



Les 1: Blink
Les 2: Blink Blink Blink
Les 3: Knop if else
Les 4: Knop flip

Figure 1: Boek 1: LED en knop

Contents

Voorwoord	1
Les 1: Blink	2
Les 2: Blink Blink Blink	11
Les 3: Knop if . . . else	17
Les 4: Knop flip	24

Voorwoord

Dit is een boek over Arduino, geschreven voor jonge tieners. Een Arduino is een machine die je kunt programmeren. Dit boek leert je hoe je elektronica op de Arduino aansluit, en hoe je deze programmeert.

Over dit boek

Dit boek heeft een CC-BY-NC-SA licentie.



Figure 1: De licentie van dit boek

(C) Richèl Bilderbeek en alle docenten en alle leerlingen

Met dit boekje mag je alles doen wat je wilt, als je maar verwijst naar de oorspronkelijke versie op deze website: https://github.com/richelbilderbeek/arduino_voor_jonge_tieners. Dit boekje zal altijd gratis, vrij en open blijven.

Het is nog een beetje een slordig boek. Er zitten tiepvauten in en de opmaak is **niet altijd even mooi**. Omdat dit boek op een website staat, kan iedereen die dit boek te slordig vindt minder slordig maken.

Les 1: Blink

In deze les gaan we de allergemakkelijkste schakeling maken die er is: Blink!



We beginnen makkelijk!

1.1 Blink: een Arduino aansluiten

Sluit een Arduino zo aan:

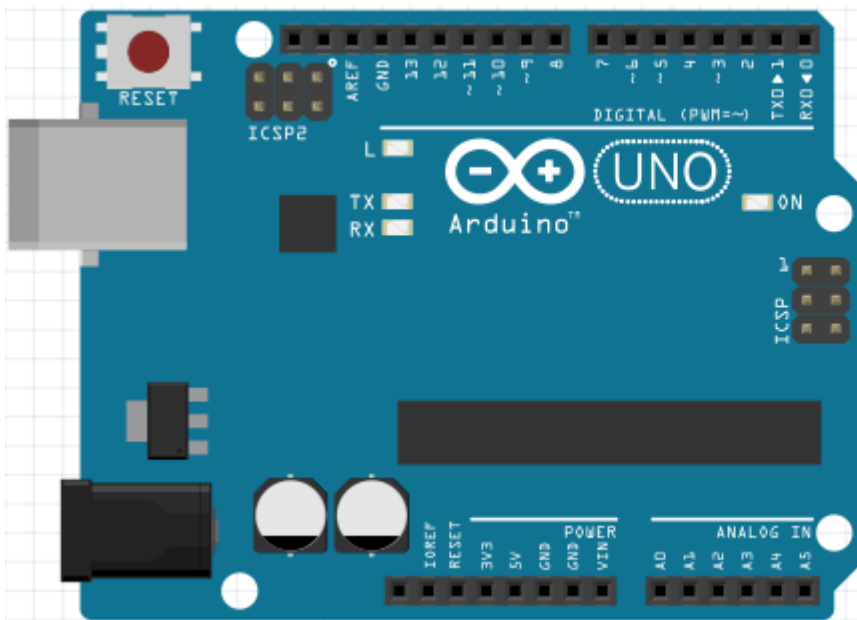


Figure 2: Blink zonder led

Doe een USB kabel in je Arduino en in je laptop.

1.2 Blink: opdracht 1

Sluit de Arduino aan.



De Arduino heeft zelf al een lampje dat je kunt programmeren.

1.3 Blink: de Arduino IDE opstarten



We programmeren de Arduino met de Arduino IDE



‘IDE’ spreek je uit als ‘ie-dee-ee’

Start de Arduino IDE door:

- Klik op de snelkoppeling op het Bureaublad
- Druk op de Windows toets (linksonder, tussen **Ctrl** en **Alt**). Type dan **arduino** (kleine letters) en dan Enter

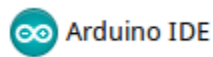


Figure 3: Logo van de Arduino IDE



Figure 4: Win toets

Je ziet nu je de Arduino IDE:

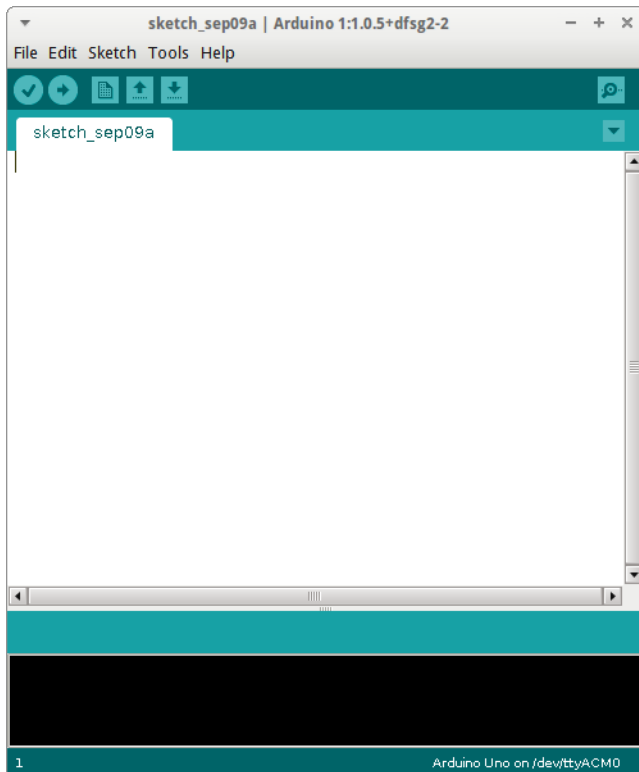


Figure 5: De Arduino IDE



IDE betekent 'Integrated Development Environment', het programma waarmee je programmeert.

1.4 Blink: ons eerste programma

Dit is (een versie van) de code van Blink:

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

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



`void setup() { }`



‘Lieve computer, doe wat tussen accolades staat een keer.’



`void loop() { }`



‘Lieve computer, doe de hele tijd wat tussen accolades staat.’

Type de code over in de Arduino IDE en klik op Upload.



Figure 6: Hier kun je op Upload klikken

1.5 Blink: opdracht 2

Laat het LEDje 10x zo snel knipperen. Hint: het LEDje is nu 1000 milliseconden aan en 1000 milliseconden uit.

