

TINKEREN MET DE ARDUINO

Workshop 1. Introductie

WAT BETEKENT TINKER?

tinker ('tɪŋkə/ noun: tinker; plural noun: tinkers)

1. (especially in former times) a person who makes a living by travelling from place to place mending pans and other metal utensils.

British derogatory: **a Gypsy or other person living in an itinerant community.**

2. British informal: **a mischievous child.**

"little tinkers, we were"

3. an act of attempting to repair something. (verb: tinker; 3rd person present: tinkers; past tense: tinkered; past participle: tinkered; gerund or present participle: tinkering)

1. **attempt to repair or improve something in a casual or desultory way.**

PROGRAMMA

- Wat is een Arduino
- Introductie elektronica en schakelingen
- De arduino IDE
- C-programmeren (zonder pointer ellende)
- Zelf tinkeren - een stoplicht

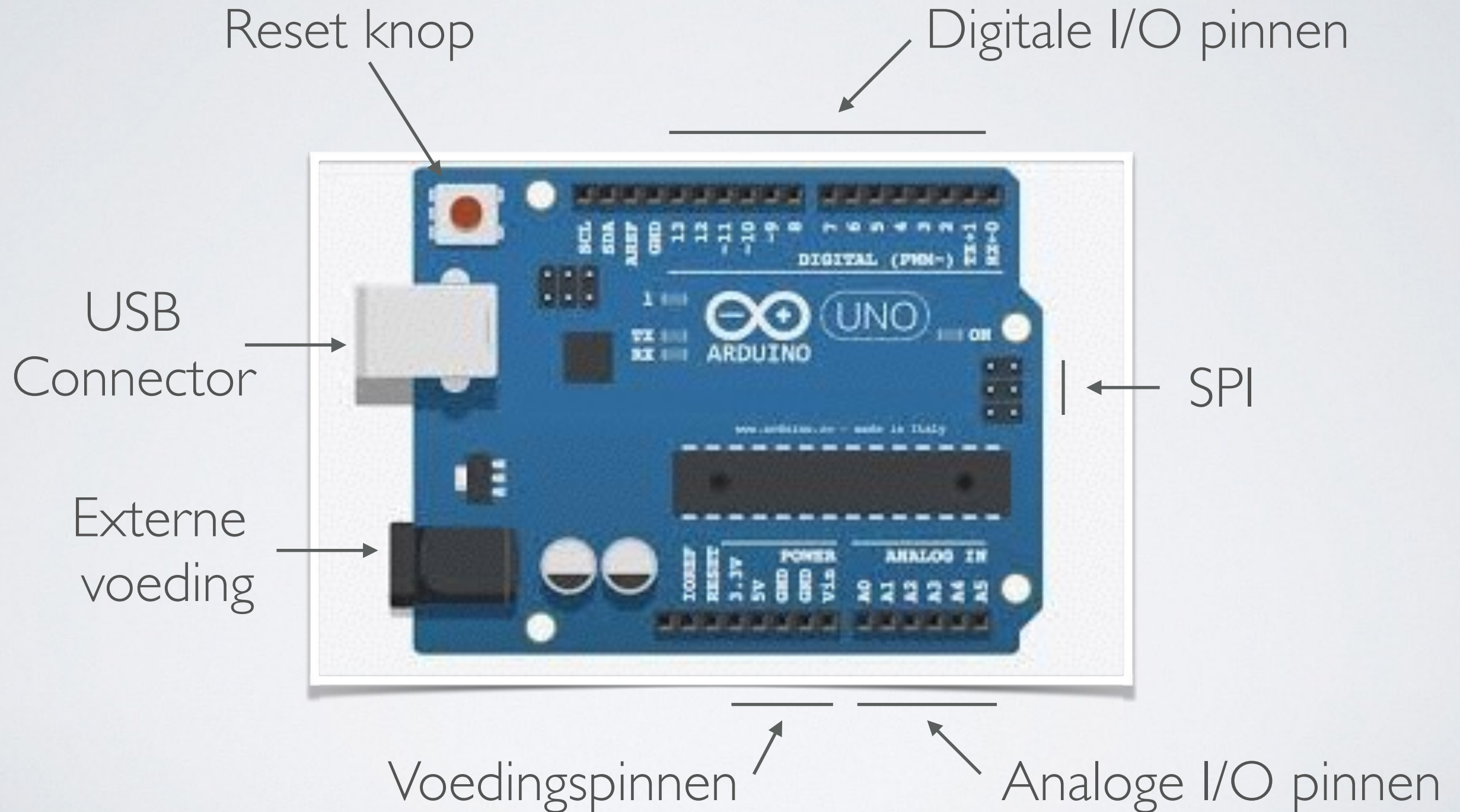
WAT IS EEN ARDUINO

- Een Arduino is een microcontroller, met 14 digitale en 6 analoge pinnen. 32kB geheugen waarvan 2k voor variabelen e.d.
- Een microcontroller kan maar een ding, maar dat doet hij wel heel goed (je kunt geen programma's toevoegen zoals op een normale computer).
- Een microcontroller kun je op de milliseconde nauwkeurig timen.
- Nadeel, heeft een beperkte hoeveelheid geheugen. Weinig basisfunctionaliteit (wel makkelijk uit te breiden).
- Er zijn veel verschillende uitvoeringen van de Arduino, we maken gebruik van de Arduino Uno.

