



# Arduino uitleg voor werkbladen start

Coderdojo Tervuren

Leren door doen

# Werkbladen 1

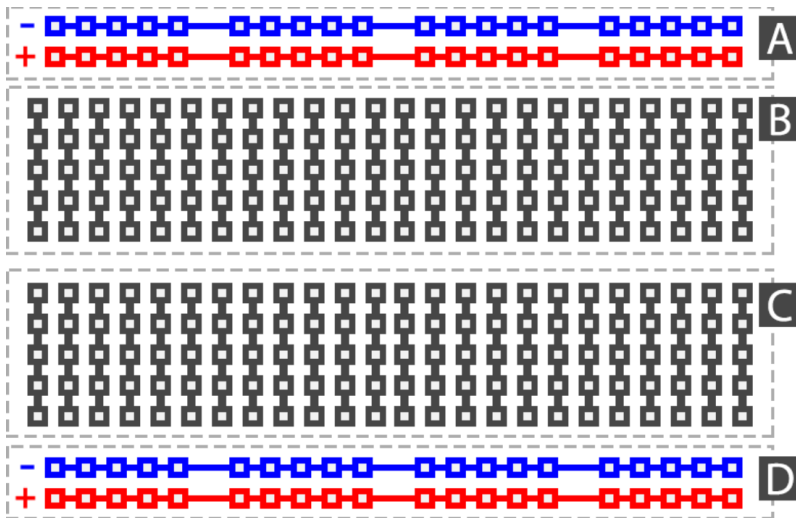
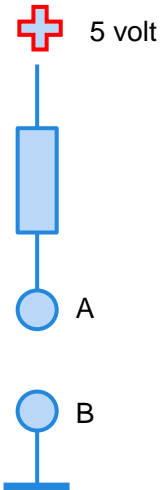
De inhoud van Werkbladen 1 concentreert op kennismaking en eenvoudige programmeer voorbeelden en oefeningen

- Kennismaken met de taal
- Oefeningen met LED, drukknop, toon opwekking en LCD aansturing

Leren door doen

- Maak de oefeningen individueel of in een groepje
- Thuis oefenen
- Kijk op internet of vraag aan anderen als je vastzit

