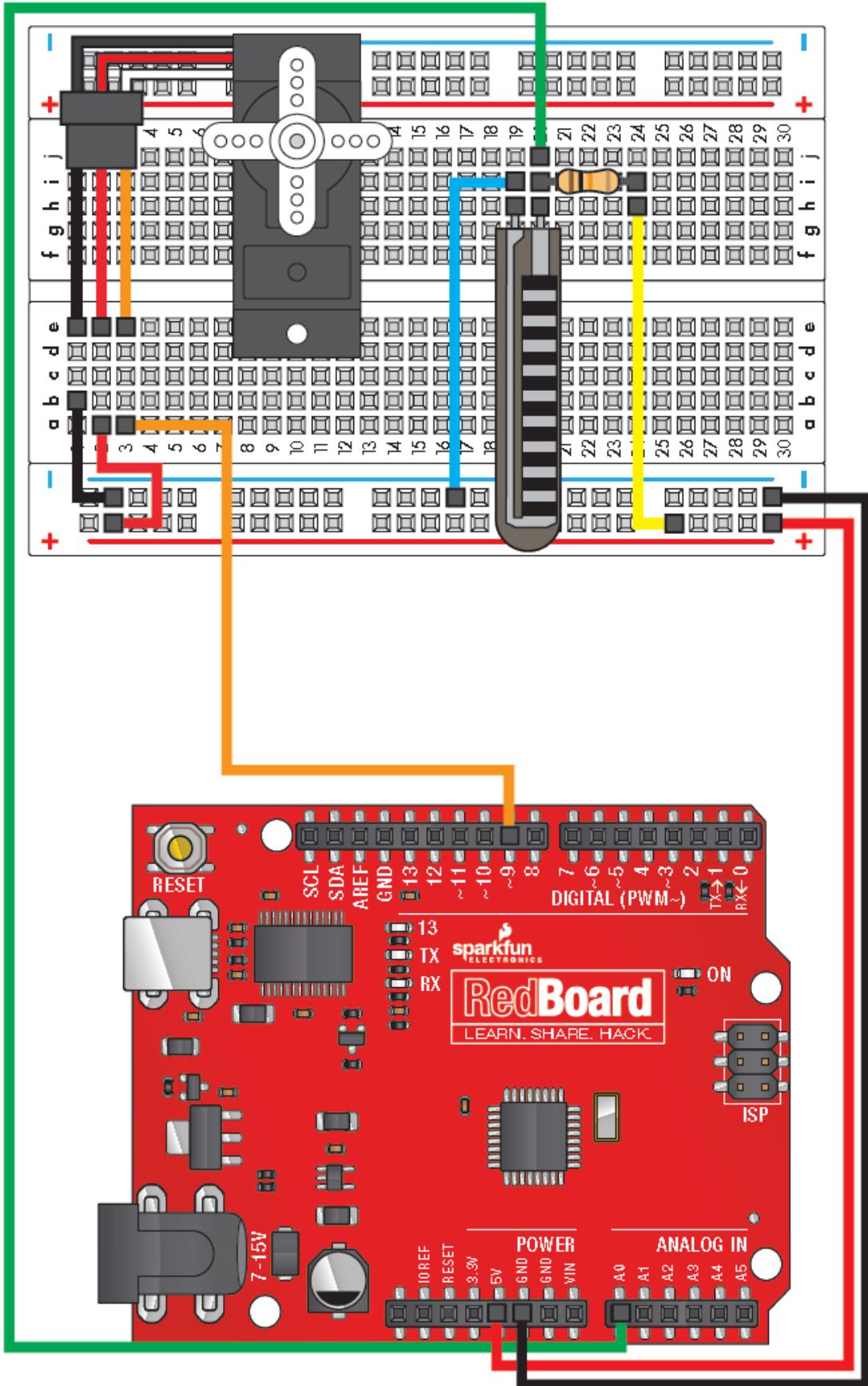


### Flex sensor

Servo's zijn ideaal voor bepaalde elektronische applicaties, omdat ze iets kunnen dat motoren niet kunnen, de as verplaatsen naar een exact ingestelde positie. Door het wijzigen van de pulse width van het voltage kan de servo naar een specifieke positie bewegen. Bv. een pulse van 1.5 milliseconden draait de servo-koppeling 90°. In dit circuit leer je PWM (pulse width modulation) te gebruiken om een servo-motor te controleren en met precisie te roteren.



Circuit 9: Flex Sensor



Debugging een sketch m.b.v. de seriële monitor:	
Het kan altijd gebeuren - je schrijft een sketch, dat succesvol compileert en wordt geupload, maar zonder dat je het weet, het niet precies doet wat het moet doen. Met grote computers, die beeldschermen, toetsenborden, en muisen heeft, kun je je code debuggen, maar een Arduino board heeft dat allemaal niet.	
De sleutel tot zichtbaarheid van een microcontroller is de output. Dit kan van alles zijn, inclusief LEDs en piepers, maar één van de meest behulpzame tools is de seriële monitor. Met Serial.print() en println() kun leesbare output genereren van de Arduino board naar een venster op de computer. Dit is prima voor de normale output van een sketch, maar is ongelooflijk bruikbaar voor debugging.	<pre>for (x=0; x&lt;8; x++) {   Serial.print(x); }</pre> <p>Stel je wilt een for() loop van 1 tot 8, maar de code lijkt niet goed te werken. Voeg <b>Serial.begin(9600)</b>; toe aan de <b>setup()</b> functie en een <b>Serial.println()</b> aan de loop:</p>
	<pre>for (x=1; x&lt;9; x++) {   Serial.print(x); }</pre> <p>Als je de code nu opnieuw runt, dan is de output precies zoals je wilde: van 1 tot 8.</p>

Component:	Afbeelding:		
Servo		e1   e2   e3	
Draadbrug		e1   e2   e3	
Draadbrug		e2   e3	
Draadbrug		e1   e2   e3	
Draadbrug		e1   e2   e3	
Flex sensor		h19   h20	
10kΩ weerstand		i20   i24	
Draadbrug		j19   j20	
Draadbrug		j20   j20	
Draadbrug		b1   -	
Draadbrug		h24   +	
Draadbrug		a2   +	
Draadbrug		a3   -	
Draadbrug		5V   +	
Draadbrug		GND   -	

# 9



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → [Circuit\\_09](#)



Notitie code:

```
servoposition=map(flexposition,600,900,0,180);  
map(value, fromLow, fromHigh, toLow, toHigh)
```

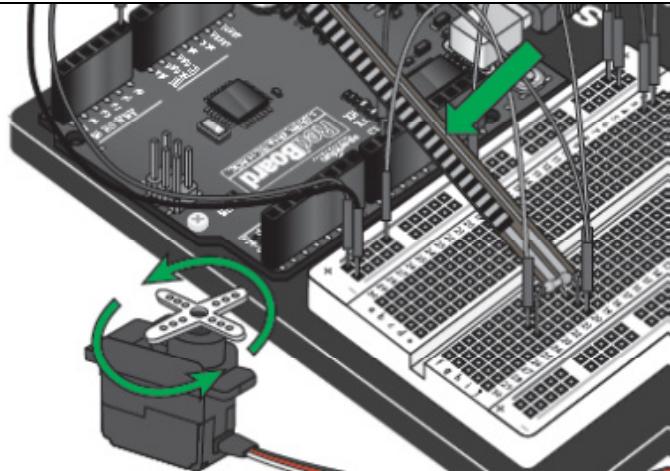
Omdat de flexsensor-weerstand combinatie niet de volledige range van 0 tot 5V geeft, gebruiken we map() om de range aan te passen. Hier worden alleen waarden geaccepteerd tussen 600 en 900 i.p.v. 0 en 1023.

```
servoposition=constraint(servoposition,0,180);  
constraint(x, a, b)
```

De map() functie kan nog steeds waarden teruggeven tussen 0 en 1023. Met constraint beperken we de output, zodat het binnen de range blijft. Buiten de range wordt de waarde teruggebracht naar de kleinste of grootste waarde.

## Wat gebeurt er nu:

De draairichting van de motor zou nu in overeenstemming moeten zijn met de buiging van de flex sensor, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven.



### Tips:

#### Servo draait niet

Zelfs met gekleurde draden kan het voorkomen, dat de servo niet goed is verbonden. Controleer de verbindingen.

#### Servo draait niet als verwacht

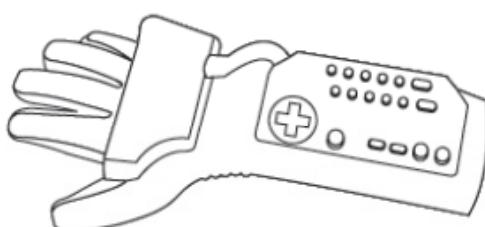
De sensor werkt maar in één richting. Probeer de sensor de ander kant op te buigen.

#### Servo beweegt niet veel

Je moet de range waarden in de map() functie aanpassen.

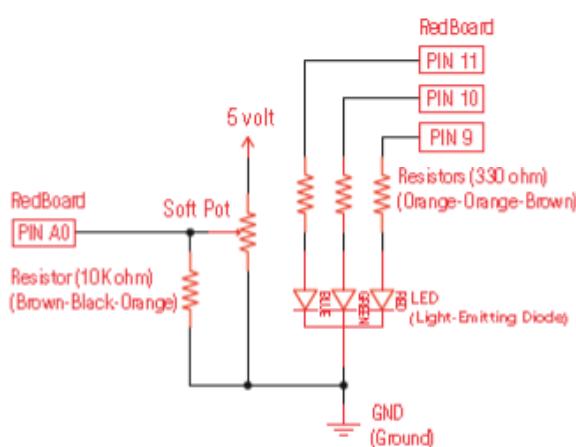
### Applicatie in de echte wereld:

Controller accessoires voor videogames, zoals de 'Power Glove', gebruiken flex-sensors. Het was de eerste videogame, die handbewegingen op het scherm in 'real-time' probeerde na te doen.