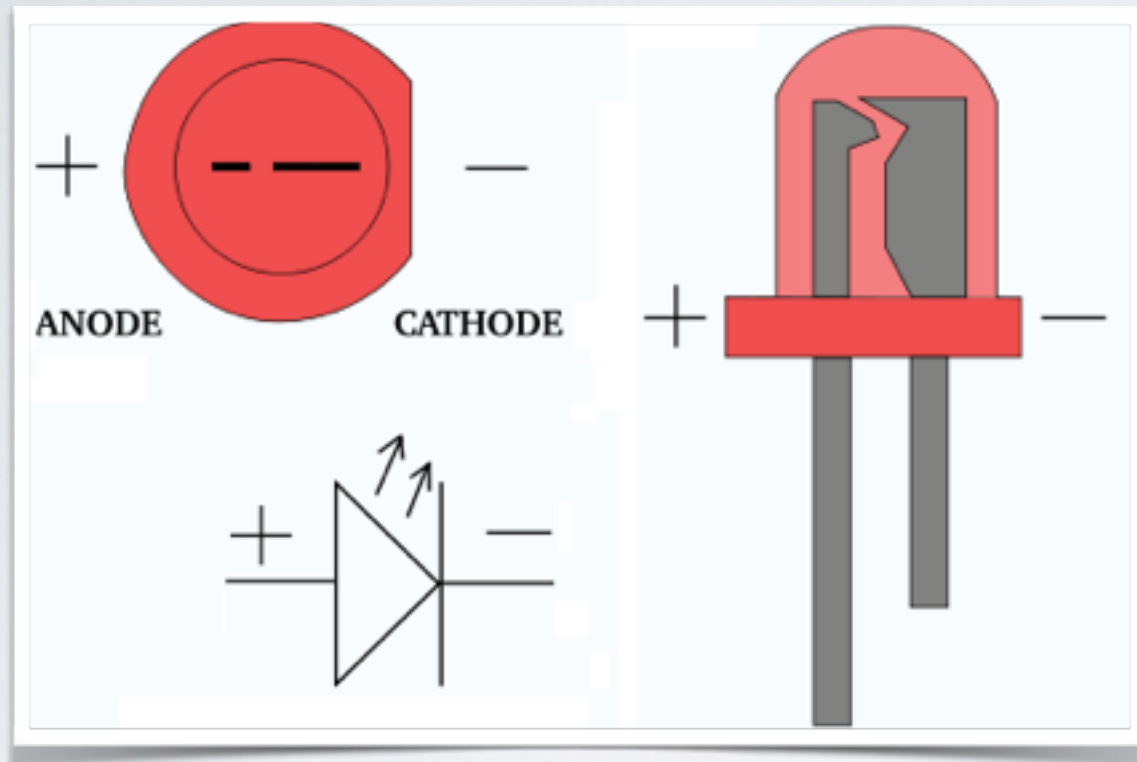
WAT MAAKT EEN ARDUINO LEUK EN TOCH OOK LASTIG

- Het programmeren van een Arduino is vrij makkelijk
- De combinatie elektronica met software, maakt fouten opsporen niet altijd even makkelijk
- Elektronica lijkt in het begin “eng”, maar met een beetje basiskennis kom je al gauw heel ver (de meeste componenten kun je zo op de Arduino aansluiten).

OVERZICHT ARDUINO (UNO)



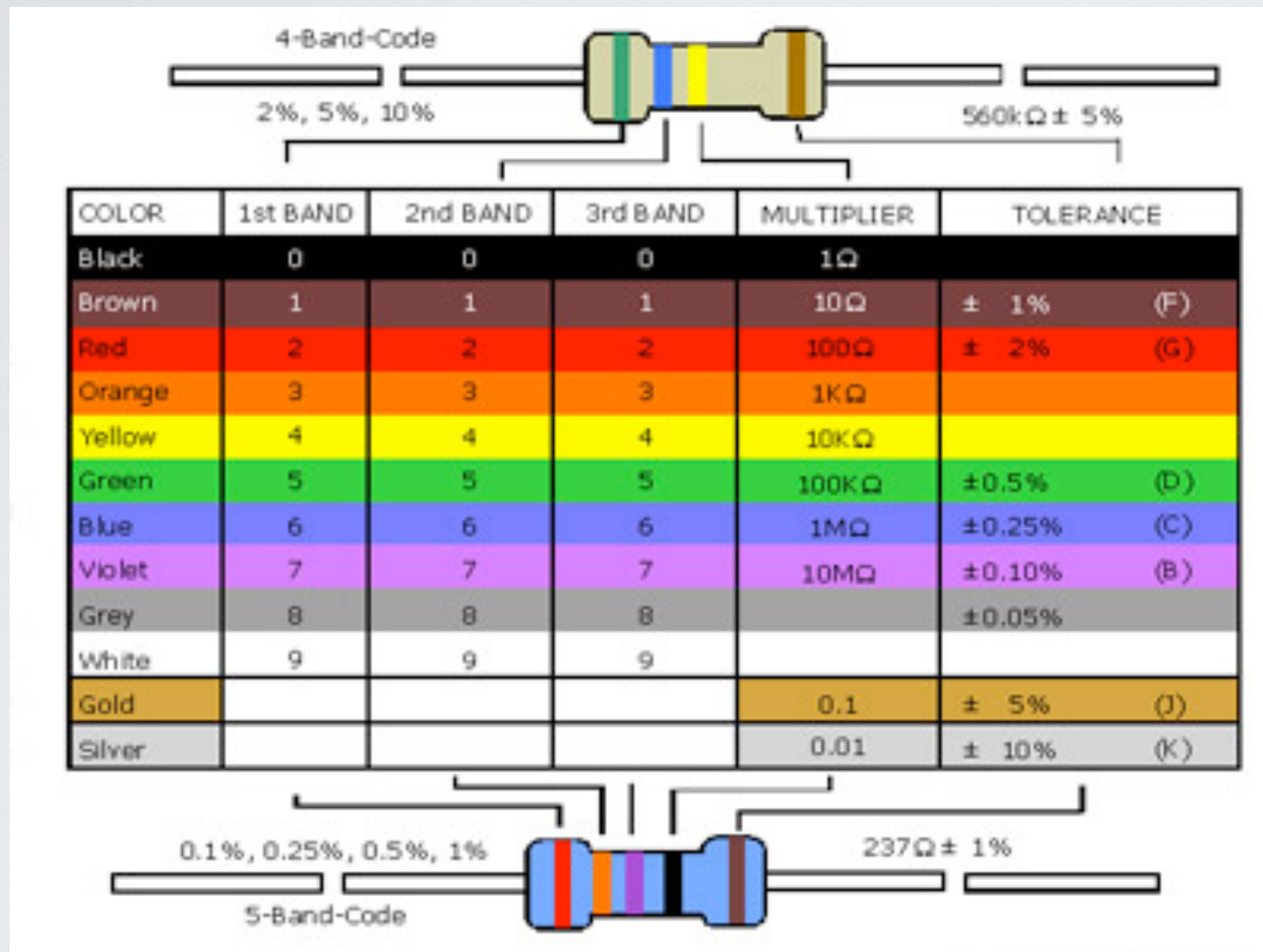
ELEKTRICITEIT - POLARITEIT



Stroom elektronen die van + naar - bewegen (in Qoulomb).
Stroom (Ampere) heeft spanning (Voltage) nodig om te kunnen stromen.