# Pull-up & Breadboard



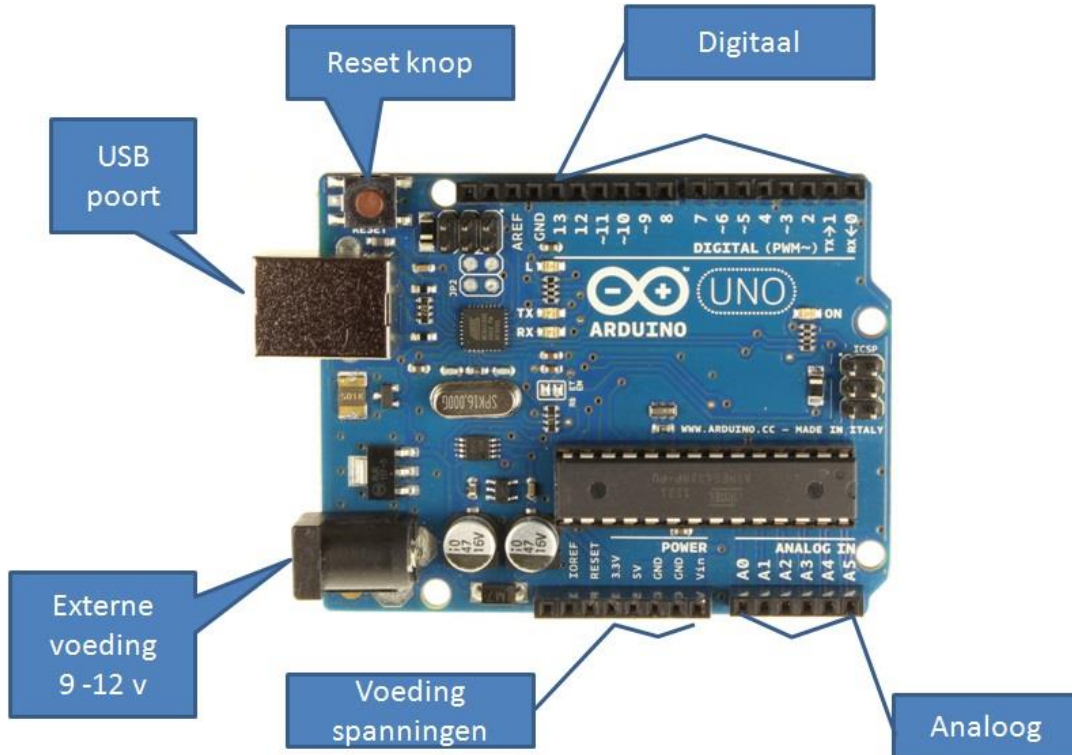
Verbindingen  
horizontaal

Verbindingen  
verticaal

Verbindingen  
verticaal

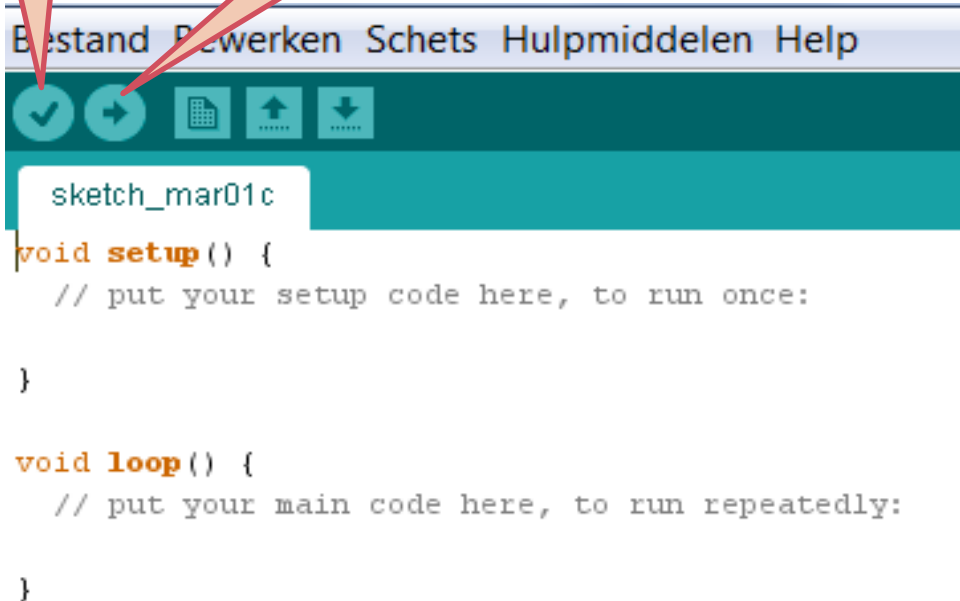
Verbindingen  
horizontaal

# De Arduino UNO



Hiermee  
vertaal je  
het  
programma

Hiermee laad je het  
programma in de  
Arduino



```
Bestand Bewerken Schets Hulpmiddelen Help  
sketch_mar01c  
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

# De programmeertaal

## **void setup() {.....}**

De code binnen de {.....} wordt alleen aan het begin uitgevoerd. Dit is wanneer de Arduino gereset wordt.

## **void loop() {.....}**

De code binnen de {.....} wordt als een oneindige loop uitgevoerd.

## **pinMode(pin, mode);**

Met pinMode definieer je een bepaalde pin als in of uitgang. Mode kan INPUT of OUTPUT zijn. Bijvoorbeeld: pinMode(13, OUTPUT);

## **digitalWrite(pin, output);**

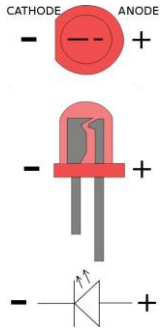
Hiermee bepaal je de digitale waarde van een bepaalde pin. Output kan HIGH of LOW zijn. Bijvoorbeeld: digitalWrite(13, HIGH);

## **delay(time);**

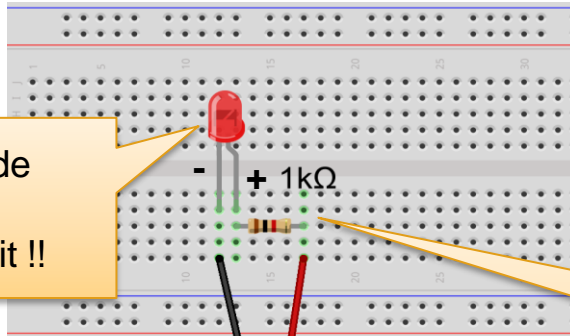
Hiermee laat je het programma pauzeren, de tijd wordt aangegeven in milliseconden. Bijvoorbeeld: delay(1000);

**// commentaar.** Wat je achter **//** op de regel schrijft wordt niet door het programma gebruikt

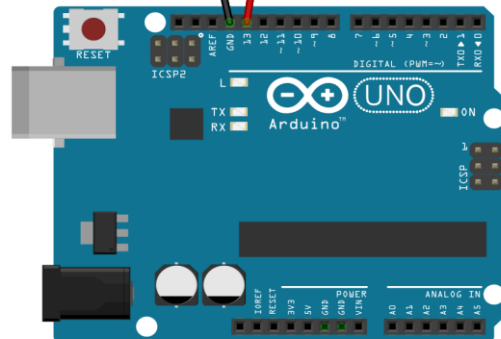
# De knipperende LED



Let op de  
juiste  
polariteit !!



Weerstand  
mag ook 220  
of 470Ω zijn





# De setup

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

# Omzetten naar 'C'

Begin:

doe de led aan

wacht 1 seconde 

doe de led uit

wacht 1 seconde

ga naar Begin