1.6 Blink: oplossing 2

```
void setup()
{
  // ...
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(100);
  digitalWrite(13, LOW);
  delay(100);
}
```



// ... betekent 'De code die je daar al hebt staan'



pinMode(13, OUTPUT); 'Zorg dat er spanning uit pin 13 kan komen'



digitalWrite(13, HIGH); 'Zet spanning op pin 13'



delay(1000); 'Wacht 1000 milliseconden'

1.7 Blink: opdracht 3

Laat het LEDje zo snel als kan knipperen. Wat zie je?

1.8 Blink: oplossing 3

Er zijn meer oplossingen, dit is er een:

```
void setup()
{
  // ...
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1);
  digitalWrite(13, LOW);
  delay(1);
}
```

Je ziet het lampje niet meer knipperen.



Je ziet het lampje niet meer knipperen!



Het lampje knippert wel, maar onze ogen kunnen niet snel genoeg kijken om het te zien.



`digitalWrite(13, LOW);` ‘Haal de spanning van pin 13 af’



Ook goed: `delay(0);` of de regels met `delay` helemaal weghalen.

1.9 Blink: LED aansluiten

Nu is het tijd Blink aan te sluiten:

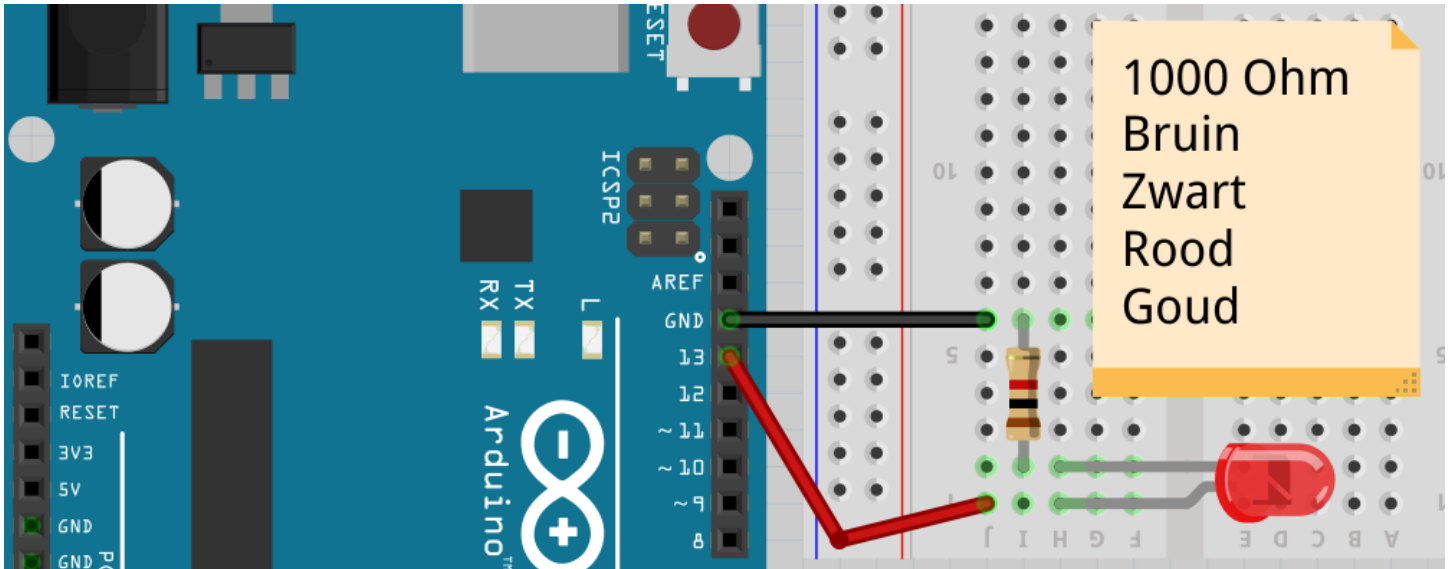


Figure 7: Blink

- Haal het USB snoer uit de computer, zodat de Arduino geen spanning meer heeft
- Sluit Blink aan zoals op de tekening
- Zet de spanning weer op de Arduino
- Als het goed is, knippert het rode LEDje nu mee met het LEDje op de Arduino.



Als het LEDje niet brandt, draai deze dan om.

1.10 Blink: opdracht 4

Sluit je LEDje nu aan op pin 12.



Het is cool om na pin 13 pin 12 te pakken



Dit is ook omdat pinnen 0 en 1 speciaal zijn

1.11 Blink: oplossing 4

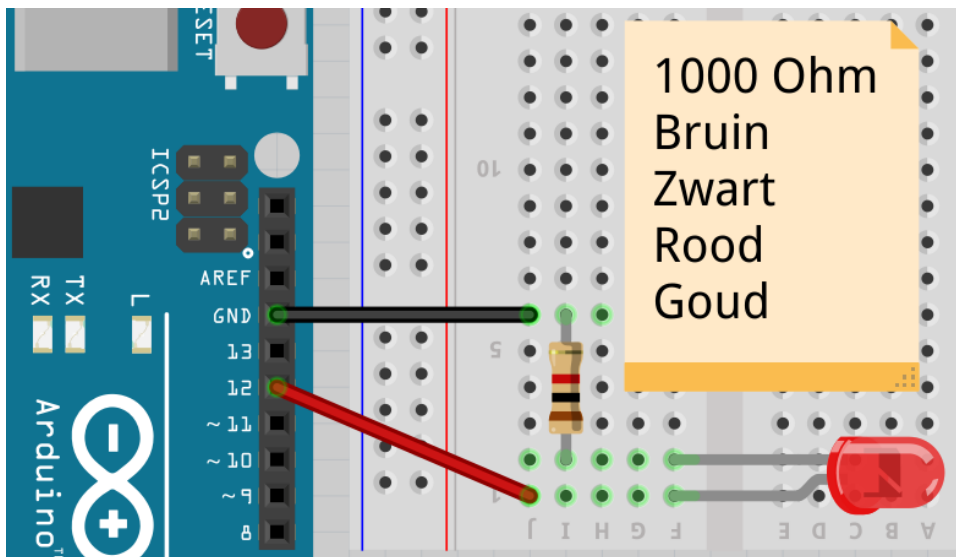


Figure 8: Blink met LEDje op pin 12

1.12 Blink: opdracht 5

Zorg nu dat het LEDje op pin 12 gaat knipperen op de seconde.