Weerstand (in Ohm) verlaagt stroom.

WEERSTAND KLEURCODERING

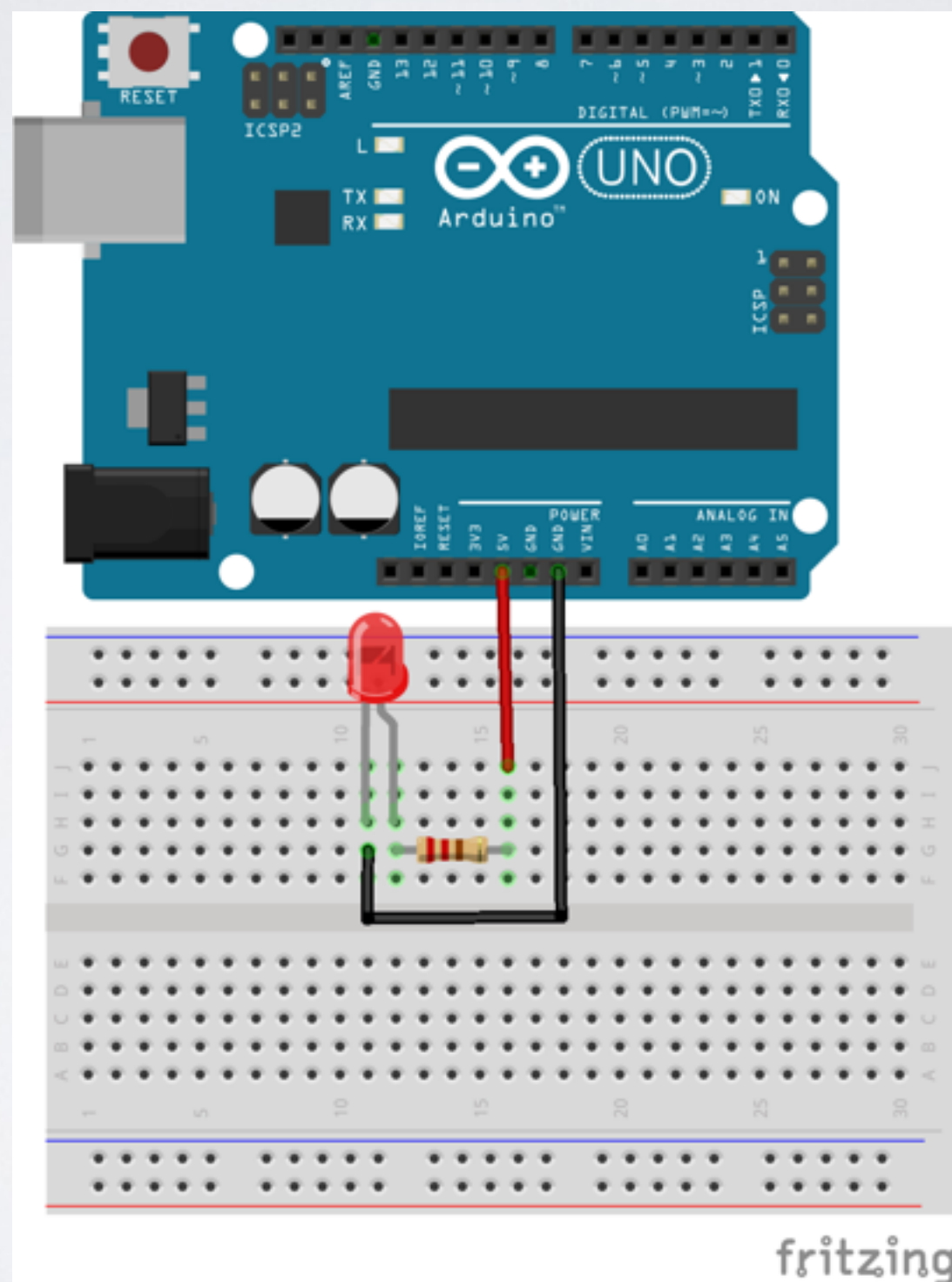


Zij bracht rozen, op gerrits graf bij vies grauw weer
(Zwart, BRuin, Rood, Oranje, GEEL, GROEN, BLAUW, VIOLET, GRIJS, WIT)

DE WET VAN OHM

- $U = I * R$
- $I = U / R$
- $R = U / I$
- $I = 5 / 220 = 0.023 = 23 \text{ mA}$ led mag 20mA hebben bij 3V.