```
void loop()
```

```
{
```

```
  digitalWrite(13, HIGH);
```

```
  delay(1000); // wachten
```

```
  digitalWrite(13, LOW);
```

```
  delay(1000); // wachten
```

```
}
```

# Oefening 1



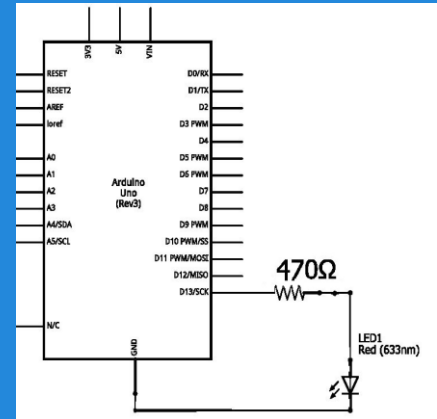
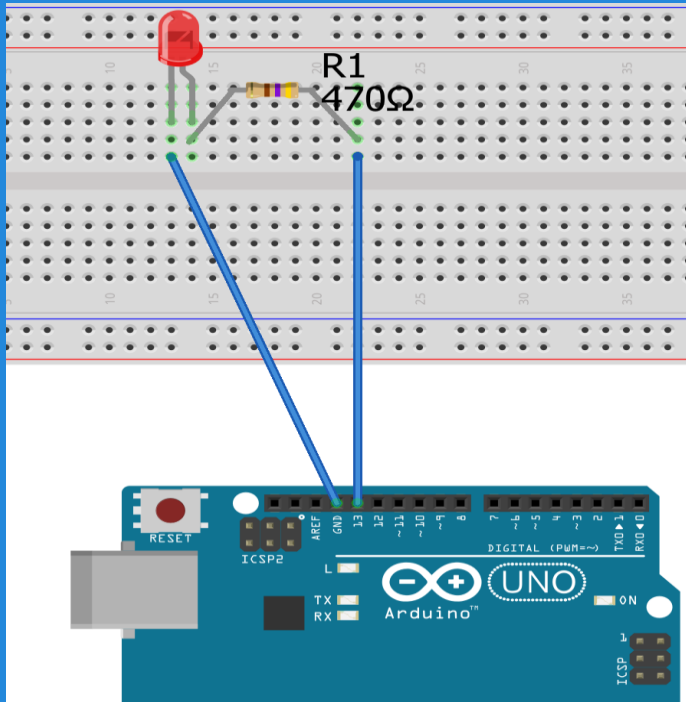
Bestand Bewerken Schets Hulpmiddelen Help



sketch\_oefening1