LED is een afkorting voor 'Light Emitting Diode'



Goed gezegd! Beter te laat dan nooit!



LED is in het Nederlands: 'Diode waar licht uit komt'

1.13 Blink: oplossing 5

```
void setup()
{
  pinMode(12, OUTPUT);
}

void loop()
{
  digitalWrite(12, HIGH);
  delay(100);
  digitalWrite(12, LOW);
  delay(100);
}
```

1.14 Blink: eindopdracht

Sluit de LED aan op pin 11 en laat deze 2 keer per seconde knipperen.



Gelukt? Laat dit zien aan een volwassene voor een paraaf!

Les 2: Blink Blink Blink

Deze les heet 'Blink Blink Blink', omdat het de les Blink is, maar dan met drie lampjes.

2.1: Blink Blink Blink: Blink

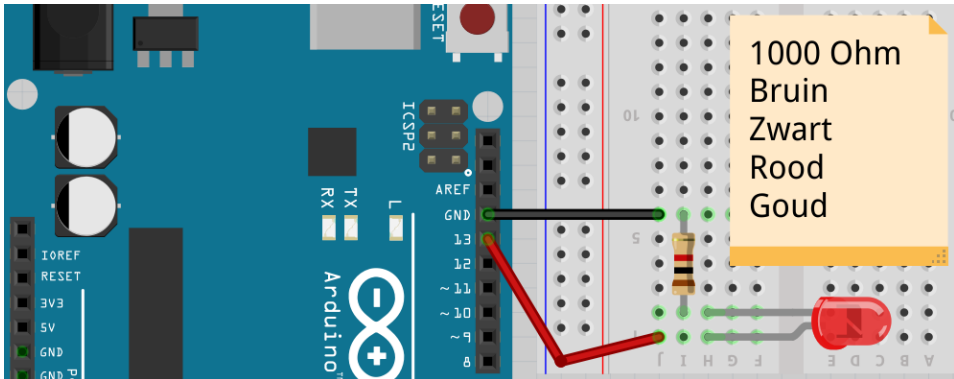


Figure 9: Blink

Dit is de code van Blink op een andere manier:

```
const int pin_led = 13;

void setup()
{
  pinMode(pin_led, OUTPUT);
}

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

`pin_led` wordt een variabele genoemd: een stukje computergeheugen met een naam.



```
const int pin_led = 13;
```

‘Lieve computer, onthoud een heel getal met de naam `pin_led` met beginwaarde 13.’

2.2: Blink Blink Blink: Opdracht 1

Sluit de LED aan op pin 12 en verander de code zodat deze gaat knipperen.

2.3: Blink Blink Blink: Oplossing 1

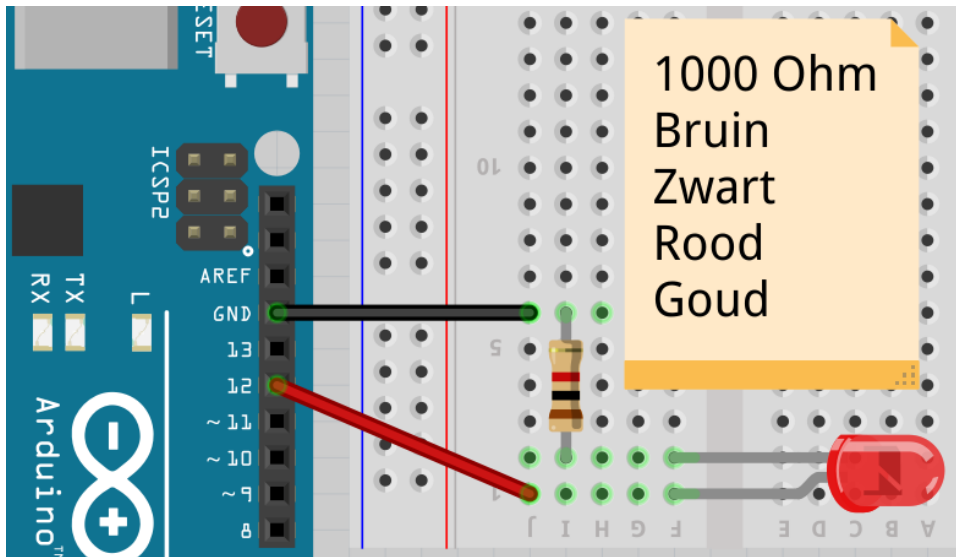


Figure 10: Blink met LEDje op 12

Je hoeft maar een regel te veranderen:

```
const int pin_led = 12;

void setup()
{
  // ...
}

void loop()
{
  // ...
}
```

2.4: Blink Blink Blink: Opdracht 2

Maak zelf een nieuwe variabele met de naam `wachttijd`. `wachttijd` is een heel getal met beginwaarde 1000. Gebruik `wachttijd` in de regels met `delay`.



Slim! Als je `wachttijd` leest, weet je waar het voor is. Bij 1000 weet je dat niet

2.5: Blink Blink Blink: Oplossing 2

```
// ... [maak pin_led met waarde 12]
const int wachttijd = 1000;