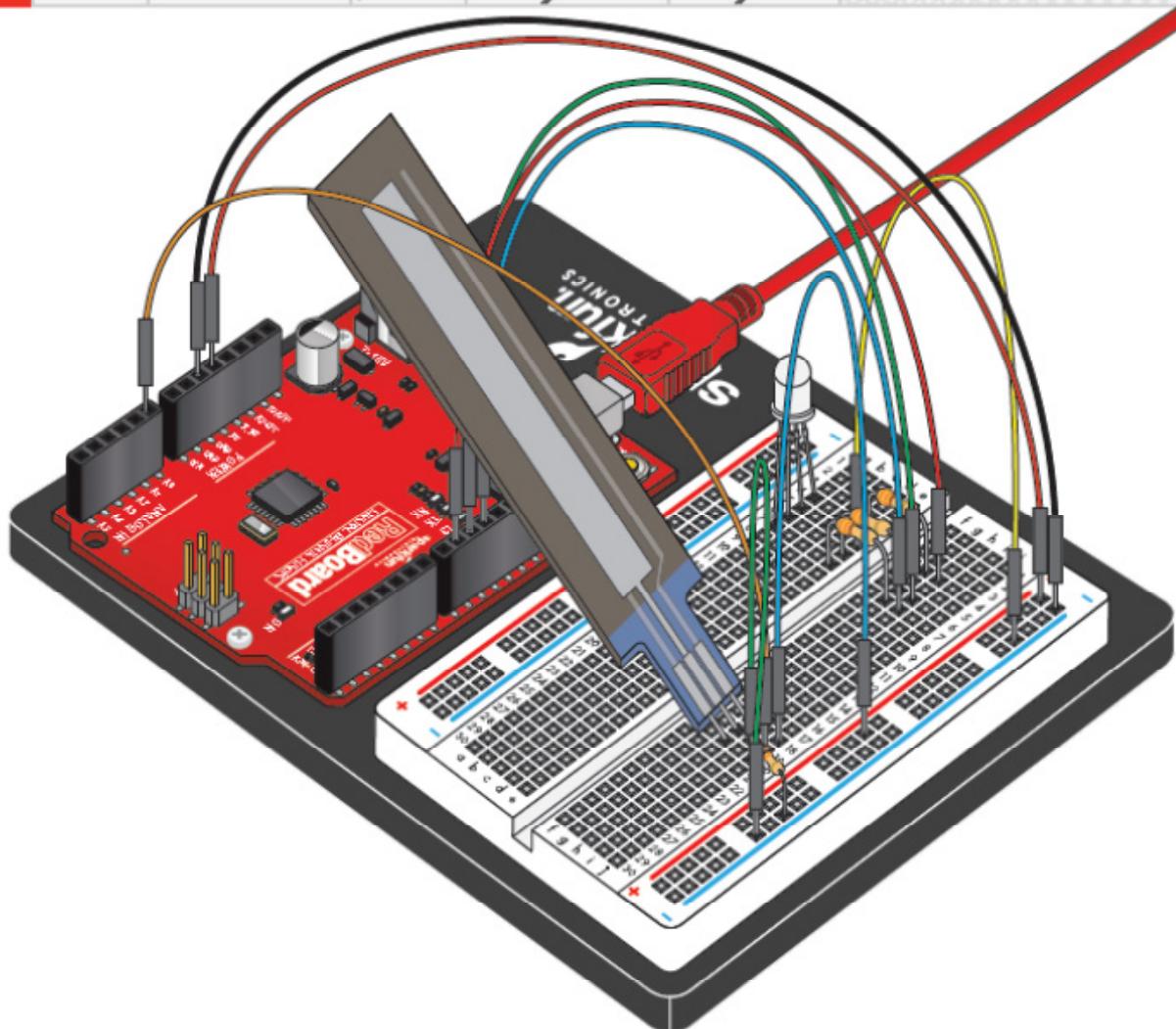
## Circuit #10

# 10

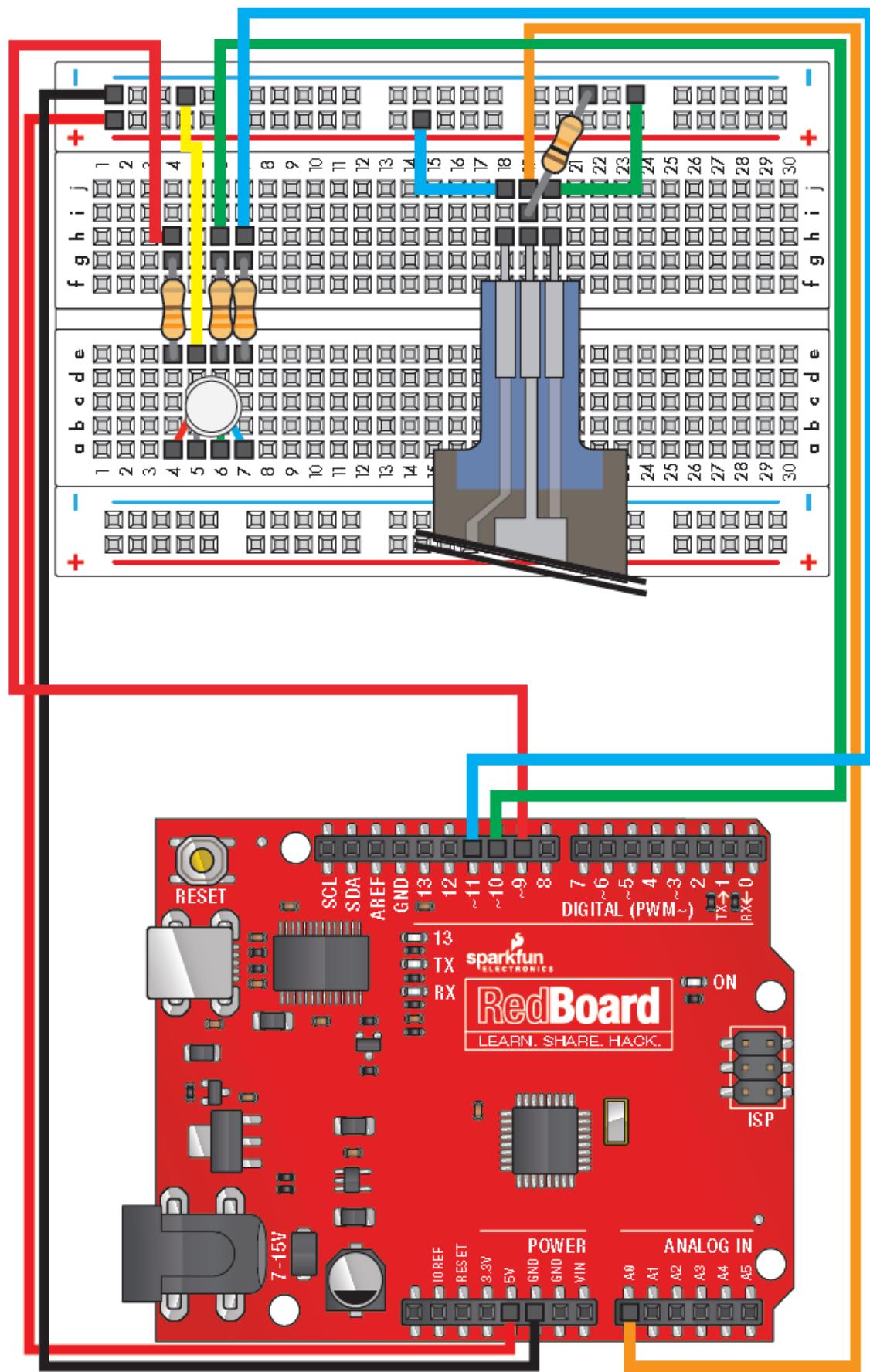


### Zachte potentiometer

Servo's zijn ideaal voor bepaalde elektronische applicaties, omdat ze iets kunnen dat motoren niet kunnen, de as verplaatsen naar een exact ingestelde positie. Door het wijzigen van de pulse width van het voltage kan de servo naar een specifieke positie bewegen. Bv. een pulse van 1.5 milliseconden draait de servokoppeling 90°. In dit circuit leer je PWM (pulse width modulation) te gebruiken om een servo-motor te controleren en met precisie te roteren.



Circuit 10: Zachte potentiometer



Component:	Afbeelding:	Component:	Afbeelding:
RGB LED (5 mm)			
Zachte potentiometer			
330Ω weerstand			
330Ω weerstand			
330Ω weerstand			
10kΩ weerstand			
Draadbrug			
Draadbrug			
Draadbrug			
Draadbrug			
Draadbrug			
Draadbrug			
Draadbrug			
Draadbrug			
Draadbrug			
Draadbrug			
Draadbrug			

# 10



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → [Circuit\\_10](#)



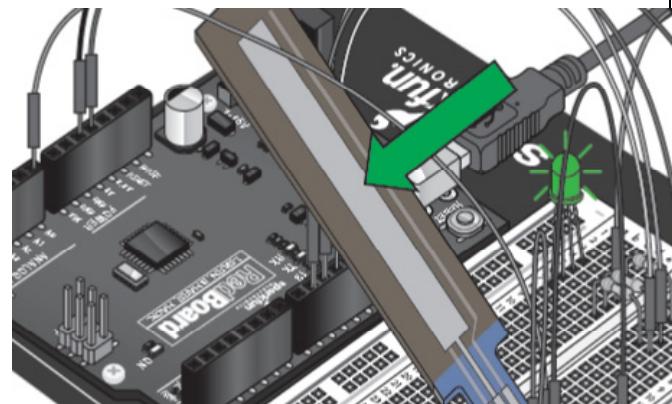
Notitie code:

```
redValue = constrain(map(RGBposition, 0, 341, 255, 0), 0, 255)  
+ constrain(map(RGBposition, 682, 1023, 0, 255), 0, 255);  
  
greenValue = constrain(map(RGBposition, 0, 341, 0, 255), 0, 255)  
- constrain(map(RGBposition, 341, 682, 0, 255), 0, 255);  
  
blueValue = constrain(map(RGBposition, 341, 682, 0, 255), 0, 255)  
- constrain(map(RGBposition, 682, 1023, 0, 255), 0, 255);
```

Deze lange, vreemde functies sturen een RGB led aan en berekenen de drie RGB waarden om een regenboog van kleuren te maken. Ze maken drie 'pieken' voor rood, groen en blauw, die elkaar overlappen en gemixt worden voor nieuwe kleuren. Bekijk de code! Ook als je niet 100% weet hoe het werkt, kun je de code hiernaast kopiëren en in je eigen code gebruiken en aan te passen. Als je meer wilt weten over het creëren van je eigen functies, kijk dan naar circuit #11.

## Wat gebeurt er nu:

Je zou de kleur van de RGB LED moeten zien veranderen aan de hand van de positie waar de zachte potentiometer wordt aangeraakt, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hieronder worden beschreven.



### Tips:

#### LED blijft donker of laat verkeerde kleuren zien

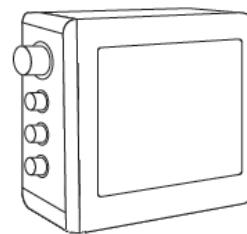
Met de vier pinnen van de LED kan het gebeuren, dat ze verkeerd zijn aangesloten. Controleer de verbindingen.

#### Bizarre resultaten

Dit gebeurt meestal als de potentiometer op meerdere plekken wordt aangeraakt. Dit is normaal en kan gebruikt worden om bizarre resultaten te genereren.

### Applicatie in de echte wereld:

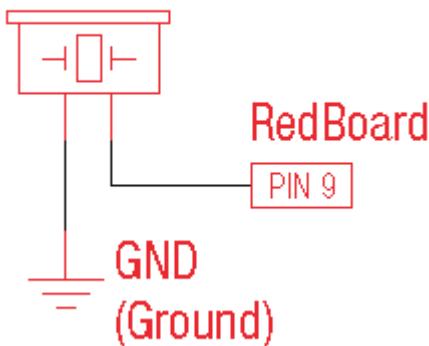
De knoppen op verschillende objecten, zoals die van een radio, gebruikt het concept dat in dit circuit wordt besproken.