```
1 //oefening 1 de knipperende LED
2 void setup()
3 {
4   pinMode(13, OUTPUT);    // pin 13 wordt een digitale uitgang
5 }
6 void loop()
7 {
8   digitalWrite(13, HIGH); // pin 13 wordt hoog; de led gaat branden
9   delay(1000);           // wacht 1000 milliseconden
10  digitalWrite(13, LOW); // pin 13 wordt laag; de led gaat uit
11  delay(500);            // wacht 500 milliseconden
12 }
```

# Knipper LED



# Integer variabele

```
int x = 13; // declareer de variabele 'x' als integer
```

Aan de integer 'x' wordt de waarde 13 toegekend.

Integer variabelen zijn bedoeld om getallen te bewaren zonder een decimale punt, met een waarde tussen -32768 tot +32767

# Oefening 1A



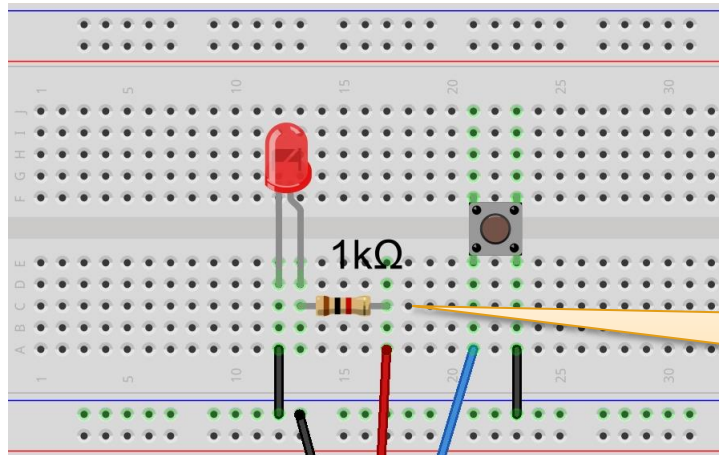
Bestand Bewerken Schets Hulpmiddelen Help



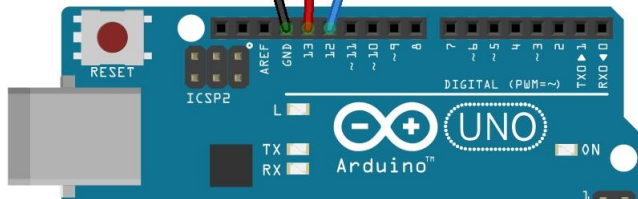
sketch\_oefening1A

```
1 // oefening 1A aangepast LED knipperprogramma
2 int ledpin = 13;           // de integer variabele ledpin wordt 13
3 void setup()
4 {
5   pinMode(ledpin, OUTPUT); // de pin ledpin wordt een digitale uitgang
6 }
7 void loop()
8 {
9   digitalWrite(ledpin, HIGH); // de pin ledpin wordt hoog; de led gaat branden
10  delay(1000);                // wacht 1 seconde
11  digitalWrite(ledpin, LOW);  // de pin ledpin wordt laag; de led gaat uit
12  delay(500);                 // wacht 0,5 seconde
13 }
```

# Oefening 2: de drukknop



Weerstand mag ook 220 of 470Ω zijn





# Enkele nieuwe statements

## **digitalWrite(inputpin, HIGH)**

Er wordt een pull-up weerstand geactiveerd waardoor deze pin standaard HIGH wordt, tenzij er een verbinding met de ground wordt gemaakt.

## **digitalRead(pin)**

Leest de digitale waarde van een digitale INPUT pin. (Geeft een boolean terug, HIGH, LOW of TRUE, FALSE of 1, 0)

## **if (condition) {.....} else {.....}**

Voert de code binnen de eerste haakjes uit wanneer wat tussen de ronde haakjes staat waar (TRUE, HIGH of 1) is, anders wordt de code tussen de haakjes na else uitgevoerd.

**==** Betekent 'is gelijk aan'! Niet verwarren met de enkele '=', want die kent een waarde toe aan een variabele

# Het 'if' statement

```
if(condition) {.....} else {.....}
```

Voert de code binnen de eerste haakjes uit wanneer wat tussen de ronde haakjes staat waar (TRUE, HIGH of 1) is, anders wordt de code tussen de haakjes na else uitgevoerd.