void setup()
{
  // ...
}

void loop()
{
  // ... [zet spanning op pin_led]
  delay(wachttijd);
  // ... [geen spanning op pin_led af]
  delay(wachttijd);
}
```



// ... [wat] betekent 'De code die je daar al hebt staan die gaat over wat er tussen blokhaken staat'

2.6: Blink Blink Blink: Blink Blink Blink aansluiten

Nu is het tijd 'Blink Blink Blink' aan te sluiten:

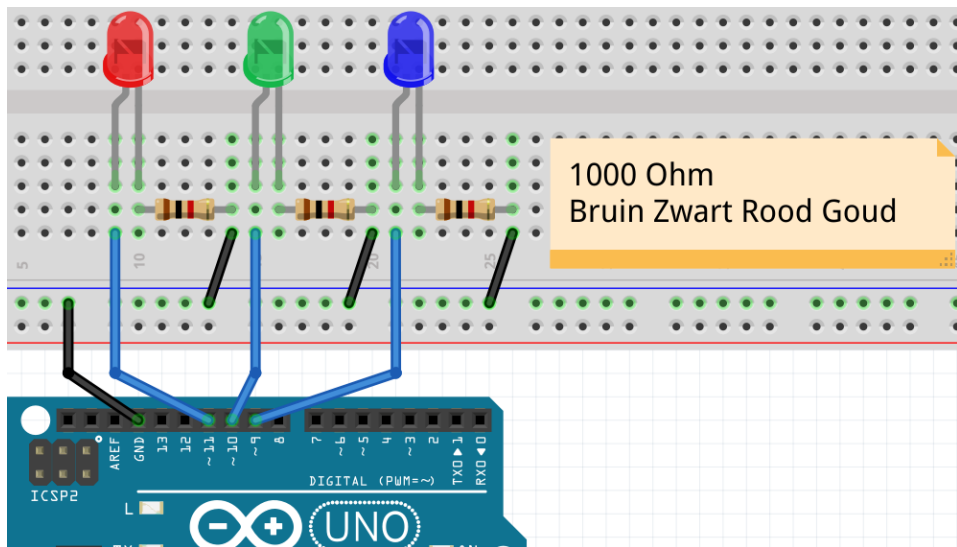


Figure 11: BlinkBlinkBlink

- Haal het USB snoer uit de computer, zodat de Arduino geen spanning meer heeft
- Sluit de onderdelen aan zoals op de tekening

2.7: Blink Blink Blink: Opdracht 3

Sluit 'Blink Blink Blink' aan. Hernoem de variable `pin_led` naar `pin_led_1` en zorg dat 'ie de juiste beginwaarde heeft.

Oplossing 3

```
const int pin_led_1 = 11;
// ... [onthoud wachttijd]

void setup()
{
  pinMode(pin_led_1, OUTPUT);
}

void loop()
{
  digitalWrite(pin_led_1, HIGH);
  // [wacht wachttijd milliseconden]
  digitalWrite(pin_led_1, LOW);
  // [wacht wachttijd milliseconden]
}
```



Programmeurs gebruiken veel variabelen, omdat de code dan beter te snappen is.

2.8: Blink Blink Blink: Opdracht 4

Maak een nieuwe variabele `pin_led_2`. Laat dan eerst het eerste LEDje aan en uit gaan, laat dan het tweede LEDje aan en uit gaan.



De schrijfwijze `pin_led_2` wordt 'snake case' genoemd: 'snake' is Engels voor slang



Zou je schrijven `pinLed2`, dan wordt dat 'camel case' genoemd: 'camel' is Engels voor kameel



Mij maakt het niet uit welke je kiest

2.9: Blink Blink Blink: Oplossing 4

```
const int pin_led_1 = 11;
const int pin_led_2 = 10;
// ...

void setup()
{
  pinMode(pin_led_1, OUTPUT);
  pinMode(pin_led_2, OUTPUT);
}

void loop()
{
  digitalWrite(pin_led_1, HIGH);
  // ... [wacht wachttijd milliseconden]
  digitalWrite(pin_led_1, LOW);
  // ... [wacht wachttijd milliseconden]
  digitalWrite(pin_led_2, HIGH);
  // ... [wacht wachttijd milliseconden]
  digitalWrite(pin_led_2, LOW);
  // ... [wacht wachttijd milliseconden]
}
```

2.10: Blink Blink Blink: Opdracht 5

Maak een derde variabele `pin_led_3`. Laat nu steeds alle lampjes tegelijk knipperen: allemaal aan, dan allemaal uit.



Dit is best veel typewerk! Later leer je hoe dit slimmer kan



Klopt! Met arrays kun je meerdere getallen in een variabele opslaan

2.11: Blink Blink Blink: Oplossing 5

```
// ... [maak wachttijd, pin_led_1 en pin_led_2]
const int pin_led_3 = 9;

void setup()
{
  // ... [pin_led_1 en pin_led_2 geven spanning]
  pinMode(pin_led_3, OUTPUT);
}