## Circuit #11

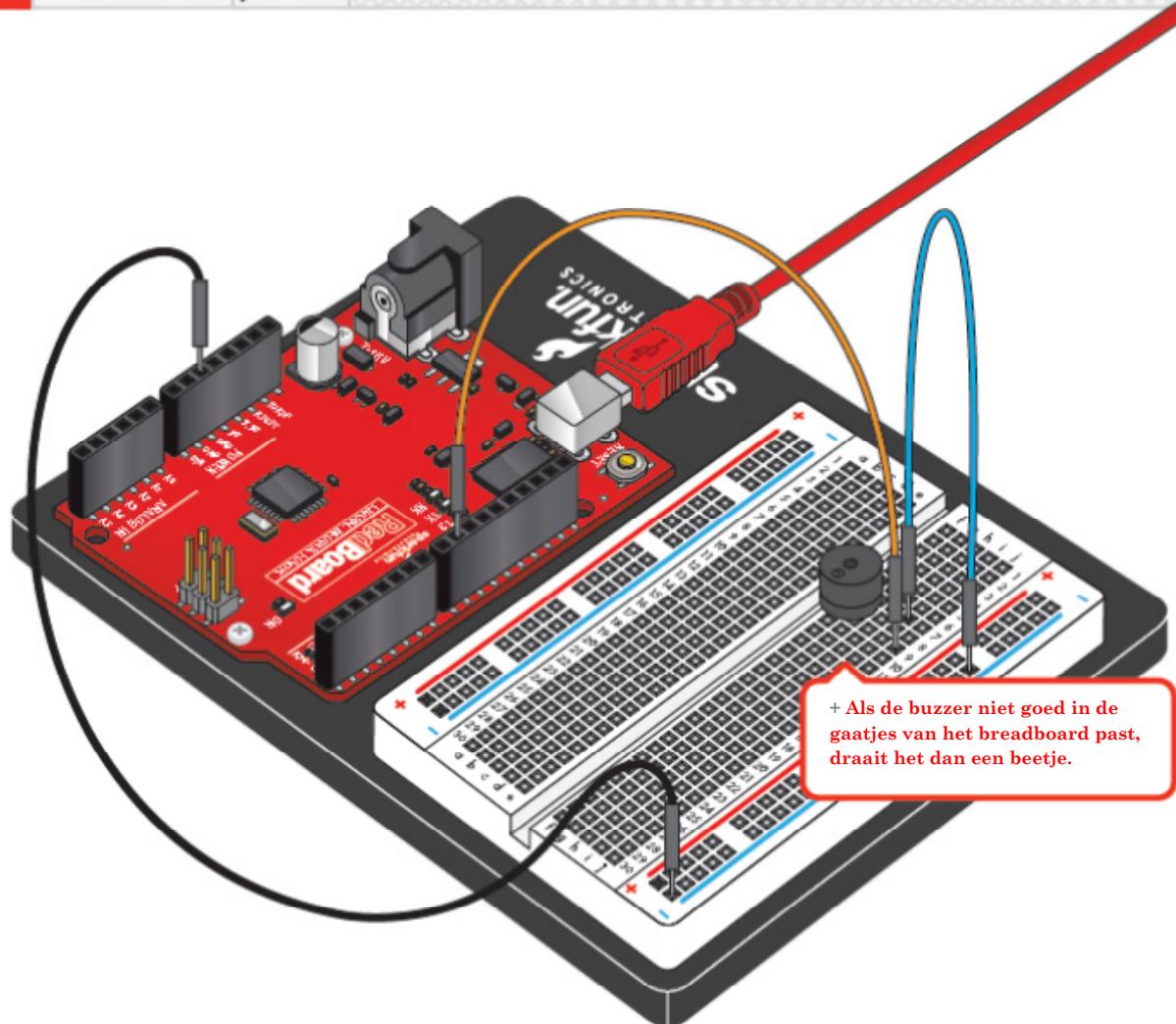
# 11

### Piezo Buzzer



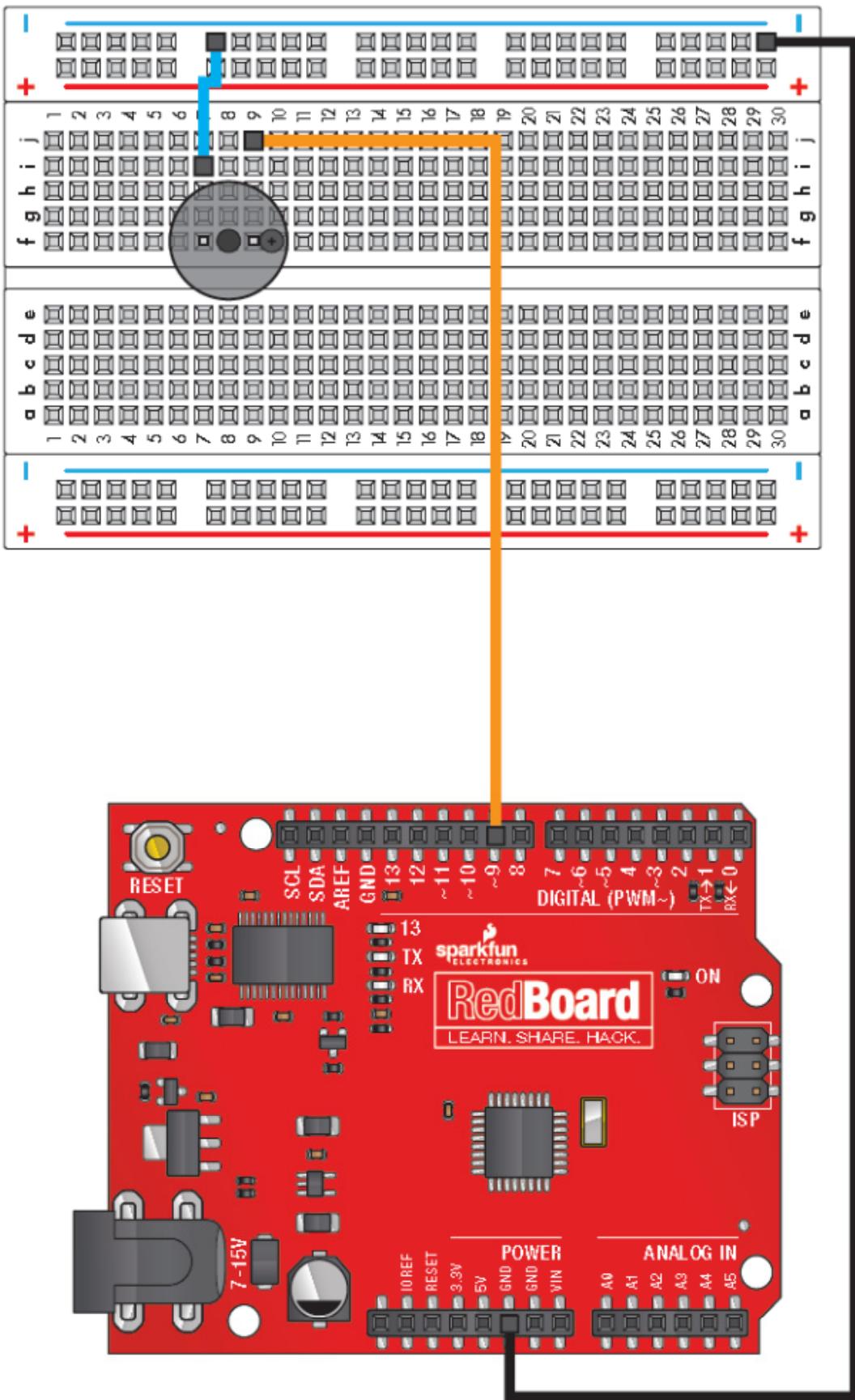
### Piëzo buzzer

In dit circuit proberen we een brug te slaan tussen de digitale en analoge wereld. We gebruiken een buzzer, dat een kleine klik geeft op het moment dat er een voltage door loopt. Dit alleen is nou niet echt spannend te noemen. Maar als je honderden keren per seconde het voltage in- en uitschakelt, is het mogelijk om de buzzer een toon te laten produceren. En als je meerdere tonen achter elkaar genereert, krijg je muziek! Dit circuit en de sketch speelt een klassiek deuntje.



+ Als de buzzer niet goed in de gaatjes van het breadboard past, draait het dan een beetje.

Circuit 11: Piezo buzzer



Schrijf je eigen functies:	
	<p>Arduino bevat al een grote hoeveelheid ingebouwde functies, die bruikbaar zijn voor allerlei dingen. (Zie <a href="http://arduino.cc/en/reference">http://arduino.cc/en/reference</a> voor een lijst). Maar je kunt ook op makkelijk je eigen functies schrijven.</p> <p>Eerst moeten we een functie declareren. Hier is een simpel voorbeeld met de naam optellen', dat twee nummers optelt en het resultaat teruggeeft. Laten we het in detail bekijken.</p> <pre>int add(int getal1, int getal2)  {     int x      x = getal1 + getal2;      return(x); }</pre> <p>Deze functie neemt getallen op ("getal"), en geeft (return) een waarde terug. Als je parameters meestuurt in de functie, vermeldt dan ook het type tussen de haakjes na de functienaam. Zonder parameters worden alleen de haakjes geschreven.</p> <p>Als de functie ook nog een waarde teruggeeft, vermeldt dan het type van de waarde die terug wordt gegeven voor de naam van de functie (hier 'int'). De terugwaarde wordt tussen haakjes na de return statement geschreven (hier 'return(x)'). De functie zonder een waarde terug te geven wordt voorafgegaan met 'void' net als in de declaraties voor setup() en loop() functies.</p> <p>Als je eigen functies wilt schrijven, kun je je eigen code netter schrijven en ook hergebruikt worden. Zie <a href="http://arduino.cc/en/reference/functiondeclaration">http://arduino.cc/en/reference/functiondeclaration</a> voor meer informatie over functies.</p>

Component:	Afbeelding:				
Piezo buzzer					
Draadbrug					
Draadbrug					

# 11



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → [Circuit\\_11](#)



Notitie code:

```
char notes[] = "cdfda ag cdfdg gf";  
char names[] = {'c','d','e','f','g','a','b','C'};
```

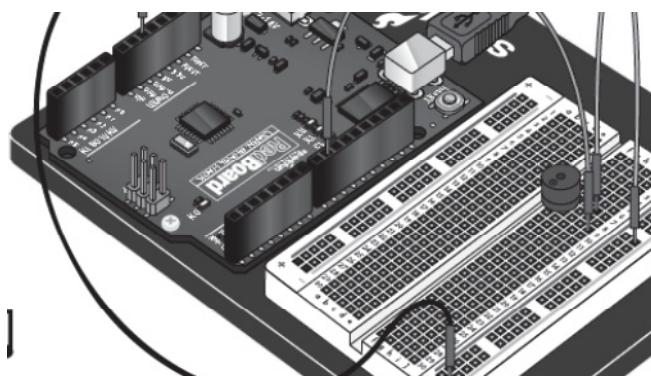
Tot nu toe hebben we alleen nog maar gewerkt met getallen, maar de Arduino kan ook met letters werken. Karakters (enkele, afdruckbare, letters, nummer en andere symbolen) hebben een eigen type, genaamd "char". Een array van karakters kan worden gedefinieerd door ze tussen dubbele aanhalingstekens te plaatsen (ook wel "string" genoemd, of als een lijst met karakters tussen enkele aanhalingstekens).

```
tone(pin, frequency, duration);
```

Eén van de vele bruikbare, ingebouwde commando's van de Arduino is de tone() functie. Deze functie zorgt voor een zekere frequentie aan de output pin, hoorbaar via buzzers of speakers. Als je een tijdsduur (in milliseconden) opgeeft, wordt de toon gespeeld en stopt weer. Zonder tijdsduur blijft de toon hoorbaar, maar worden gestopt met een andere functie, noTone().

## Wat gebeurt er nu:

Je ziet niets gebeuren, maar je zou wel een liedje moeten horen, dat wordt afgespeeld, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven.



### Tips:

#### Geen geluid

Door de grootte van de buzzer is moeilijk te zien waar de pinnen zitten. Controleer de verbindingen.

#### Kan niet denken als de melodie speelt

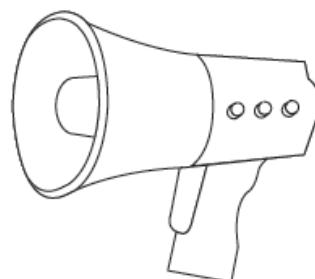
Haal de piezo-buzzer uit het breadboard, upload het programma en steek de buzzer weer terug.

#### Voel me een beetje verloren

De code is zo geschreven, dat het makkelijk is om je eigen melodie te schrijven en te uploaden.

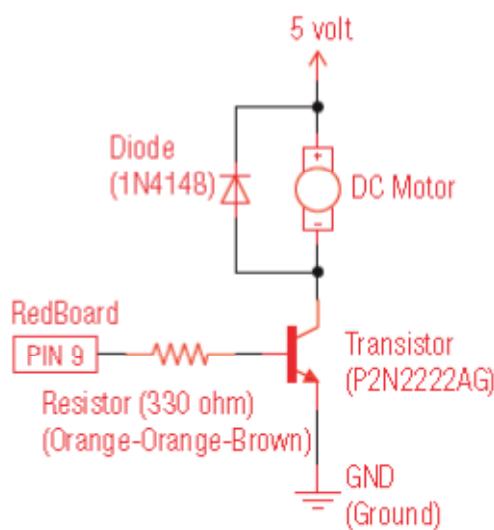
### Applicatie in de echte wereld:

In de hedendaagse megafoons zitten schakelingen, die een buzzer extra versterkt. Ze zijn meestal zo luid, dat je zeker aandacht krijgt als je zo'n ding gebruikt.