```
if (digitalRead(dkpin) == LOW)           // controleer of drukknop is ingedrukt
```

```
{  
  digitalWrite(ledpin, HIGH);           // zet LED aan  
}
```

```
else
```

```
{  
  digitalWrite(ledpin, LOW);           // zet LED uit  
}
```

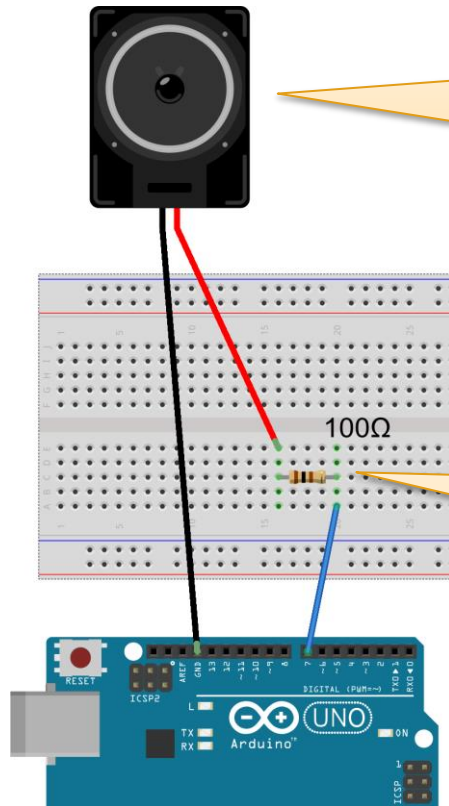
# Oefening 2



Bestand Bewerken Schets Hulpmiddelen Help

```
sketch_oefening_2
1 // oefening 2 de Druknop
2 // benoem de pin nummers:
3 int dkpin = 12; // pin nummer voor de drukknoop
4 int ledpin = 13; // pin nummer voor de LED
5
6 void setup()
7 {
8   pinMode(dkpin, INPUT); // de pin voor de drukknoop pin is een input
9   pinMode(ledpin, OUTPUT); // de pin voor de LED pin is een output
10  digitalWrite(dkpin, HIGH); // activeer pull-up weerstand
11 }
12 void loop()
13 {
14   if (digitalRead(dkpin) == LOW) // controleer of drukknoop is ingedrukt
15   {
16     digitalWrite(ledpin, HIGH); // zet LED aan
17   }
18   else
19   {
20     digitalWrite(ledpin, LOW); // zet LED uit
21   }
22 }
```

# Oefening 3: Toon generatie



De luidspreker kan ook een passieve buzzer zijn; let dan wel op dat deze op de juiste manier wordt aangesloten

De weerstand van  $100\Omega$  is soms niet nodig. (vraag dit aan de coach)



# Toon generatie statements

## **tone(pin, frequentie);**

Er wordt een toon signal op uitgang 'pin' gegenereerd.  
Bijvoorbeeld `tone(7,800);`

## **noTone(pin);**

Stop de toon generatie op de uitgang 'pin'.  
Bijvoorbeeld: `noTone(7);`

# Oefening 3

Bestand Bewerken Schets Hulpmiddelen Help

A screenshot of the Arduino IDE interface. At the top, there is a menu bar with "Bestand", "Bewerken", "Schets", "Hulpmiddelen", and "Help". Below the menu bar is a toolbar with icons for a checkmark, a right arrow, a document, an upload arrow, and a download arrow. The main workspace shows a code editor with the filename "sketch\_oefening\_3\$" and the following code:

```
1 // Oefening 3 toon opwekking
2 void setup()
3 {
4     pinMode(7, OUTPUT);      // pin 7 wordt de aansluiting voor de luidspreker
5 }
6 void loop()
7 {
8     tone(7,800);             // toon van 800 Hz op pin 7
9     delay(500);              // laat toon 500 ms aan
10    noTone(7);                // schakel toon uit
11    delay(1000);
12 }
```

# De subroutine