void loop()
{
  // ... [zet spanning op LED 1 en 2]
  digitalWrite(pin_led_3, HIGH);
  // ... [wacht wachttijd milliseconden]
  // ... [zet spanning op LED 1 en 2]
  digitalWrite(pin_led_3, LOW);
  // ... [wacht wachttijd milliseconden]
}
```

2.12: Blink Blink Blink: Eindopdracht

Laat de lampjes nu in een ‘Knight Rider patroon’ gaan: 1-2-3-2. Er moet altijd precies een lampje branden.



Knight Rider was een TV serie met een pratende auto.

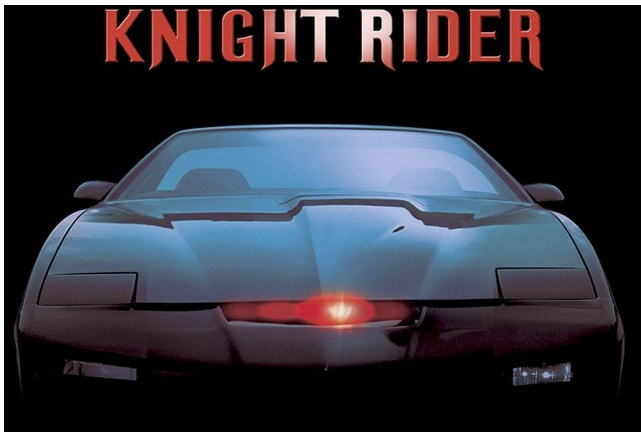


Figure 12: Knight Rider

In deze les laten we de Arduino vragen aan zichzelf stellen.



De Arduino wordt dus een soort filosoof!

3.1. Knop if ... else: Aansluiten

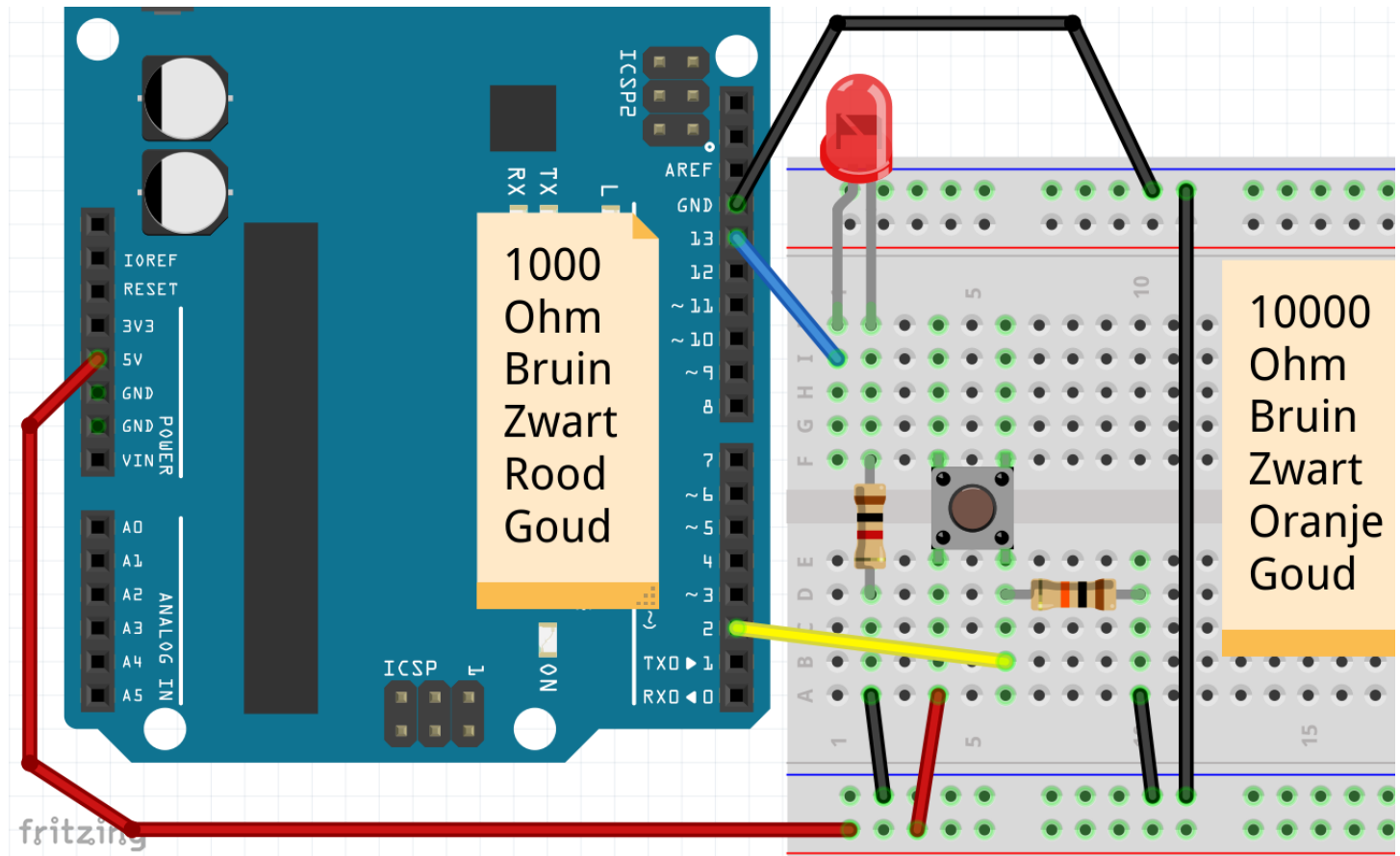


Figure 13: Stroomschema



De weerstand aan de knop wordt een ‘Pull Down’ weerstand genoemd

3.2. Knop if ... else: Code

Hier is code om het lampje aan te laten gaan, als de knop wordt ingedrukt:

```
const int pin_led = 13;
const int pin_knop = 2;

void setup()
{
  pinMode(pin_led, OUTPUT);
  pinMode(pin_knop, INPUT);
}

void loop()
{
  digitalWrite(pin_led, digitalRead(pin_knop));
}
```



`digitalRead(pin_knop)`

‘Geef mij HIGH als er op `pin_knop` spanning staat. Geef anders LOW’

`digitalWrite(pin_led, digitalRead(pin_knop))`

‘Zet op `pin_led` spanning als er op `pin_knop` spanning staat’



De ‘Pull Down’ weerstand zorgt dat pin 2 verbonden is met GND als de knop niet ingedrukt is

3.3. Knop if ... else: Vragen

Als je de knop indrukt, gaat de LED dan uit of aan?

3.4. Knop if ... else: Antwoorden

De LED gaat dan aan



De LED gaat dan aan

3.5. Knop if ... else: if

Met if kun je de Arduino iets laten doen, als iets zo is:

```
if (digitalRead(pin_knop) == HIGH)
{
    digitalWrite(pin_led_rood, HIGH);
}
else
{
    digitalWrite(pin_led_groen, LOW);
}
```

In deze code wordt gekeken of op `pin_knop` spanning staat. Zo ja, dan zet de Arduino spanning op `pin_led_rood`. Anders (else) haalt de Arduino de spanning van `pin_led_groen` af.



```
if (digitalRead(pin_knop) == HIGH) {}
```

‘Lieve computer, als er spanning op `pin_knop` staat, doe dan dat tussen accolades’



De = kun je uitspreken als ‘zet op’. De == kun je uitspreken als ‘is gelijk aan’

3.6. Knop if ... else: Opdracht 1

Dit is voorbeeld code, die niet helemaal af is:

```
void setup()
{
  pinMode(4, INPUT );
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}

void loop()
{
  if (digitalRead(4) == HIGH)
  {
    digitalWrite(5, HIGH);
  }
  else
  {
    digitalWrite(6, LOW);
  }
}
```

Maak de code af, zodat:

- als je op de knop drukt, het lampje aan gaat
- als je op de knop niet indrukt, het lampje uit gaat
- gebruik variabelen om de pinnen een naam te geven



Na de ronde haken van `if` komt geen puntkomma

3.7. Knop if ... else: Oplossing 1

```
// ...  
  