## Circuit #12

# 12

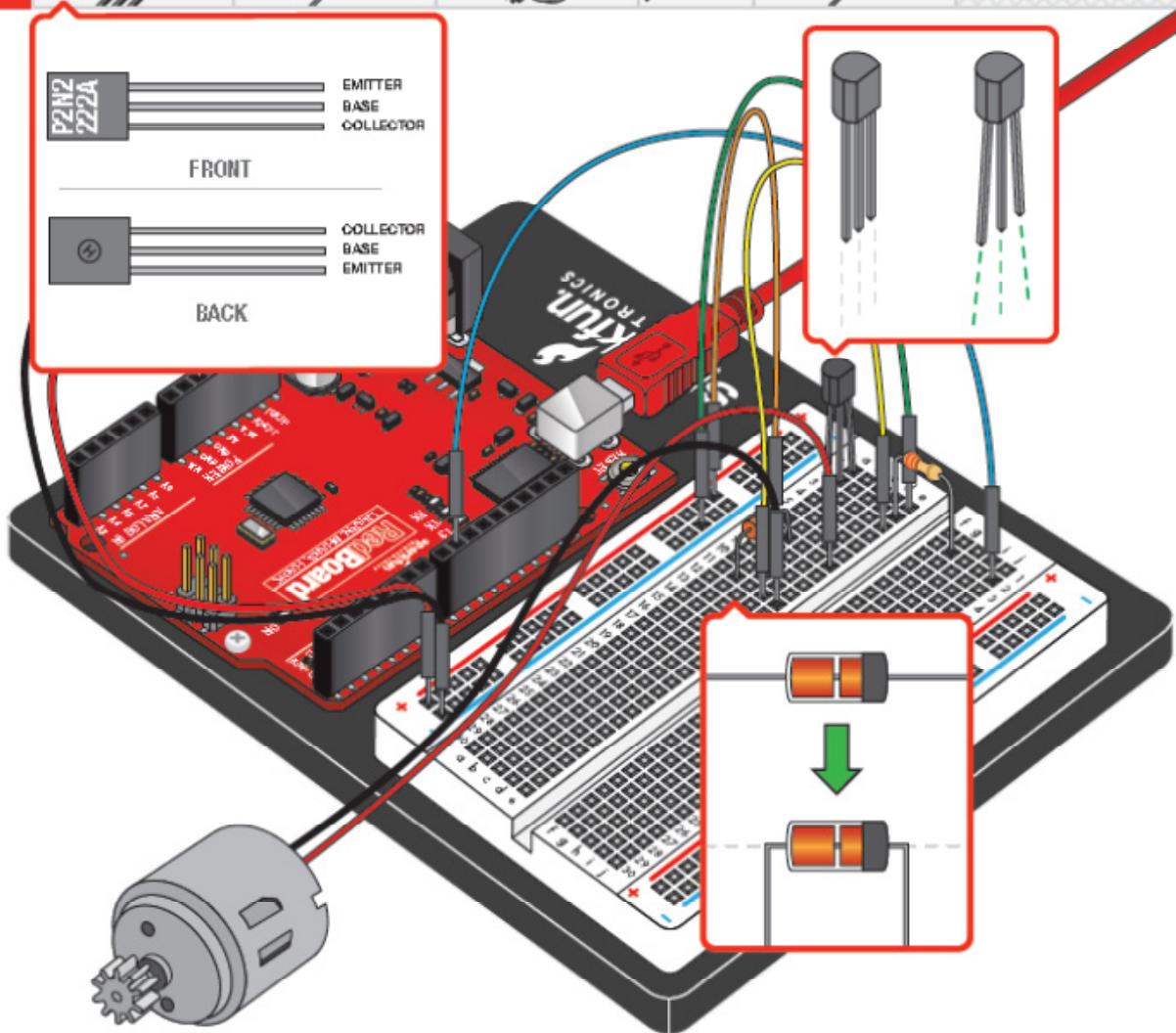


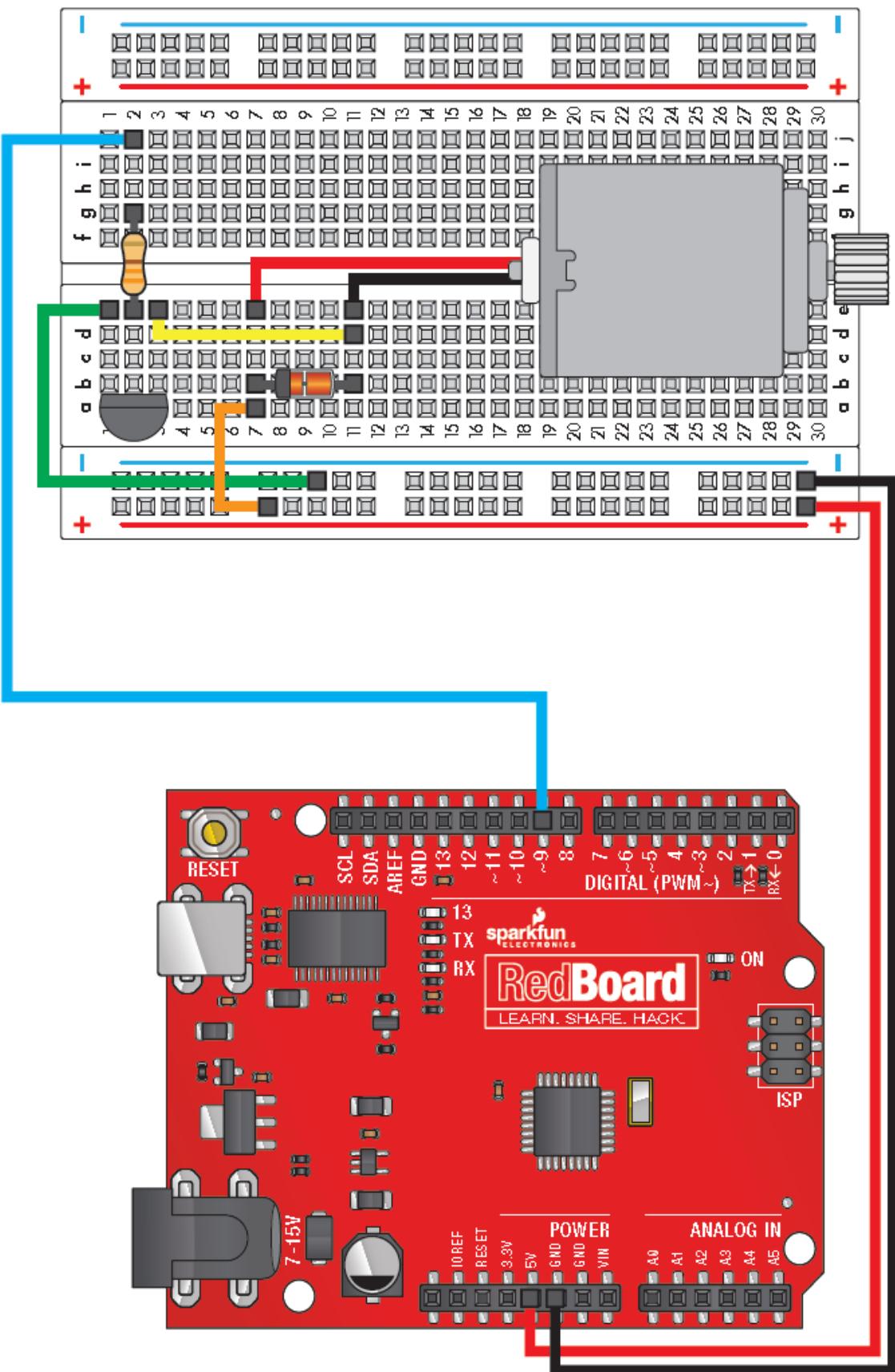
### Een motor draaien

Je hebt al met een servo motor gespeeld. Nu gaan we een motor laten draaien. Hiervoor hebben we een transistor nodig; die kan een hogere stroom laten vloeien dan de Arduino. Als je een tranistor gebruikt, moet je goed letten op de specificaties en er zeker van zijn dat hij de opstelling aan kan. De transistor hier kan tot 40V en 200 mA schakelen - perfect voor deze motor. Wanneer de motor draait en plotseling wordt uitgeschakeld, veroorzaakt het magnetisch veld in de motor een voltage piek. Dit kan de transistor vernielen. Om dat te voorkomen, plaatsen we een 'flyback diode' om de voltage piek om te leiden en de transistor beschermd.



Wanneer je het circuit bouwt, zorg er dan voor dat je de sensor niet verwart met de transistor in het pakket; ze lijken heel veel op elkaar. Kijk naar de tekst "P2N2222A" op de transistor.





Circuit 12: Een motor draaien

Gebruik alles wat je geleerd hebt:	
<p>Op het ogenblik zul je waarschijnlijk al je eigen ideeën hebben om schakelingen in elkaar te zetten, die leuke dingen doen of problemen oplossen. Prima! Hier zijn wat tips over het programmeren in het algemeen.</p> <p>De meeste sketches, die je zult schrijven, zullen uit loops bestaan met enkele van onderstaande stappen of alle stappen:</p> <ol style="list-style-type: none"><li>1. Creëer een vorm van input</li><li>2. Maak calculaties en beslissingen</li><li>3. Creëer een vorm van output</li><li>4. Herhaal (of niet)</li></ol> <p>We hebben al enkele manieren van input sensoren en output laten zien (en er komen er nog een stel). Gebruik en pas je eigen sketches aan en experimenteer - dat is de bedoeling van "Open Source".</p> <p>Normal gesproken is het makkelijk om verschillende sketches te combineren. Daarom is het noodzakelijk om goede programmeer gewoontes te ontwikkelen. Gebruik een constante voor de pins, deel de sketch op in functies: dat maakt hergebruik van sketches aanzienlijk simpeler. Bijv. als je 2 sketches dezelfde pin gebruiken, hoeft je maar één constante te veranderen voor de nieuwe pin. Bedenk wel dat niet alle pinnen analogWrite()</p>	<p>Als je hulp nodig hebt dan kun je bij verschillende Arduino forums vragen stellen en oplossingen vinden. Probeer eens het Arduino forum op <a href="http://arduino.cc/forum">arduino.cc/forum</a> of bij SparkFun's forum.sparkfun.com. Geavanceerde topics kun je vinden op de tutorial website <a href="http://arduino.cc/tutorial">arduino.cc/tutorial</a>.</p> <p>Als je zelf iets hebt gemaakt, deel het dan met anderen via social media of op de website <a href="https://www.sparkfun.com/project_calls">https://www.sparkfun.com/project_calls</a>, zodat we het op de website kunnen zetten.</p>

# 12



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → [Circuit\\_12](#)



Notitie code:

**while (Serial.available() > 0)**

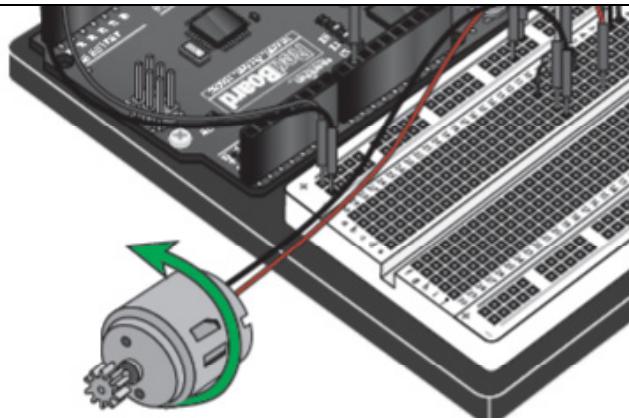
De Arduino seriële poort kan worden gebruikt om data te verzenden en ontvangen. Omdat op elk moment data kan worden verstuurd, zorgt Arduino ervoor om data in een buffer op te vangen en wacht tot het gebruikt kan worden. Het Serial.available() commando controleert het aantal ontvangen karakters, maar nog niet in jouw sketch is gebruikt. Nul betekent, dat er nog geen data is verstuurd.

**speed = Serial.parseInt();**

Als de poort data heeft ontvangen, zijn er verschillende manieren om het te gebruiken. Omdat we nummers hebben ingetypd, dan kan het commando Serial.parseInt() de nummers doorgeven ('parse'). Heb je "1" "0" "0" getypt, dan zal de functie het getal 100 doorgeven.

## Wat gebeurt er nu:

De DC motor zou nu moeten draaien, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hieronder worden beschreven.



## Tips:

### **Motor draait niet**

Als je een andere transistor gebruikt, zorg er dan voor dat de aansluitingen goed zijn en compatibel met de P2N2222. Check de datasheet.

### **Draait nog steeds niet**

Als je ook een andere motor gebruikt, controleer dan of hij werkt op 5V en niet teveel stroom trekt.

### **Het werkt nog steeds niet**

Soms kan het gebeuren, dat de Arduino geen contact meer heeft met de computer. Maak de Arduino los van de computer en verbind de board opnieuw. Meestal wordt de USB-poort weer hersteld.

## Applicatie in de echte wereld:

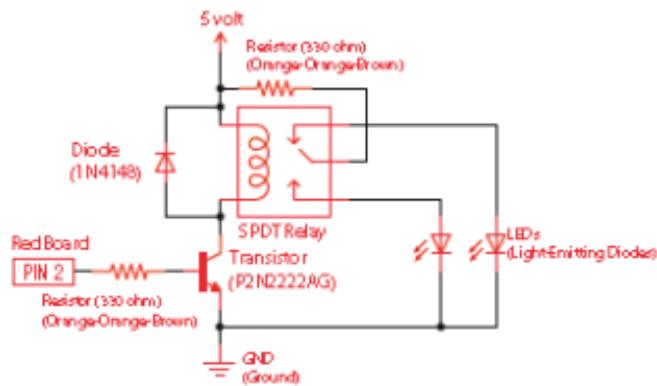
Radio Controlled (RC) auto's gebruiken DC motors om de wielen te laten draaien.