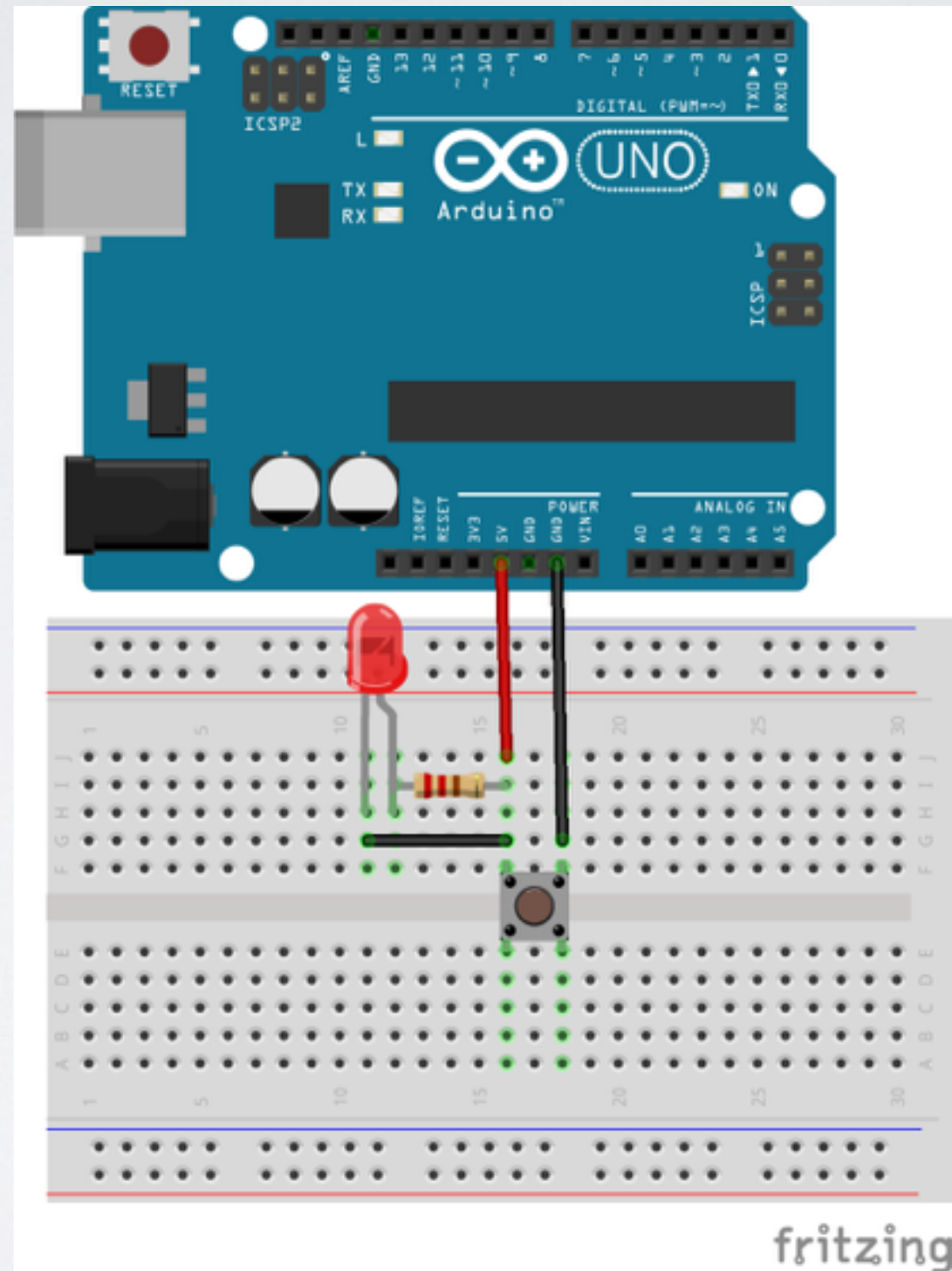
DE EERSTE SCHAKELING



SCHAKELINGEN OPBOUWEN

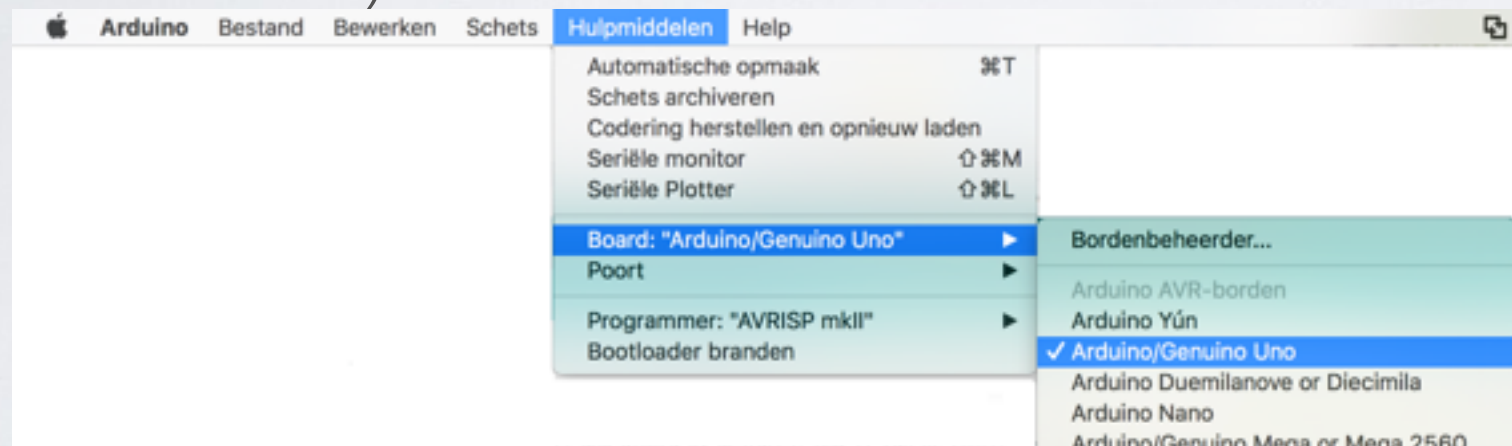
- Het programmeren van een Arduino is vrij makkelijk
- De combinatie electronica met software, maakt fouten opsporen niet altijd even makkelijk
- Electronica lijkt in het begin “eng”, maar met een beetje basiskennis kom je al gauw heel ver (de meeste componenten kun je zo op de Arduino aansluiten).