void setup()  
{  
  // ...  
}  
  
void loop()  
{  
  if (digitalRead(pin_knop) == HIGH)  
  {  
    digitalWrite(pin_led, HIGH);  
  }  
  else  
  {  
    digitalWrite(pin_led, LOW);  
  }  
}
```



// ... betekent 'De code die je daar al hebt staan'

3.8. Knop if ... else: Opdracht 2

Sluit een tweede, groen LEDje aan. Maak de code zo, dat:

- als je op de knop drukt, dat het eerste lampje aan gaat en het tweede LEDje uit
- als je op de knop niet indrukt, dat het eerste lampje uit gaat en het tweede LEDje uit
- Noem de variabele voor de pin van de rode LED `pin_led_rood`, voor de groene LED `pin_led_groen`



Binnen de accolades van een `if` kun je meerdere regels typen. Net als tussen de accolades van `setup` en `loop`!



Vergeet de `else` ('doe anders') niet!

3.9. Knop if ... else: Oplossing 2

Figuur Oplossing van 'Knop met twee LEDjes' laat zien hoe je dit aan moet sluiten.

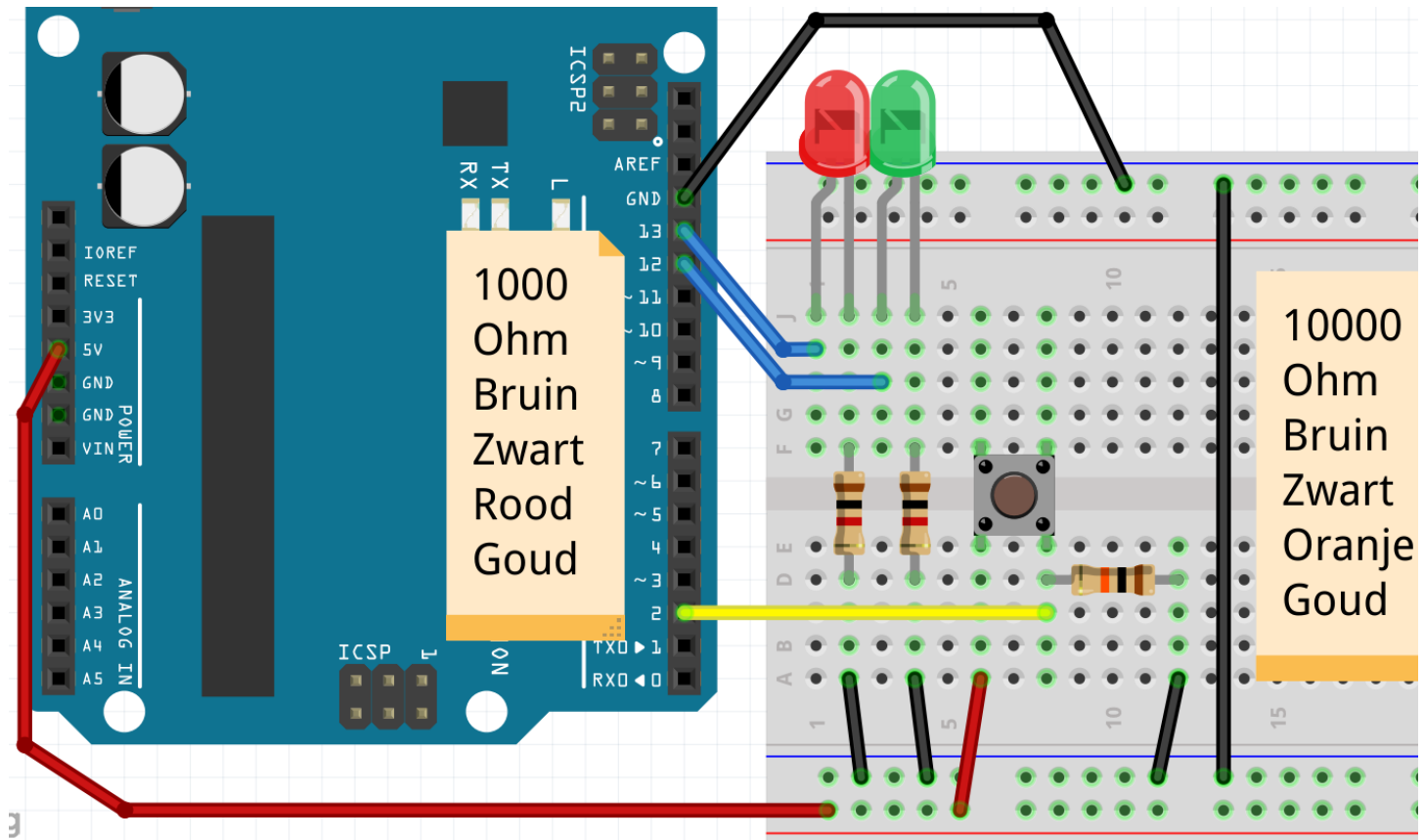


Figure 14: Oplossing van 'Knop met twee LEDjes'

Dit is de code:

```
const int pin_led_rood = 13;
const int pin_led_groen = 12;
// ...

void setup()
{
  pinMode(pin_led_rood, OUTPUT);
  pinMode(pin_led_groen, OUTPUT);
  // ...
}