## Circuit #13

# 13

### Relais



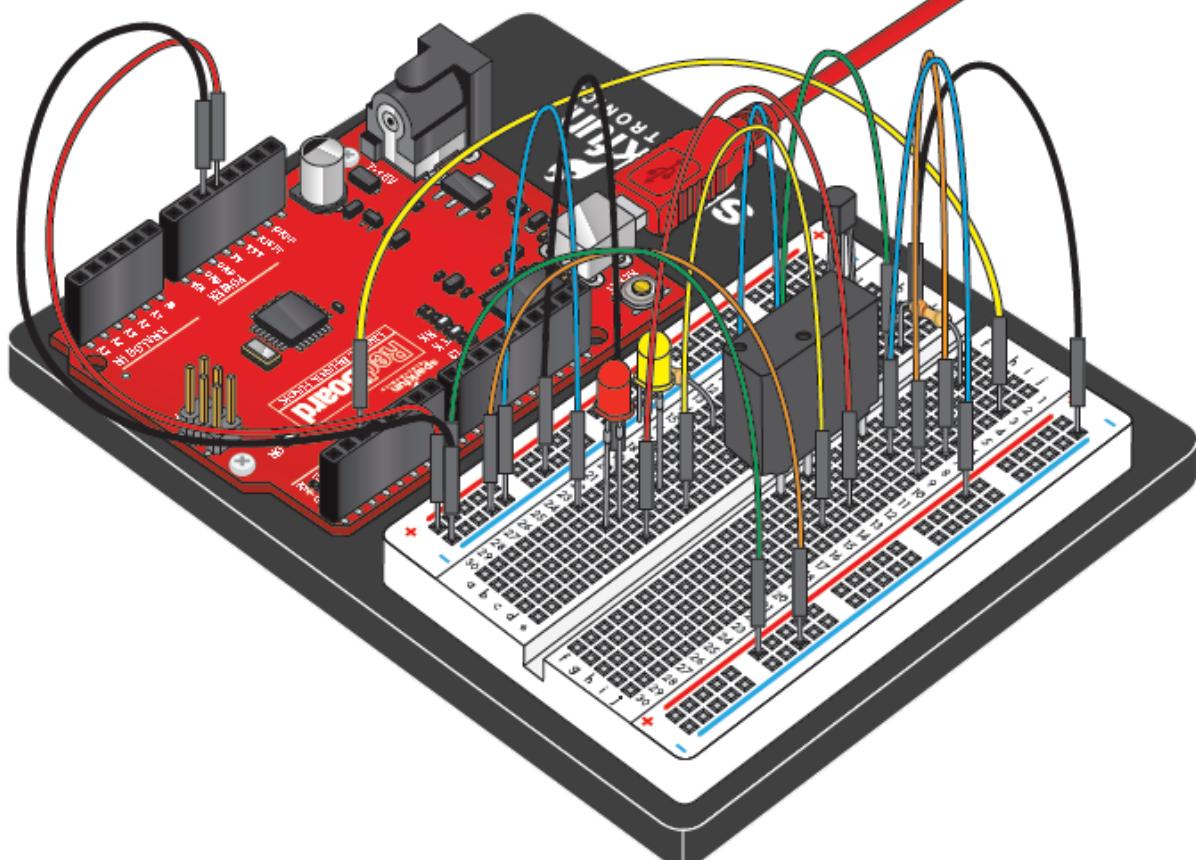
Wat we bij circuit #12 geleerd hebben, gaan we toepassen op het aansturen van een relais. Een relais is een elektrisch-mechanische schakelaar. Een elektromagneet in het relais bedient een schakelaar. Behalve de pinnen voor de spoel, zijn er bij een simpel relais nog 3 pinnen. Een COM (common) pin, NC (normally closed) pin en een NO (normally open) pin.

Een relais geeft de Arduino nog meer power. Met een laagstroom schakeling kan een 220V apparaat in- en uitgeschakeld worden, maar kan ook gebruikt worden om een andere laagstroom circuit te schakelen.

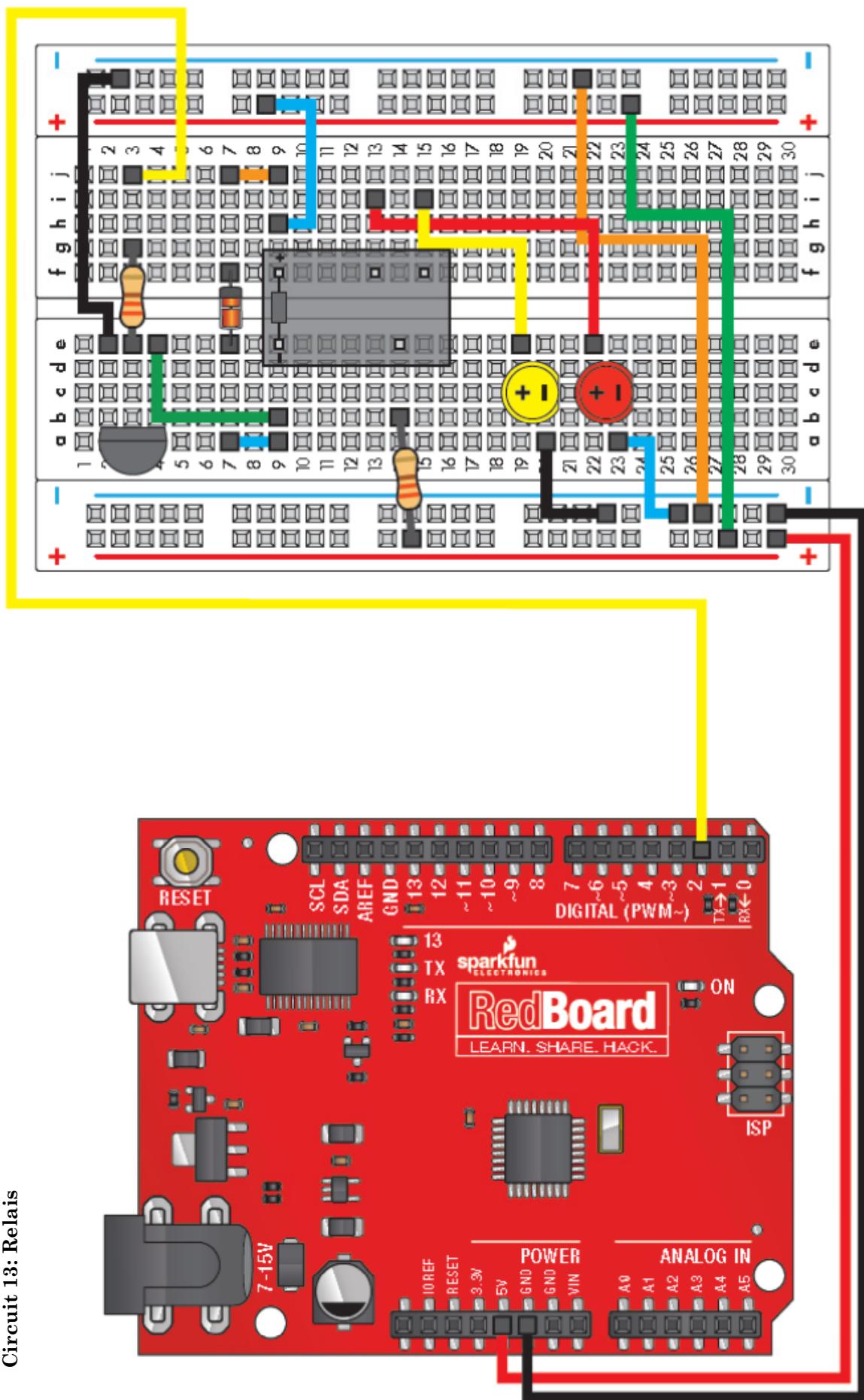


COM pin is geschakeld met NC pin als het relais uit staat. Wanneer stroom vloeit door de spoel, schakelt het relais de COM poort met de NO pin.

PARTS:	Relay	Transistor P2N2222A	Diode 1N4148	330Ω Resistor	LED	Wire
	x1	x1	x1	x1	x2	x14



Circuit 13: Relais



Component:	Afbeelding:	Component:	Afbeelding:
Relais		e14 e9 f15 f12 f9	Draadbrug
Transistor (P2N2222)		E B C a2 a3 a4	Draadbrug
LED (5 mm)		+ - c19 c20	Draadbrug
LED (5 mm)		+ - c22 c23	Draadbrug
Diode 1N4148		e7 f7	Draadbrug
330Ω weerstand		e3 g3	Draadbrug
330Ω weerstand		b14 +	Draadbrug
Draadbrug		e2 -	Draadbrug
Draadbrug		Pin 2 j3	
Draadbrug		j7 j9	
Draadbrug		h9 +	
Draadbrug		i13 e22	
Draadbrug		i15 e19	

# 13



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → [Circuit\\_13](#)



Notitie code:

**digitalWrite(relayPin, HIGH);**

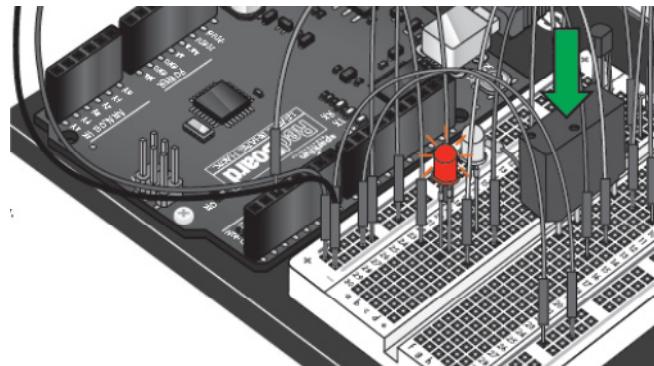
Als we de transistor inschakelen, gaat er stroom vloeien door de spoel van het relais en verandert de stand van de schakelaar. Dit verbindt de COM pin met de NO (normally open) pin. Wat met deze pinnen is verbonden, zal nu worden ingeschakeld. Hier gebruiken we LEDs, maar kan in principe van alles zijn.

**digitalWrite(relayPin, LOW);**

Het relais heeft nog een contact, NC pin (normally closed). Dit is verbonden met de COM pin als het relais niet is geactiveerd. Je kunt één van de pinnen gebruiken, afhankelijk of iets normaal aan of uit moet staan. Als beide pinnen worden gebruikt, wisselt de schakeling de stroomtoevoer naar de onderdelen, die zijn verbonden.

## Wat gebeurt er nu:

Je zou nu het klikken van het relais moeten horen samen met het alternerend aan- en uitgaan van de 2 LEDs met een interval van 1 seconde, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven.



## Tips:

### LEDs branden niet

Controleer de verbindingen. De langere pin (aan de niet-afgevlakte kant) is de positieve pin.

### Geen klikgeluid van het relais

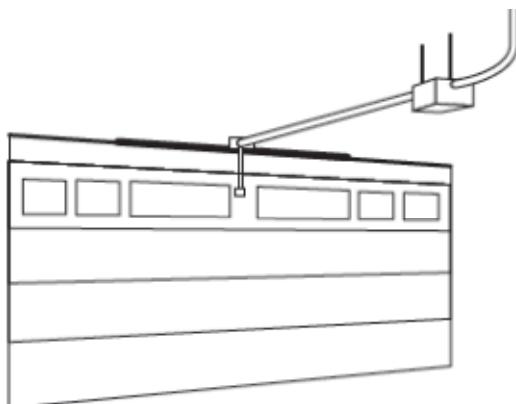
De transistor of de spoel van het relais is niet goed verbonden. Controleer of de pinnen van de transistor goed zijn aangesloten.

### Het werkt nog steeds niet

Het relais, dat hier gebruikt wordt, is bedoeld om op een print gesoldeerd te worden. In een breadboard kan dat problemen veroorzaken door de dikte van de pinnen. Het relais schiet er makkelijk uit. Druk het relais goed aan. Zorg ervoor dat je de transistor gebruikt en niet de temperatuur sensor, die dezelfde vorm heeft.

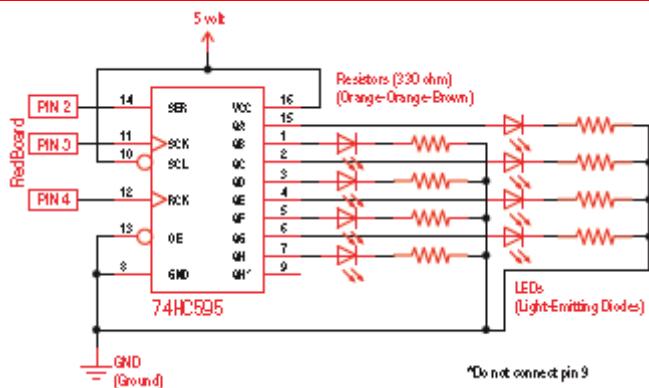
## Applicatie in de echte wereld:

Garagedeuren worden meestal met een relais bediend. Je zou zelfs het klikken van het relais kunnen horen als je goed luisterd.