LED MET SCHAKELAAR

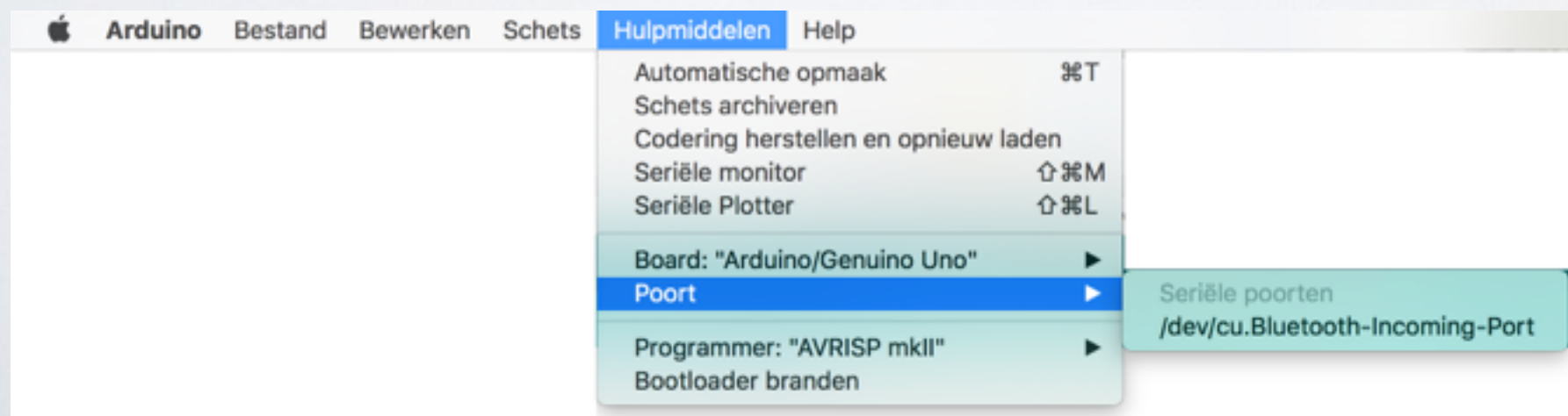


ARDUINO IDE

- De omgeving is down te loaden via <https://www.arduino.cc/en/Main/Software>
- Na download eerste het juiste boardje kiezen (wij gebruiken de Arduino/Genuino Uno)



- Daarna de juiste poort kiezen.



DE STRUCTUUR VAN EEN SKETCH

```
void setup() {
```

```
    // methode voor het eenmalig initialiseren van variabele, bibliotheken
```

```
    // en de hardware pinnen
```

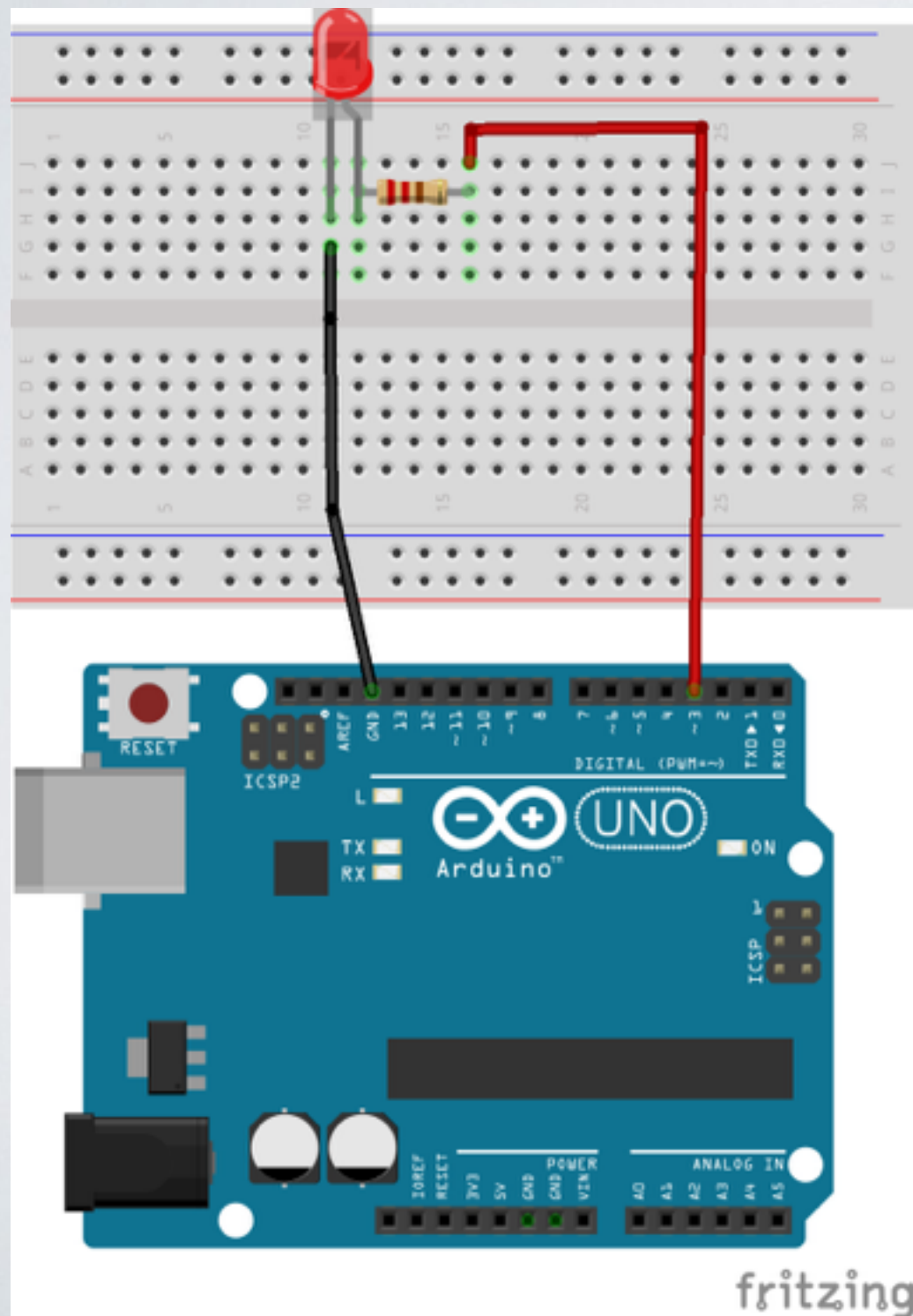
```
}
```

```
void loop() {
```

```
    // De echte code komt hier. De loop wordt continu herhaald
```

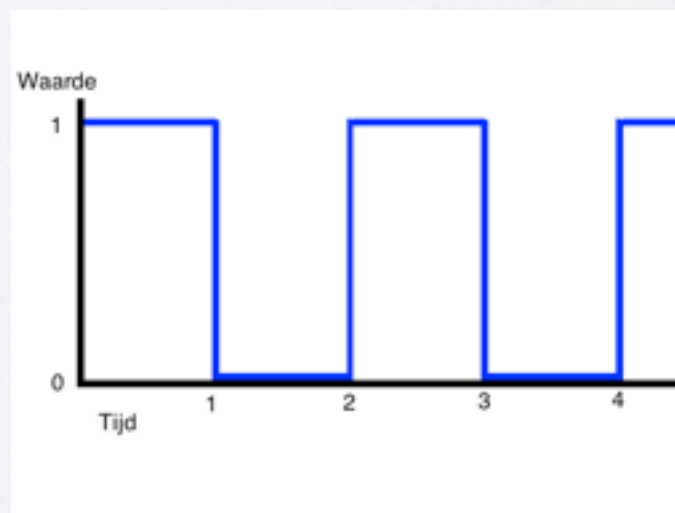
```
}
```


DE ARDUINO AAN HET WERK ZETTEN



```
void setup() {  
  pinMode( 3, OUTPUT ); // Laat de Arduino weten  
                        // dat we pin 3 gebruiken.  
}
```

```
void loop() {  
  digitalWrite( 3, HIGH ); // Zet de led aan  
  delay( 1000 );           // wacht 1 seconde  
  digitalWrite( 3, LOW );  // zet de led uit  
  delay( 1000 );           // wacht 1 seconde  
}
```