void loop()
{
  if (/* ... */)
  {
    digitalWrite(pin_led_groen, LOW);
    digitalWrite(pin_led_rood, HIGH);
  }
  else
  {
    digitalWrite(pin_led_groen, HIGH);
    digitalWrite(pin_led_rood, LOW);
  }
  delay(10);
}
```



`/* ... */` betekent ook 'De code die je daar al hebt staan'

3.10. Knop if ... else: Eindopdracht

Sluit een tweede knop aan. Maak de code zo, dat

- als je op de eerste knop drukt, dat het eerste lampje aan gaat
- als je de eerste knop niet indrukt, dat het eerste lampje uit gaat
- als je op de tweede knop drukt, dat het tweede lampje uit gaat
- als je de tweede knop niet indrukt, dat het tweede lampje aan gaat



Je kunt vaker `if` na elkaar zetten



Voor een tweede knop heb je een tweede weerstand van tienduizend Ohm nodig

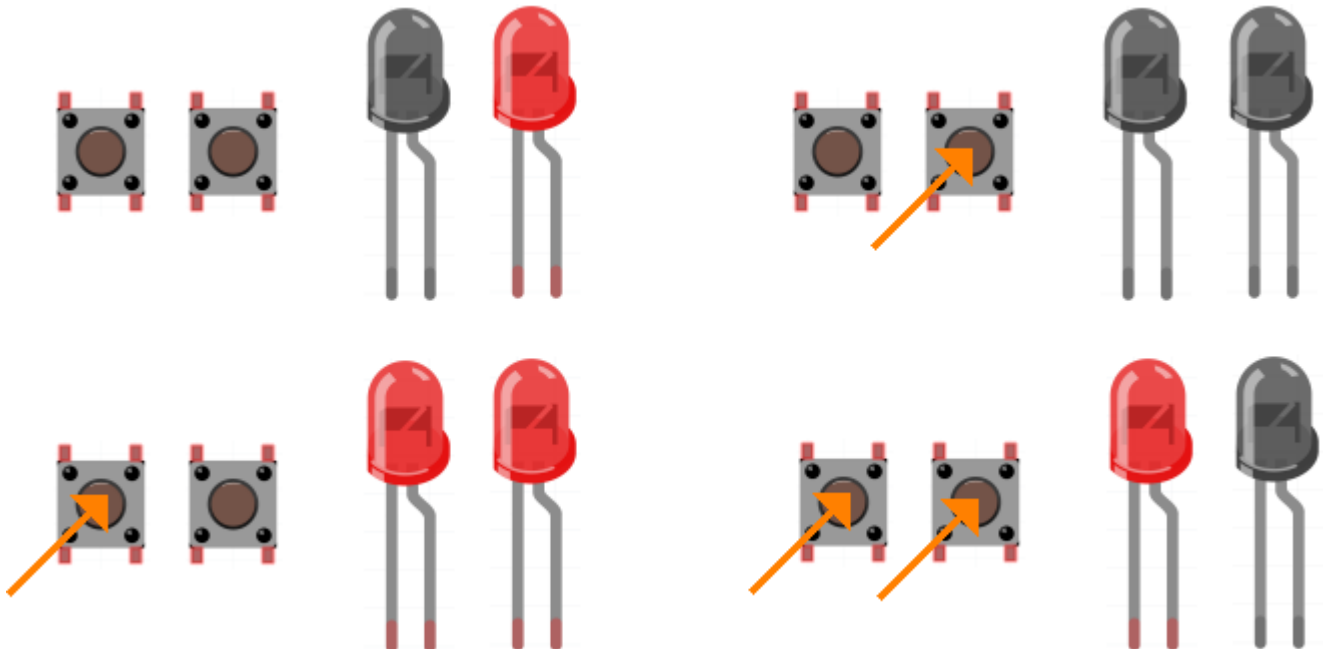


Figure 15: Eindopdracht

Les 4: Knop flip

In deze les leren we een variabele te gebruiken.



De les heet 'knop flip' omdat je met een flip-flop programmeert

4.1. Knop flip: Intro

We beginnen met deze schakeling:

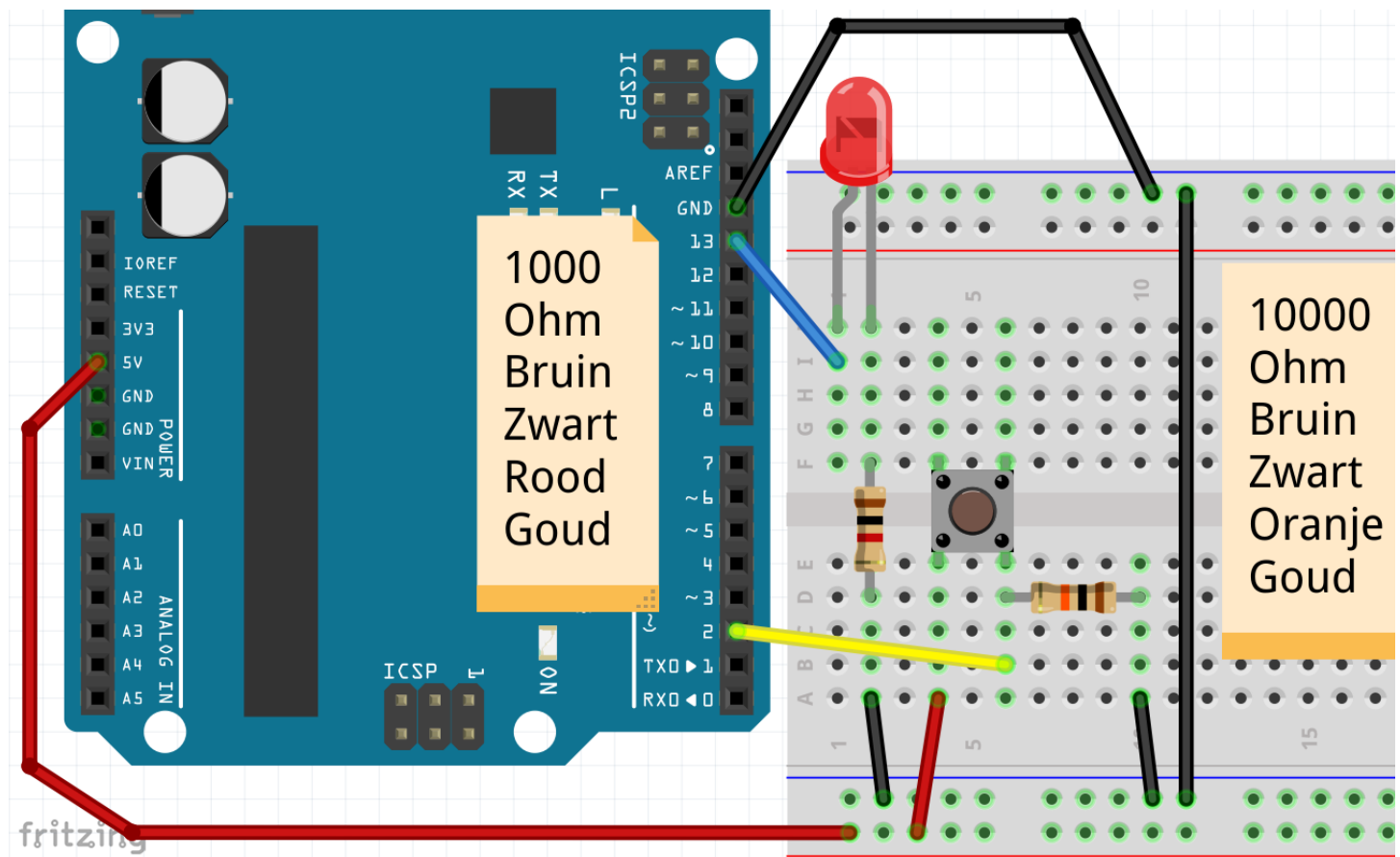


Figure 16: Stroomschema



Hee, deze schakeling herken ik!

En we gebruiken deze code:

```
const int pin_led = 13;
const int pin_knop = 2;

void setup()
{
  pinMode(pin_led, OUTPUT);
  pinMode(pin_knop, INPUT);
}

void loop()
{
  if (digitalRead(pin_knop) == HIGH)
  {
    digitalWrite(pin_led, HIGH);
  }
  else
  {
    digitalWrite(pin_led, LOW);
  }
}
```



Hee, deze code herken ik!

4.2. Knop flip: Knop eeuwig aan: opdracht 1

Zet bovenaan de code, boven `setup`:

```
boolean is_aan = false;
```

In het begin van `loop`, gebruik een `if` om te kijken of de knop is ingedrukt. Zo ja, zet `is_aan` op `true`:

```
if (/* de knop is ingedrukt */)
{
  is_aan = true;
}
```

Na de eerste `if` in `loop`, gebruik een `if` om te kijken of `is_aan` gelijk is aan `true`. Zo ja, zet de LED aan. Zo nee, zet de LED uit.