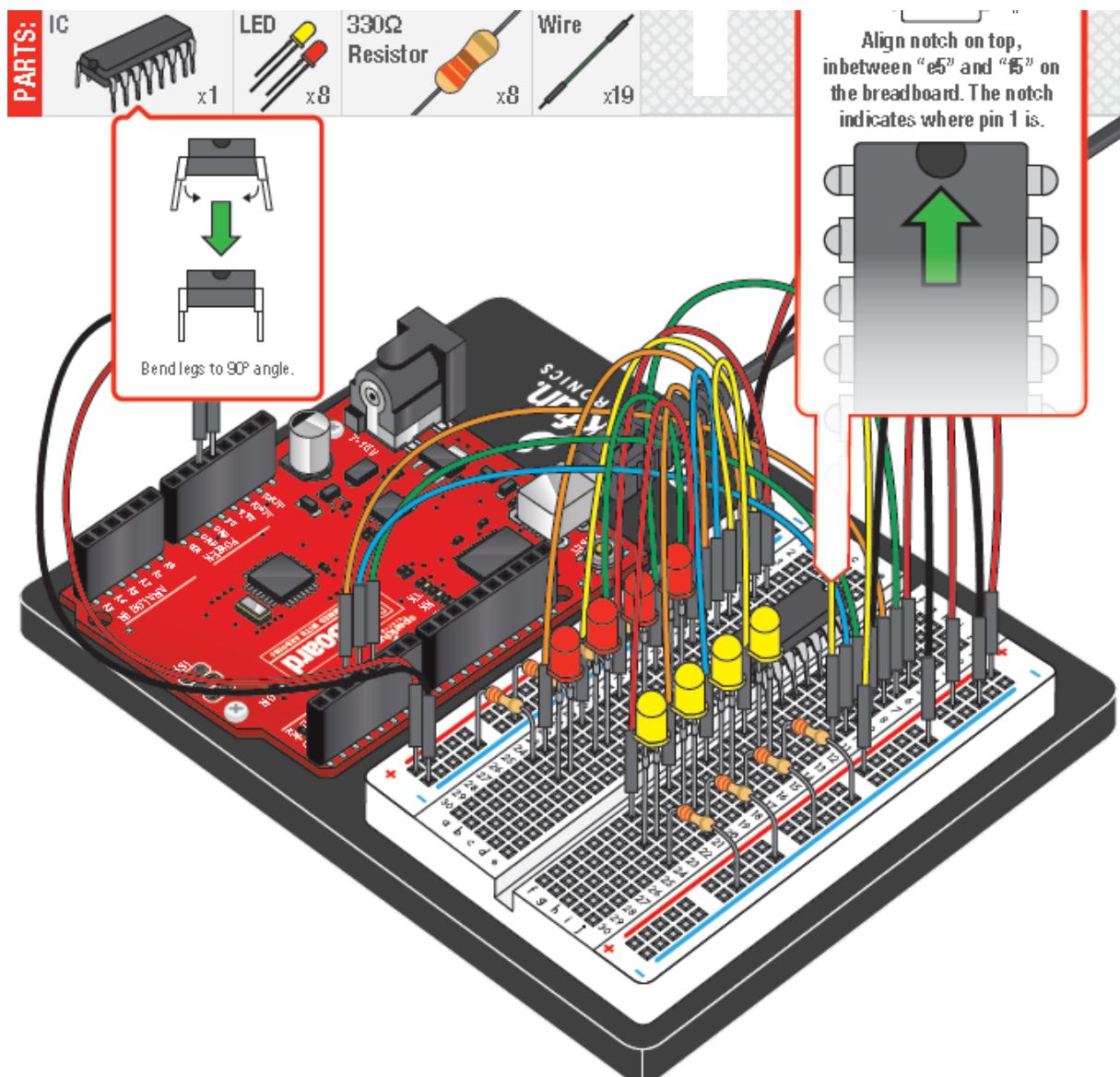
## Circuit #14

# 14

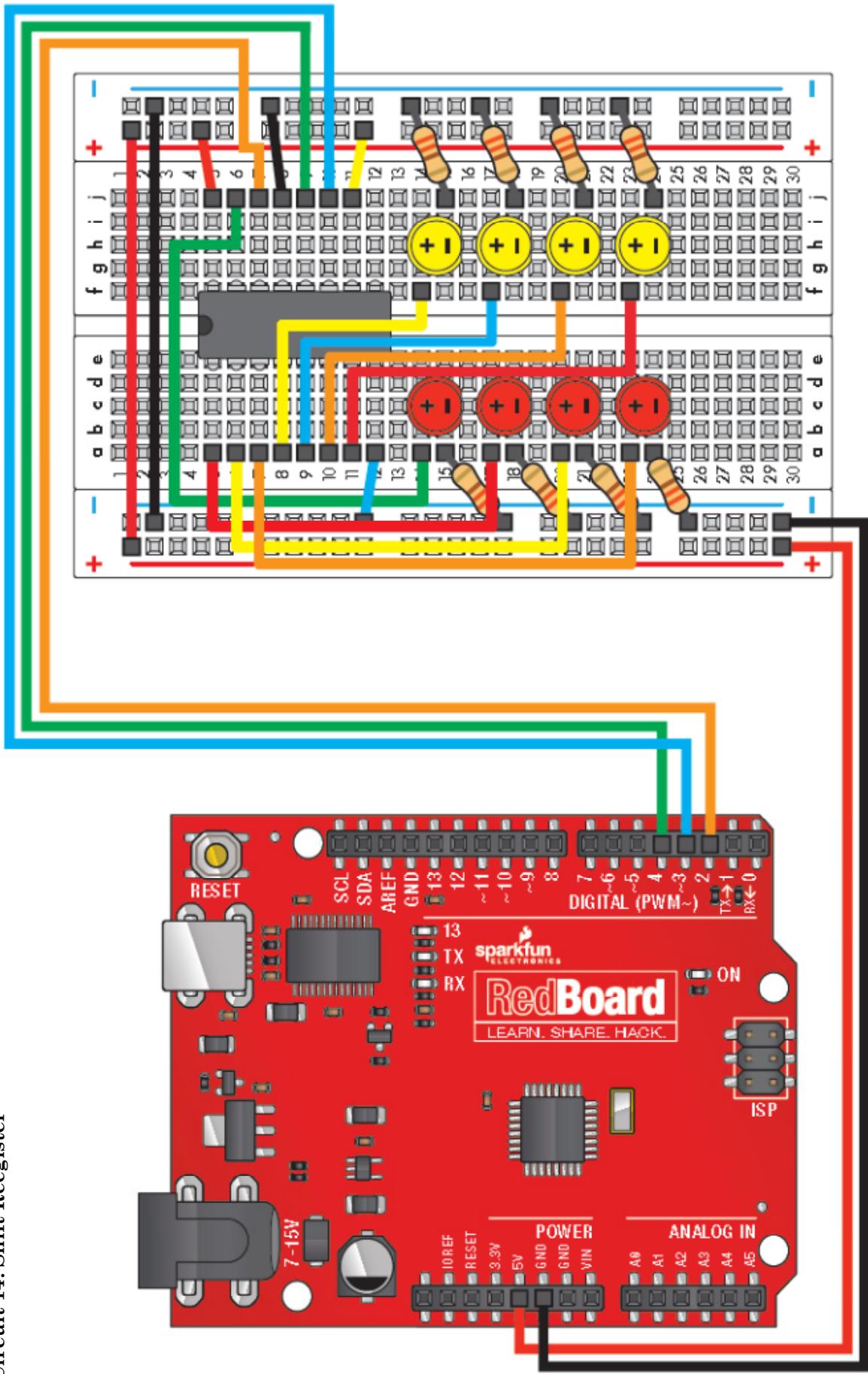


## Shift register

Nu stappen we in de wereld van de ICs (integrated circuits). In deze schakeling leer je van alles over een shift register (of serial-to-parallel converter). De shift register geeft je 8 extra outputs, terwijl er maar 3 pinnen van de Arduino worden gebruikt. De shift register wordt gebruikt om 8 LEDs te controleren.



Circuit 14: Shift Register



Component	Afbeelding	Component	Afbeelding
IC			
LED (5 mm)		Draadbrug	
LED (5 mm)		Draadbrug	
LED (5 mm)		Draadbrug	
LED (5 mm)		Draadbrug	
LED (5 mm)		Draadbrug	
LED (5 mm)		Draadbrug	
LED (5 mm)		Draadbrug	
LED (5 mm)		Draadbrug	
330Ω weerstand			
Draadbrug			



Notitie code:

**shiftOut(datapin, clockpin, MSBFIRST, data);**

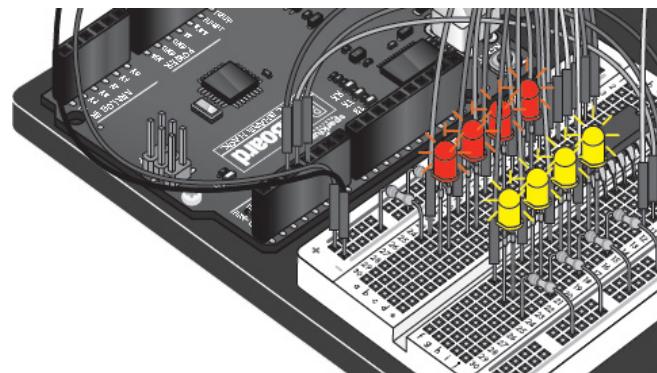
Je communiceert met de shift register (en andere onderdelen) via SPI, Serial Peripheral Interface. Dit interface werkt met een aparte clock en data pin om data met hoge snelheid van en naar het Arduino board te sturen. MSBFIRST geeft aan hoe we de data verzenden; in dit geval de Most Significant Bit eerst.

**bitWrite(byteVar, desiredBit, desiredState);**

Bits zijn de kleinste stukjes data in een computer en bevat een "0" of een "1". Grote nummers worden opgeslagen als een array van bits. Soms willen de bits rechtstreeks aansturen, bijv. door 8 bits naar het shift register te zenden met 1 en 0 om de LEDs in- of uit te schakelen. Het Arduino board heeft commando's om dat te vergemakkelijken, zoals bitWrite().

### Wat gebeurt er nu:

Je zou de LEDs moeten zien oplichten, zoals in circuit #4, maar nu maken we gebruik van een shift register, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hieronder worden beschreven.



### Tips:

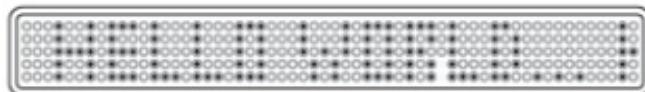
**De power LED van het board gaat uit**  
Dit kan gebeuren en meestal omdat het IC verkeerd om in het breadboard zit. Draai het IC om; er gaat niets stuk.

**Het werkt niet feilloos**  
Het lijkt wel een langspeelplaat, maar meestal heeft dat te maken met verkeerde draadverbindingen. Controleer alles nog eens.

**Gefrustreerd**  
Hoeveel de schakeling eenvoudig, maar toch complex is, kan het mis gaan. Blijf alles goed controleren. Bouw het geheel anders opnieuw op.

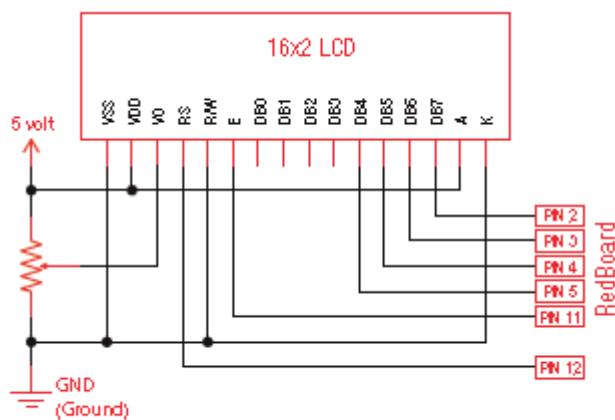
### Applicatie in de echte wereld:

Net als in circuit #4, kun je een lichtkrant maken, dat een bepaald bericht laat zien met meerdere LEDs. In principe wordt deze taak overgenomen door een shift register in dit circuit.