VERBETERINGEN

VERSIE 1

```
int LED_PIN = 3;    //Variabele voor de led pin
int interval = 1000; // variabele voor de duur
                    // van het interval (in ms)
```

```
void setup() {
  // Laat de Arduino weten dat we
  // de ledpin gebruiken.
  pinMode( LED_PIN, OUTPUT );
}
```

```
void loop() {
  // Zet de led aan
  digitalWrite( LED_PIN, HIGH );
  // wacht interval milliseconden
  delay( interval );
  // zet de led uit
  digitalWrite( LED_PIN, LOW );
  // wacht interval milliseconden
  delay( interval );
}
```

VERSIE 2

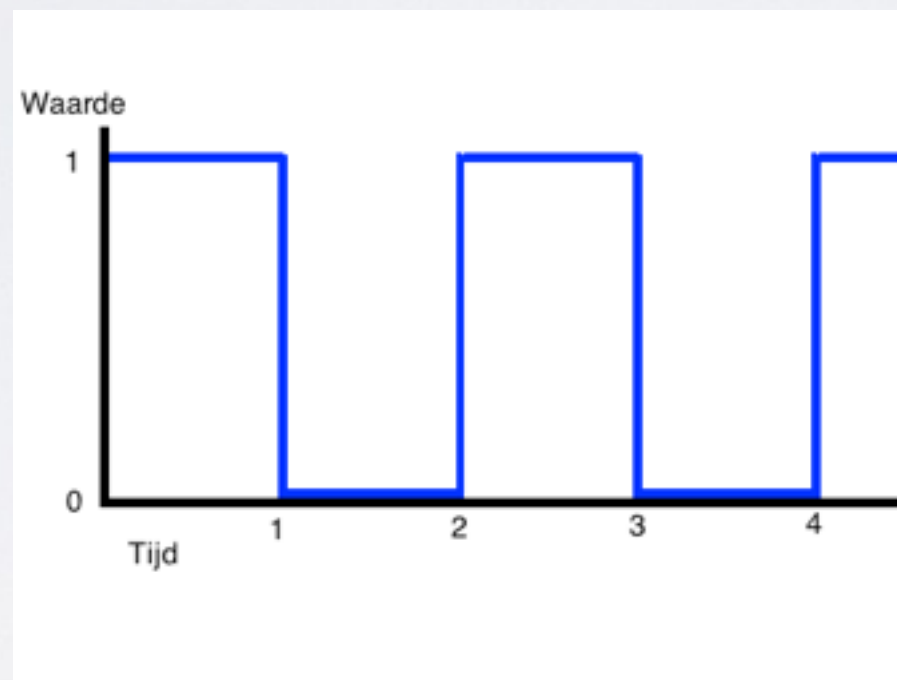
```
// Constante voor de led pin
#define LED_PIN 3
// constante voor de duur van het interval
#define interval 1000
```

```
void setup() {
  // Laat de Arduino weten dat we
  // de ledpin gebruiken.
  pinMode( LED_PIN, OUTPUT );
}
```

```
void loop() {
  // Zet de led aan
  digitalWrite( LED_PIN, HIGH );
  // wacht interval milliseconden
  delay( interval );
  // zet de led uit
  digitalWrite( LED_PIN, LOW );
  // wacht interval milliseconden
  delay( interval );
}
```

PULSE MODULATIE (PWM)

Wat gebeurt er als je het interval inkort, zodat de interval naar bijvoorbeeld microseconden gaat?



Let's find out!

PWM MET DE ARDUINO

```
#define LED_PIN 3
```

```
#define interval 1000
```

```
int i = 0;
```

```
void setup() {
```

```
    pinMode( LED_PIN, OUTPUT ); // Laat de Arduino weten dat we pin 3 gebruiken.
```

```
}
```

```
void loop() {
```

```
    for ( ; i <= 255; i++ ) {
```

```
        analogWrite( LED_PIN, i );
```

```
        delay( interval );
```

```
    }
```

```
    for ( ; i >= 0; i-- ) {
```

```
        analogWrite( LED_PIN, i );
```

```
        delay( interval );
```

```
    }
```