```
if (is_aan == true)
{
  //Zet de LED aan
}
else
{
  //Zet de LED uit
}
```

Wat zie je?

4.3. Knop flip: Knop eeuwig aan: oplossing 2

```
// ...
boolean is_aan = false;

// ...

void loop()
{
  if (digitalRead(pin_knop) == HIGH)
  {
    is_aan = true;
  }

  if (is_aan == true)
  {
    digitalWrite(pin_led, HIGH);
  }
  else
  {
    digitalWrite(pin_led, LOW);
  }
}
```

Als je op de knop drukt, blijft de LED eeuwig aan.

4.4. Knop flip: Knop eeuwig aan: opdracht 2

We gaan nu zeggen:

- Als je de knop indrukt, gaat de LED aan en blijft aan
- Als je de knop dan weer indrukt, gaat de LED uit en blijft uit
- Als je de knop dan weer indrukt, gaat de LED aan en blijft aan
- Enzovoorts

Pas de if aan die kijkt of de knop is ingedrukt:

```
if (digitalRead(pin_knop) == HIGH)
{
  if (is_aan == true)
  {
    is_aan = false;
  }
  else
  {
    is_aan = true;
  }
}
```

Upload de code. Wat zie je? Als je iets raars ziet, klopt dat!



Wat zie je?



Als je iets raars ziet, klopt dat!

4.5. Knop flip: Knop eeuwig aan: oplossing 2

```
// ...  
  
void loop()  
{  
  if (digitalRead(pin_knop) == HIGH)  
  {  
    if (is_aan == true)  
    {  
      is_aan = false;  
    }  
    else  
    {  
      is_aan = true;  
    }  
  }  
  
  // ...  
  
}
```



Als je de knop indrukt, dimt het LEDje. Laat je de knop los, dan blijft het LEDje of eeuwig aan of eeuwig uit



Dit komt omdat de Arduino snel en vaak het lampje aan en uit zet

4.6. Knop flip: Knop eeuwig aan met fijne besturing: opdracht 2

We gaan ervoor zorgen dat de knop beter reageert: als de knop ingedrukt wordt, laat de Arduino dan 200 milliseconden wachten.

4.7. Knop flip: Knop eeuwig aan met fijne besturing: oplossing 2

In de if van als de knop wordt ingedrukt, voeg een `delay(200);` regel toe. Dit kan voor of na de if statements met `is_aan`.

```
// ...

void loop()
{
  if (digitalRead(pin_knop) == HIGH)
  {
    // ... [if (is_aan == true) ...]

    delay(200);

    // ... [if (is_aan == true) ...]
  }

  // ...
}
```

4.8. Knop flip: twee LEDs besturen: opdracht

We gaan een tweede LED aansluiten. Door de knop in te drukken, kun je wisselen tussen leds.

- Sluit een tweede, groene LED aan op pin 12
- In de code, verander bovenaan:

```
// pin_led gaat weg
const int pin_led_rood = /* pin nummer */;
const int pin_led_groen = /* pin nummer */;
// ... [pin_knop blijft hetzelfde]
// is_aan gaat weg
int welke_led_aan = 1;
```

- In de code, in `setup`, zorg dat de twee LEDs en de knop worden gevonden.
- In de code, in `loop`, reageer anders op de knop:

```
if (/* de knop is ingedrukt */)
{
  welke_led_aan = welke_led_aan + 1;
  if (welke_led_aan == 3)
  {
    welke_led_aan = 1;
  }

  // ... [wacht 200 milliseconden]
}
```

- In de code, in `loop`, reageer nu op `welke_led_aan`:

```
if (welke_led_aan == 1)
{
  // Zet rode LED aan, zet groene LED uit
}
if (welke_led_aan == 2)
{
  // Zet rode LED uit, zet groene LED aan
}
```

4.9. Knop flip: twee LEDs besturen: oplossing

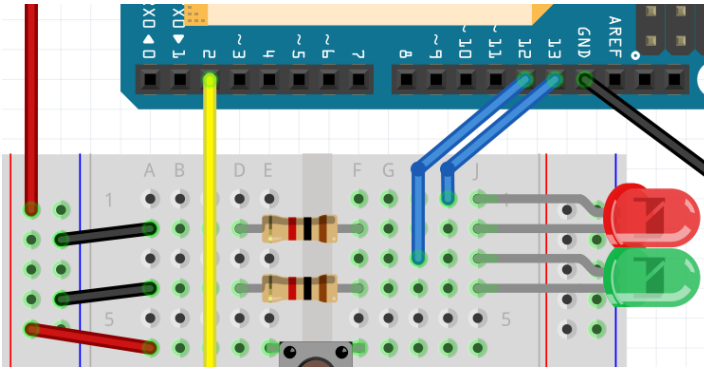


Figure 17: 4.9 Knop flip: twee LEDs besturen: oplossing

```
const int pin_led_rood = 13;
const int pin_led_groen = 12;
const int pin_knop = 2;
int welke_led_aan = 1;

void setup()
{
  pinMode(pin_led_rood, OUTPUT);
  pinMode(pin_led_groen, OUTPUT);
  pinMode(pin_knop, INPUT);
}

void loop()
{
  if (/* de knop is ingedrukt */)
  {
    welke_led_aan = welke_led_aan + 1;
    if (welke_led_aan == 3)
    {
      welke_led_aan = 1;
    }

    // ... [wacht 200 milliseconden]
  }

  if (welke_led_aan == 1)
  {
    digitalWrite(pin_led_rood, HIGH);
    digitalWrite(pin_led_groen, LOW);
  }
  if (welke_led_aan == 2)
  {
    digitalWrite(pin_led_rood, LOW);
    digitalWrite(pin_led_groen, HIGH);
  }
}
```

4.10. Knop flip: eindopdracht

We gaan een derde LED aansluiten. Door de knop in te drukken, kun je wisselen van eerste, naar tweede, naar derde LED.