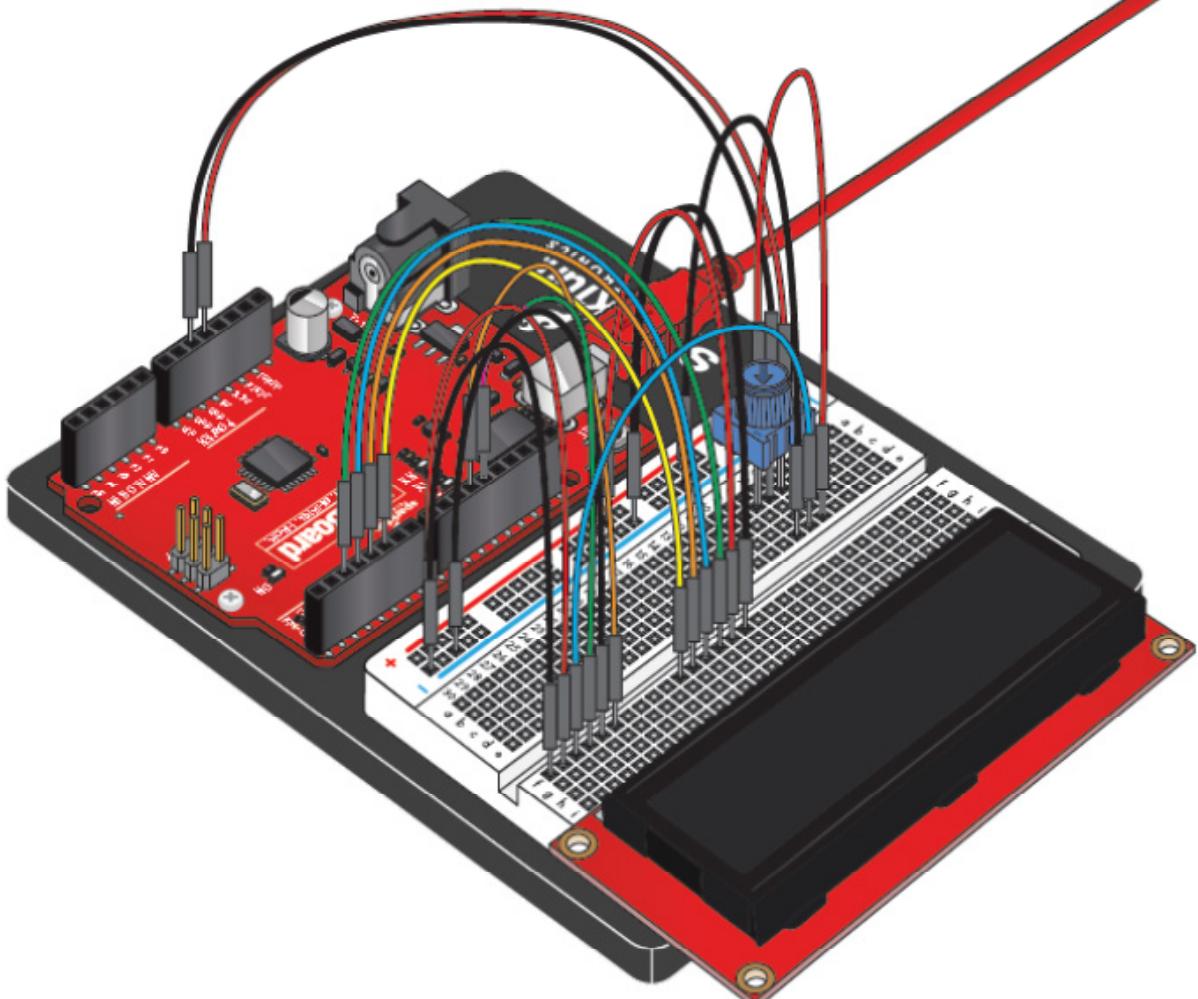
## Circuit #15

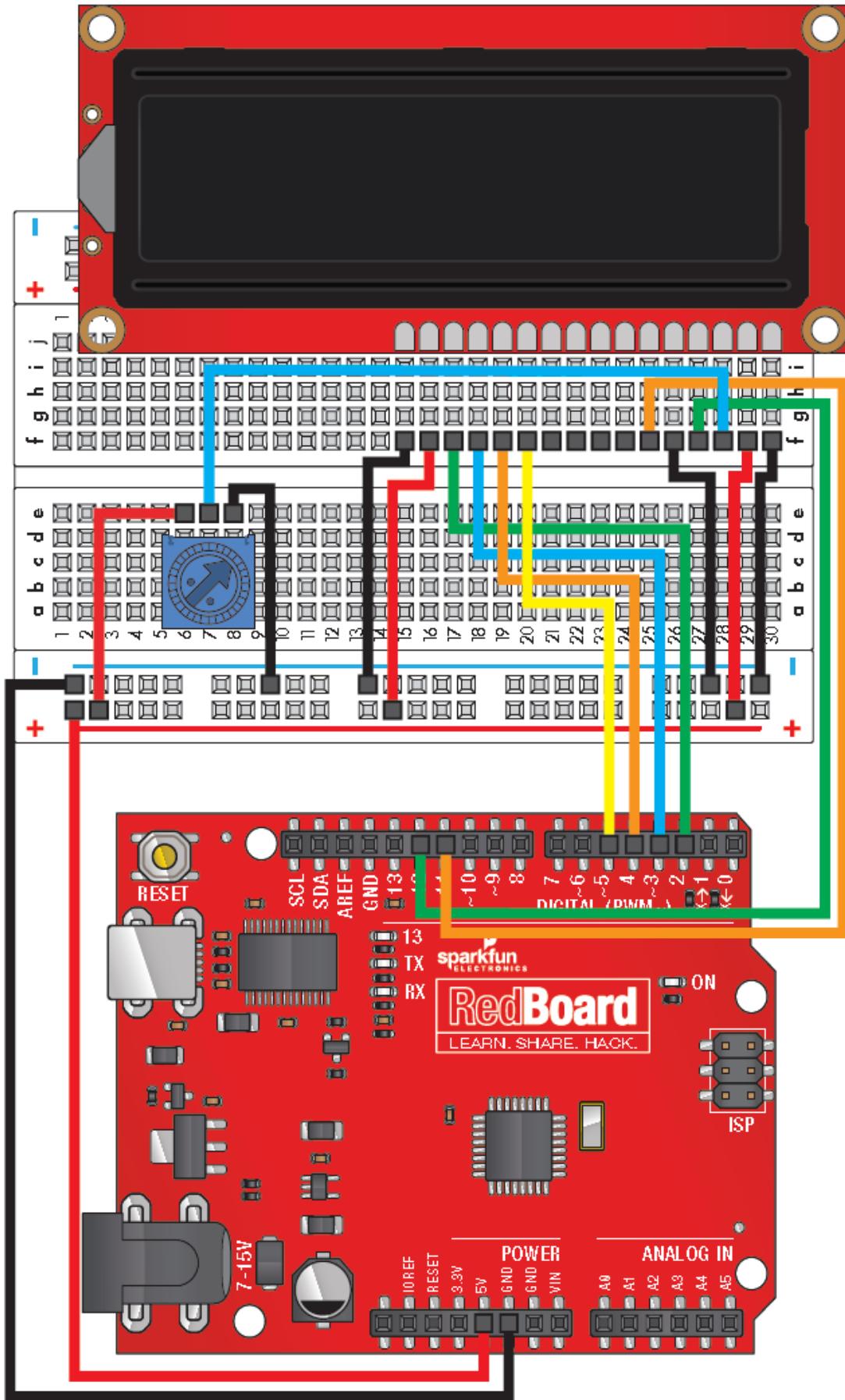
# 15



## LCD

In deze schakeling leer je om te gaan met een LCD. Een LCD (liquid crystal display), een simpel schermpje laat commando's, informatie of uitlezingen van de sensoren zien - afhankelijk van hoe de Arduino board is geprogrammeerd. Deze schakeling geeft de basis aan hoe een LCD in jouw project kan worden gebruikt. Er worden 6 pinnen van de Arduino gebruikt. Andere methodes, die minder pinnen gebruiken, zoals I2C connectie (2 pinnen) of in combinatie met een 74HC595 shift register (4 pinnen) worden hier niet behandeld.





Circuit 15: LCD

		<b>Component:</b>	<b>Afbeelding:</b>	
		Draadbrug	<b>Pin 3</b>	
		Draadbrug	<b>Pin 4</b>	
		Draadbrug	<b>Pin 5</b>	
		Draadbrug	<b>Pin 11</b>	
		Draadbrug	<b>Pin 12</b>	
		Potentiometer	<b>f18</b>	
		Draadbrug	<b>f19</b>	
		Draadbrug	<b>f20</b>	
		Draadbrug	<b>f25</b>	
		Draadbrug	<b>f26</b>	
		Draadbrug	<b>f27</b>	
		Draadbrug	<b>f29</b>	
		Draadbrug	<b>f30</b>	
		LCD	<b>5V</b>	
			<b>GND</b>	
			<b>e6</b>	
			<b>e7</b>	
			<b>e8</b>	
			<b>f15</b>	
			<b>f16</b>	
			<b>f17</b>	
			<b>Pin 2</b>	

# 15



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → [Circuit\\_15](#)



Notitie code:

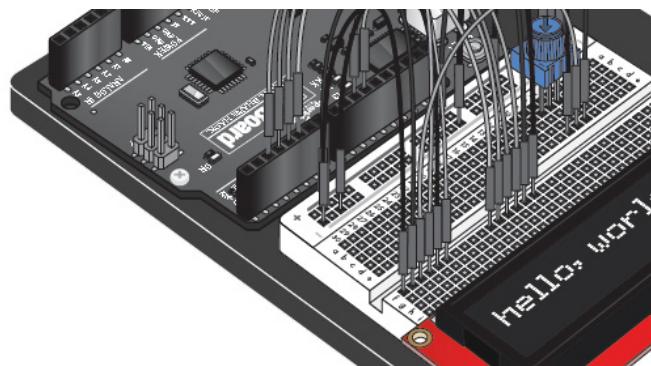
```
#include <LiquidCrystal.h>
```

```
lcd.print("hello, world!");
```

Deze code geeft aan, dat een bibliotheek om de LCD aan te sturen moet worden geladen. Zonder deze #include zullen de commando's (lcd.print) niet werken. Zorg er dus voor dat deze code bovenaan in de sketch vermeld staat.

## Wat gebeurt er nu:

Je zou nu de tekst "hello, world!" op het LCD moeten zien verschijnen, als de connecties goed zijn en de code is geverifieerd en ge-upload naar het Arduino board. Mocht dit niet het geval zijn, lees dan de tips die hier onder worden beschreven. Bedenk wel, dat het contrast van de LCD kan worden gewijzigd met de potentiometer. Een verkeerde instelling geeft geen zichtbare output of overbelichte output.



### Tips:

#### Het scherm is blanco of overbelicht

Speel met het contrast door aan de potentiometer te draaien. Als het niet goed is ingesteld, zie je niets.

#### Het werkt helemaal niet?

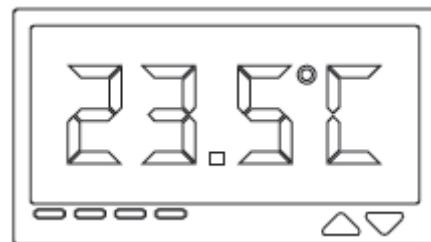
Check de sketch nog eens. Staat het gebruik van de LCD bibliotheek vermeld in jouw code.

#### Het scherm flikkert

Er zijn waarschijnlijk slechte verbindingen. Controleer de draadbruggen met de Arduino en het breadboard.

### Applicatie in de echte wereld:

LCDs zie je overal. Van geavanceerde LCDs in jouw TV tot simpele informatie schermpjes. Dit is een veel voorkomend voorbeeld.