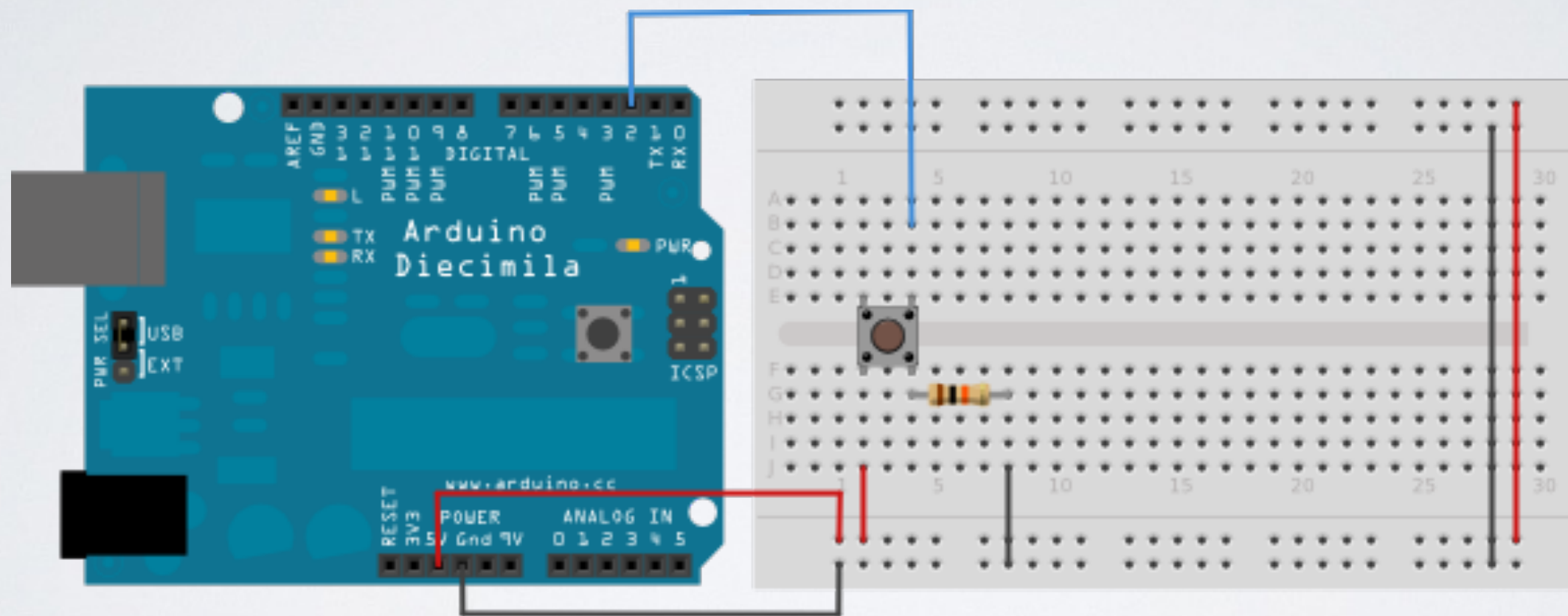
```
}
```

DE DIGITALE I/O PINNEN

- Pin 0 en pin 1, worden gebruikt voor seriële communicatie. Ook voor het uploaden van een sketch.
- Pin 3, 5, 6, 9, 10 en 11 zijn PWM pinnen
- Pin 2 en 3 kunnen gebruikt worden als interrupt pin (later meer hierover)
- Analoge pinnen (A0 t/m A5) kun je gewoon gebruiken als digitale pinnen 14 t/m 19

DIGITALE INPUT

- Een schakelaar uitlezen (met pullup weerstand)



SKETCH VOOR HET UITLEZEN

```
define buttonPin 2    // the number of the pushbutton pin

define ledPin 13      // the number of the LED pin

int buttonState = 0;   // variable for reading the pushbutton status

void setup() {

  pinMode(ledPin, OUTPUT); // initialize the LED

                           // pin as an output

  pinMode(buttonPin, INPUT); // initialize the pushbutton pin as an input

}
```

```
void loop() {

  buttonState = digitalRead(buttonPin); // read the state of the
  pushbutton value:

  // check if the pushbutton is pressed.

  if (buttonState == HIGH) { // if it is, the buttonState is HIGH:

    digitalWrite(ledPin, HIGH); // turn LED on:

  } else {

    digitalWrite(ledPin, LOW); // turn LED off:

  }

}
```

PROBLEEM MET DEZE MANIER

- De pin is niet altijd stabiel
- Hierdoor komt er soms een kleine piek op de pin
- Daardoor lijkt het alsof iemand op de knop drukt
- Komt vaak voor als de voeding niet stabiel is

OPLOSSING: DEBOUNCE

<https://github.com/thomasfredericks/Bounce2>

```
#include <Bounce2.h>

#define BUTTON_PIN 2

#define LED_PIN 13

Bounce debouncer = Bounce(); // Instantiate a Bounce object

void setup() {
  pinMode(BUTTON_PIN, INPUT_PULLUP); // Setup the button with an internal pull-up :

  debouncer.attach( BUTTON_PIN ); // After setting up the button, setup the Bounce
                                   // instance :

  debouncer.interval( 5 ); // interval in ms

  pinMode(LED_PIN, OUTPUT); // Setup the LED :
}
```

```
void loop() {

  boolean stateChanged = debouncer.update(); // Update the Bounce instance :

  int state = debouncer.read();

  if ( stateChanged && state == LOW ) { // Detect the falling edge

    if ( ledState == LOW ) {

      ledState = HIGH;

    } else {

      ledState = LOW;

    }

    digitalWrite(LED_PIN, ledState);

  }

}
```

C-PROGRAMMEREN

- Mensen met een Java achtergrond hebben een voordeel die kennen de syntax. Nadeel voor hen OO programmeren is wat anders dan embedded software!
- Logische structuren
- Conditionele loops
- functies (kijk uit die zijn “duur” in geheugen gebruik)

LOGISCHE STRUCTUREN (IF)

```
if ( voorwaarde1 ) {
```

```
    ...
```

```
} else if ( voorwaarde2 ) {
```

```
    ...
```

```
}
```

```
else {
```

```
    ...
```

```
}
```

VOORBEELD

```
if ( stateChanged && state == LOW ) {
```

```
    // uit te voeren code
```

```
}
```

LOGISCHE STRUCTURE SWITCH

```
switch ( selector ) {
```

```
    case voorwaarde:
```

```
        ...
```

```
        break;
```

```
    default:
```

```
        ...
```

```
        break;
```

```
}
```

VOORBEELD

```
#define STATE1 1
```

```
#define STATE2 2
```

```
switch ( status ) {
```

```
    case STATE1:
```

```
        Serial.println( "I" );
```

```
        break;
```

```
    default:
```

```
        Serial.println( "Anders" );
```

```
}
```