```
void punt()  
{  
  tone(7,800);           // toon aan  
  delay(t);             // toon t ms (punt) aan  
  noTone(7);           // schakel toon uit  
  delay(t);            // spatie  
}  
  
void streep()  
{  
  tone(7,800);           // toon aan  
  delay(3 *t);          // toon 3 keer t ms (streep) aan  
  noTone(7);           // schakel toon uit  
  delay(t);            // spatie  
}
```



sketch\_oefening\_3A

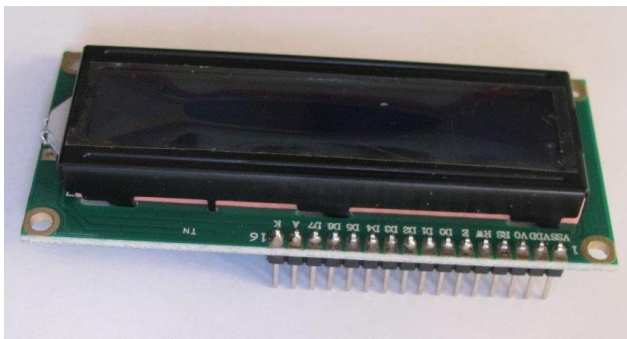
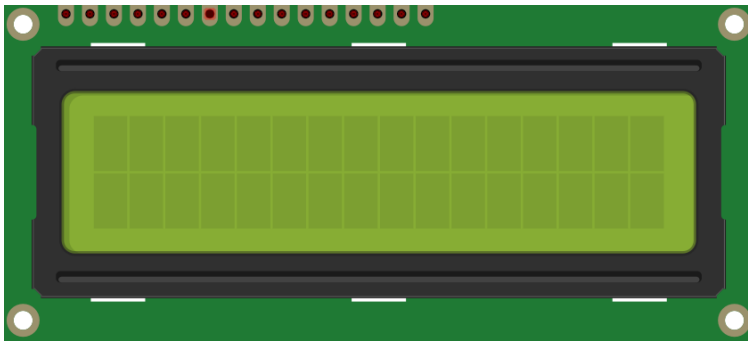
```
1 //oefening 3A het seinen van de letter r (punt, streep, punt)
2 // in dit programma worden subroutines gebruikt
3
4 int t=200;                // tijdsduur van een punt; sein snelheid
5
6 void setup()
7 {
8   pinMode(7, OUTPUT);    // pin 7 wordt de aansluiting voor de luidspreker
9 }
10
11 void punt()
12 {
13   tone(7,800);           // toon aan
14   delay(t);              // toon t ms (punt) aan
15   noTone(7);            // schakel toon uit
16   delay(t);              // spatie
17 }
18
19 void streep()
20 {
21   tone(7,800);           // toon aan
22   delay(3 *t);           // toon 3 keer t ms (streep) aan
23   noTone(7);            // schakel toon uit
24   delay(t);              // spatie
25 }
26
27
28 void loop()
29 {
30   punt();                // roep subroutine punt aan
31   streep();              // roep subroutine streep aan
32   punt();                // roep subroutine punt aan
33   delay(1000);
34 }
```

# Het LCD



1

16



LCD	Functie
1	VSS 0 volt
2	VDD 5 volt voeding
3	VO contrast instelling
4	RS Register Select
5	RW Read/Write
6	E Enable
7	D0 data signaal 0
8	D1 data signaal 1
9	D2 data signaal 2
10	D3 data signaal 2
11	D4 data signaal 4
12	D5 data signaal 5
13	D6 data signaal 6
14	D7 data signaal 6
15	A LED verlichting +
16	K LED verlichting -



# LCD en 'for' loop

**#include <LiquidCrystal.h>**

Het programma wordt geïnstrueerd om de functie 'LiquidCrystal.h' uit de bibliotheek te gebruiken.

Enkele instructies van deze bibliotheek:

**LiquidCrystal lcd(12, 11, 5, 4, 3, 2);** LCD RS pin aan digital pin 12, LCD Enable pin aan digital pin 11, LCD D4 pin aan digital pin 5, LCD D5 pin aan digital pin 4, LCD D6 pin aan digital pin 3, LCD D7 pin aan digital pin 2

**lcd.begin(16, 2);** LCD heeft 16 posities per rij en 2 rijen

**lcd.setCursor(x, y);** De cursor wordt op positie x van regel y gezet (bijvoorbeeld 0,0)

**lcd.print("tekst");** Vanaf de huidige cursor positie wordt het woord 'tekst' op het LCD geschreven

**for (i = 0 ; i < 10 ; ) { .....i ++.....};**    wat andere vorm    **for (int i = 0 ; i < 10 ; i++) { ..... };**

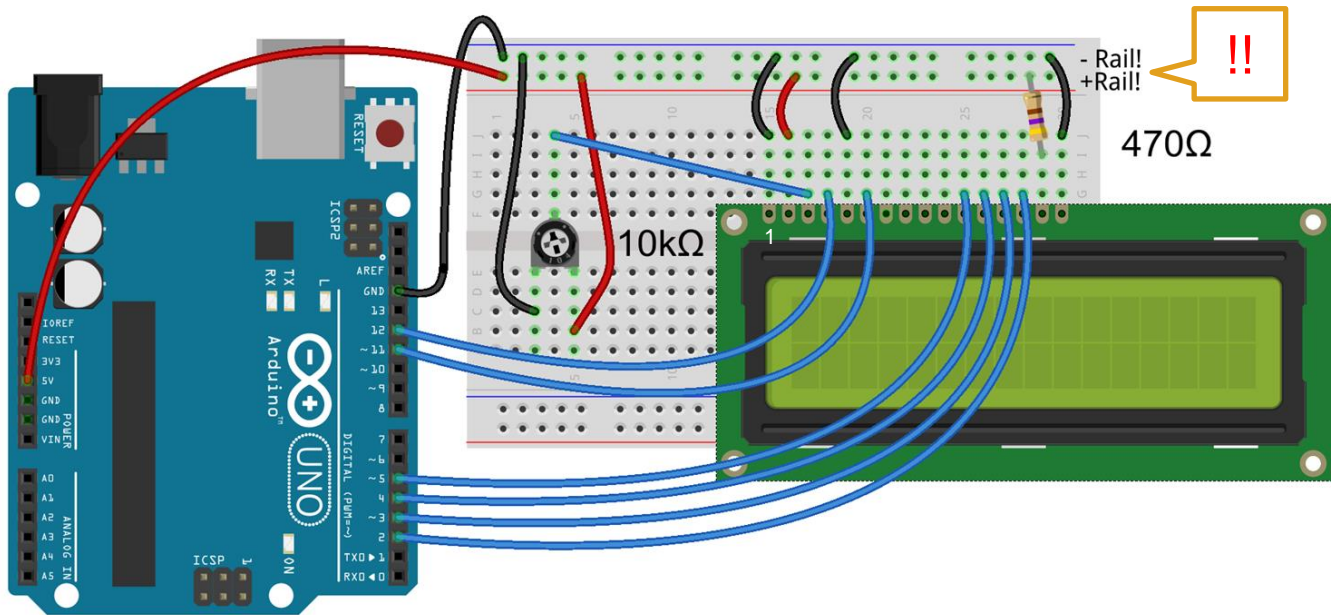
Deze code wordt gebruikt om bewerkingen te herhalen

Het eerste stuk wordt eenmalig aan het begin uitgevoerd (i = 0).

Als i kleiner is dan 10 dan wordt de code tussen de haakjes uitgevoerd, als i niet kleiner is dan 10 dan wordt het 'for' statement beëindigd

**i++** is een verkorte notatie voor  $i = i + 1$ .

# Oefening 4: Het LCD





sketch\_oefening\_4

```
1 // oefening 4 het LCD
2
3 #include <LiquidCrystal.h>
4 int i=0;
5
6 // LCD RS pin aan digital pin 12
7 // LCD Enable pin aan digital pin 11
8 // LCD R/W pin aan ground
9 // LCD D4 pin aan digital pin 5
10 // LCD D5 pin aan digital pin 4
11 // LCD D6 pin aan digital pin 3
12 // LCD D7 pin aan digital pin 2
13
14 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
15
16 void setup()
17 {
18     lcd.begin(16, 2);           // LCD type 16 x 2
19     lcd.setCursor(0, 0);       // zet cursor op positie 0 (vooraan) en rij 0 (eerste regel)
20     lcd.print("hello, world!"); // schrijf 'hello, world' op LCD
21 }
22
23 void loop()
24 {
25     for( i=0; i<10;)
26     {
27         lcd.setCursor(0, 1);    // zet cursor op positie 0 (vooraan) en rij 1 (tweede regel)
28         lcd.print(i);           // schrijf de waarde van i op het LCD
29         delay(1000);            // wacht 1 sec
30         i++;                     // hoog i met 1 op
31     }
32 }
```

# Verkeerslicht