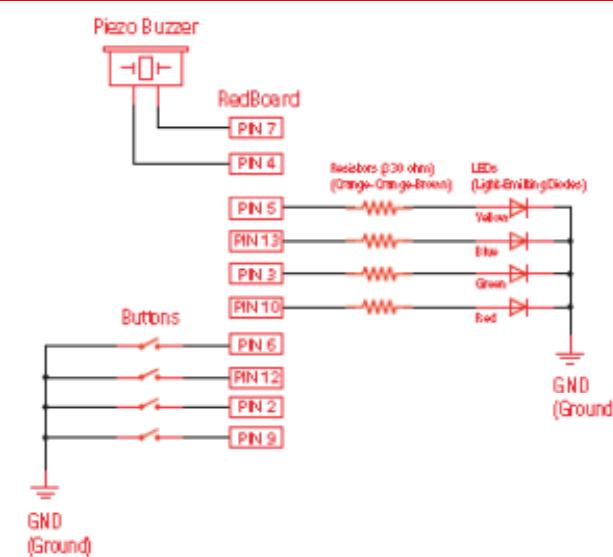


## Circuit #16

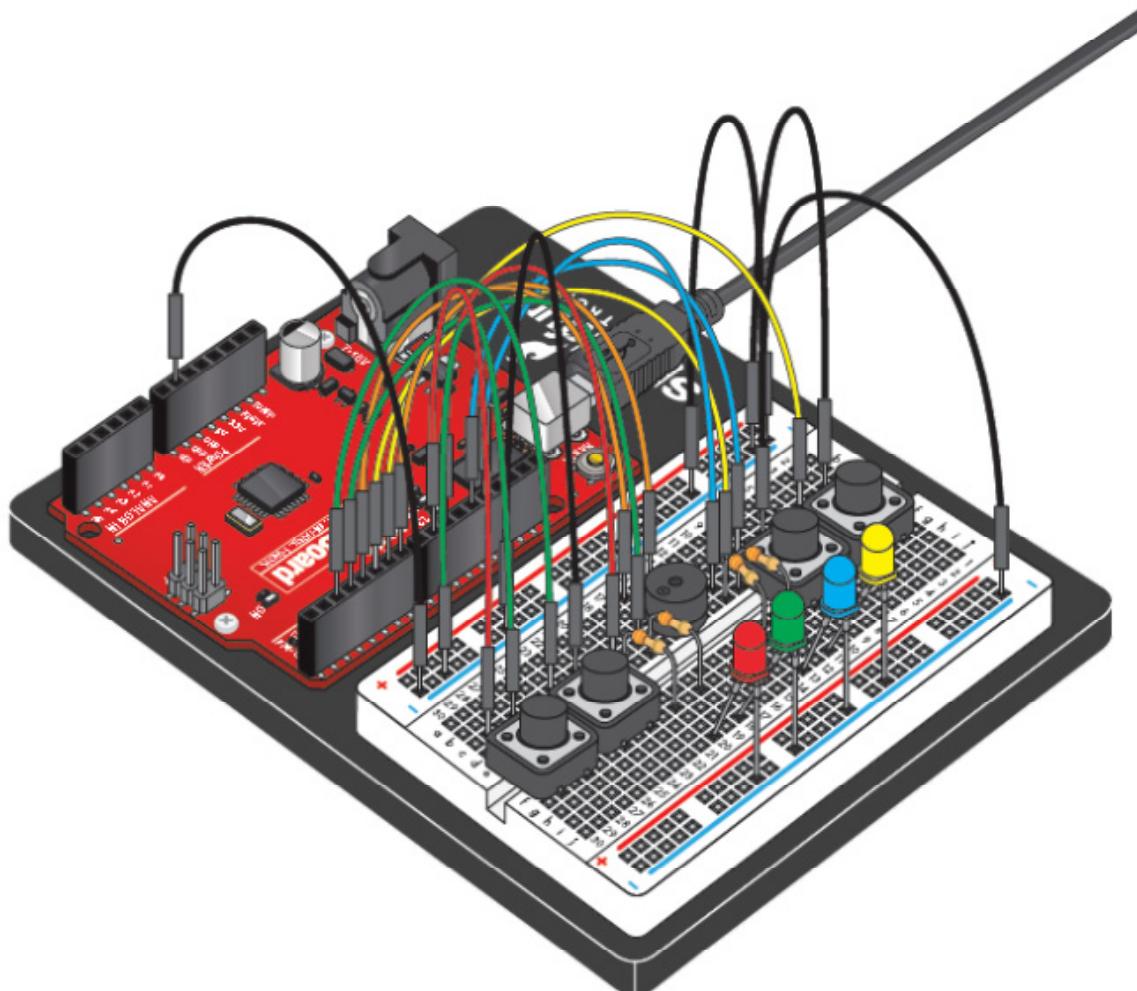
# 16

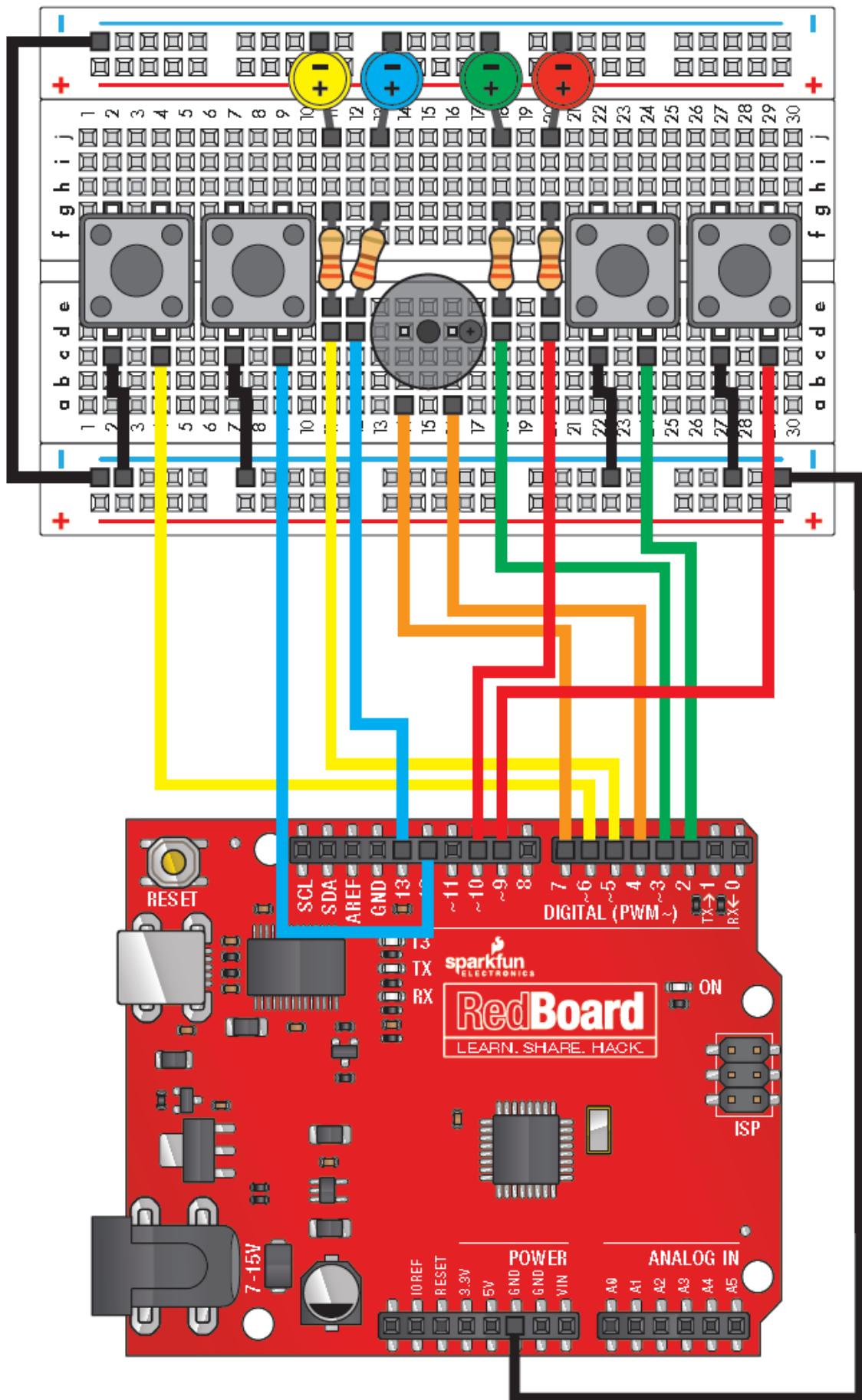
### Simon Says

Nu we de basis hebben geleerd achter alle componenten in deze ABC, laten we ze samenvoegen en er iets leuks van maken. Deze schakeling laat zien hoe jezelf een Simon Says spelletjes kunt maken. Met enkele LEDs, knoppen, buzzer en weerstanden kun je een verslavend memory-spelletje maken.



PARTS:	330Ω Resistor	LED	Push Button	Piezo Element	Wire
	x4	x4	x4	x1	x16





Circuit 16: Simon Says

Component:	Afbeelding:	Component:	Afbeelding:
330Ω weerstand		e11-g11	Draadbrug
330Ω weerstand		e12-g13	Draadbrug
330Ω weerstand		e18-g18	Draadbrug
330Ω weerstand		e20-g20	Draadbrug
LED (5 mm)		+/- J11	- Draadbrug
LED (5 mm)		+/- J12	- Draadbrug
LED (5 mm)		+/- J18	- Draadbrug
LED (5 mm)		+/- J20	- Draadbrug
Drukknop		d2-g2	Draadbrug
Drukknop		d4-g4	Draadbrug
Drukknop		d7-g7	Draadbrug
Drukknop		d9-g9	Draadbrug
Piezo buzzer		+/- a16-a14	- Draadbrug
			Draadbrug
			Draadbrug
			Draadbrug
			GND

# 16



Open Arduino IDE // Bestand → Voorbeelden → ABC-gids Code → [Circuit\\_16](#)



Notitie code:

## #define

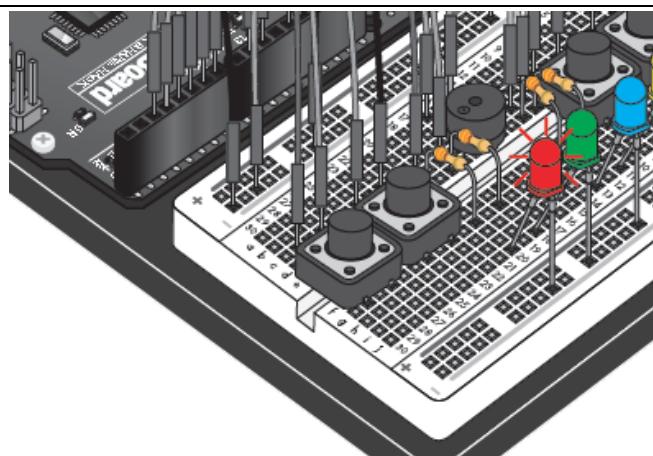
De #define opdracht wordt gebruikt om constante in de code te definiëren. Constanten zijn variabelen, die waarschijnlijk maar één waarde hebben in de code. Je kunt een waarde toewijzen (nummers, letters), die je dan later in de code kunt gebruiken. Als je iets wilt veranderen, hoef je alleen nog maar in één regel de constante te veranderen zonder de hele code te moeten doorlopen.

## byte

Byte is een andere type variabele. In de computer is een byte de ruimte waarin 8 bits zitten, en een bit is een enkele binaire waarde. Binair is een andere manier van tellen en kent alleen enen en nullen. Een byte kan allemaal enen bevatten: 11111111 of allemaal nullen: 00000000 of een combinatie: 10010110.

## Wat gebeurt er nu:

Als de schakeling klaar is, koppel de Arduino dan aan een voedingsbron. Er klinkt dan een geluidje en alle 4 LEDs knipperen. Het spel begint als één van de knoppen wordt ingedrukt. Een willekeurige LED gaat aan. Druk op de bijbehorende knop. Als je het goed hebt, gaat die LED weer aan. Nu wordt een nieuwe willekeurige LED ingeschakeld. Herhaal de gehele reeks door de bijbehorende knopjes in te drukken. Dit gaat door totdat je een fout maakt. Probeer een zo lang mogelijke reeks te onthouden en via de knopjes in te voeren. Als je het fout hebt, begint het spel opnieuw.



## Tips:

### Het halve circuit werkt

Als maar de helft werkt, zorg er dan voor dat aan die zijde een verbinding met GND is. Breadboards hebben 2 voedingsrails, die verbonden kunnen worden of doorgelust. Check de voeding en de connecties

### Geen geluid

Een buzzer op de breadboard is moeilijk te verbinden, omdat de pinnen niet zichtbaar zijn. Daardoor is de buzzer misschien niet verbonden. Check de draadbruggen aan de buzzer.

### Het spel werkt niet

Als alles goed opstart, maar je kunt het spel niet spelen dan ligt dat waarschijnlijk aan een verkeerd geplaatste knop. Kijk of de knoppen goed zijn geplaatst en welke pinnen zijn verbonden..

## Applicatie in de echte wereld:

Speelgoed en spelletjes, zoals dit Simon Says spelletje vertrouwen veelal op electronica en geeft veel plezier aan kinderen en anderen, die het leuk vinden om spelletjes te spelen.