CONDITIONELE LOOPS

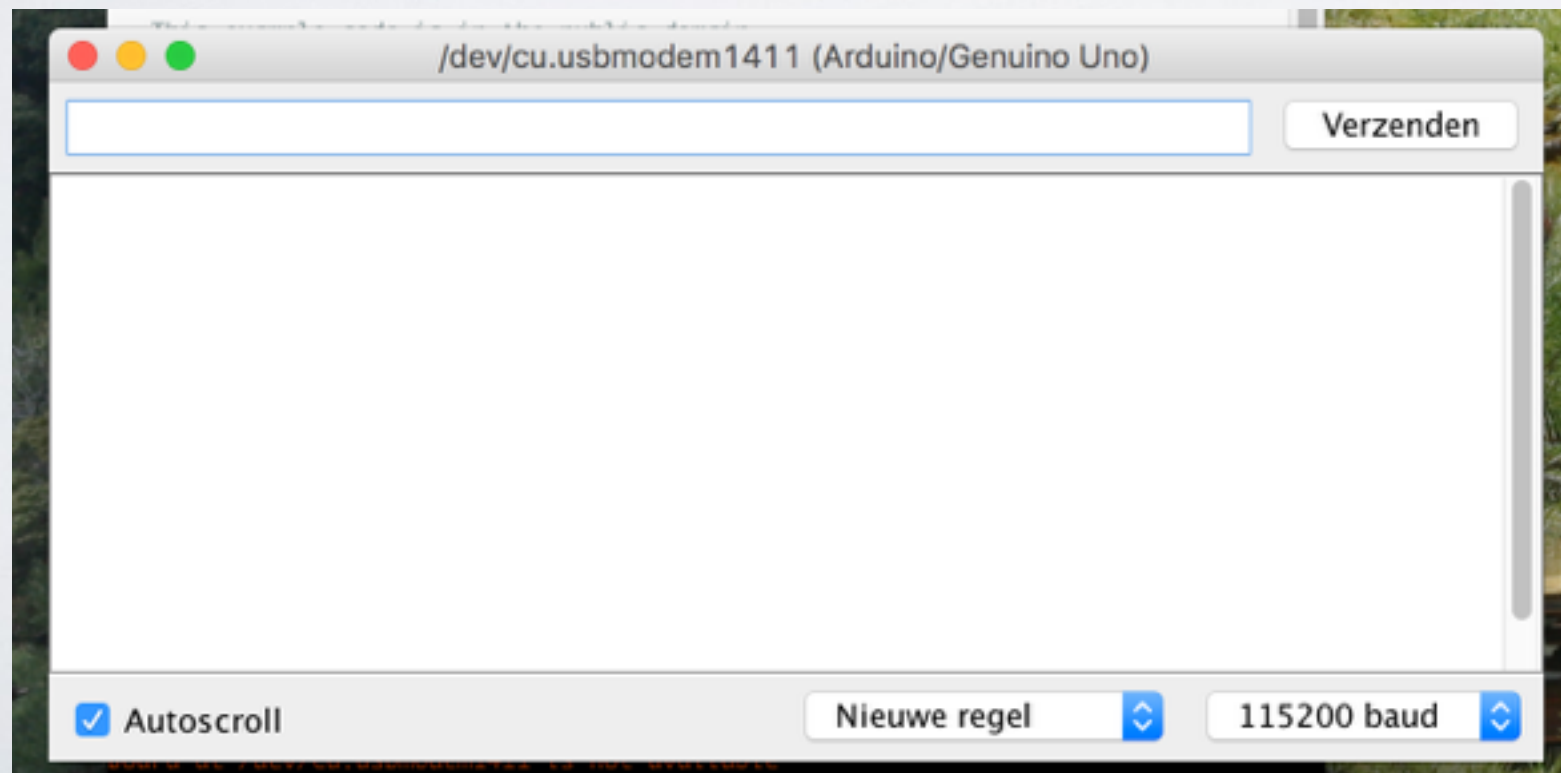
- `for (int i=0; i<10; i++) { .. }`
- `while (true == true || false == false) {}`

DEBUGGEN VIA SERIAL TERMINAL

- Opstarten via knop rechtsboven



- Monitor (met verzendknop)



SCHRIJVEN NAAR SERIËLE MONITOR

- `Serial.begin(115200);` // in setup. Baudrate
- Waar je maar debug wilt toevoegen:

```
Serial.print( "Hallo waarde: " );
```

```
Serial.println( waarde );
```

- of

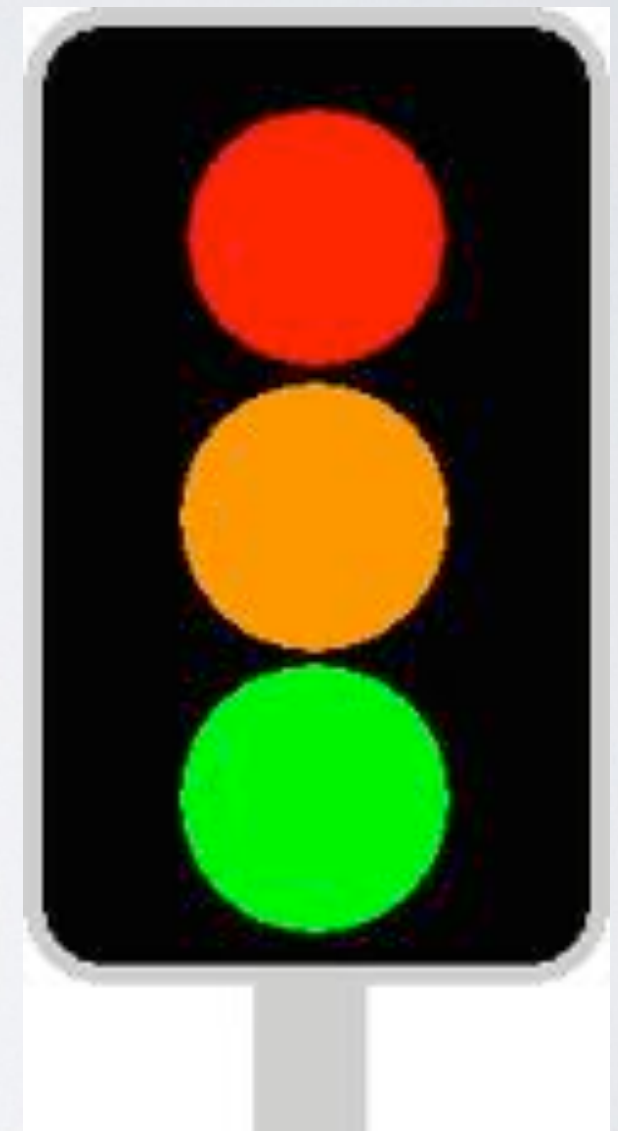
```
Serial.println( "Hallo waarde " + (String)waarde );
```

- Let op! Serial kost “veel” geheugen. De typecast naar String is ook een dure

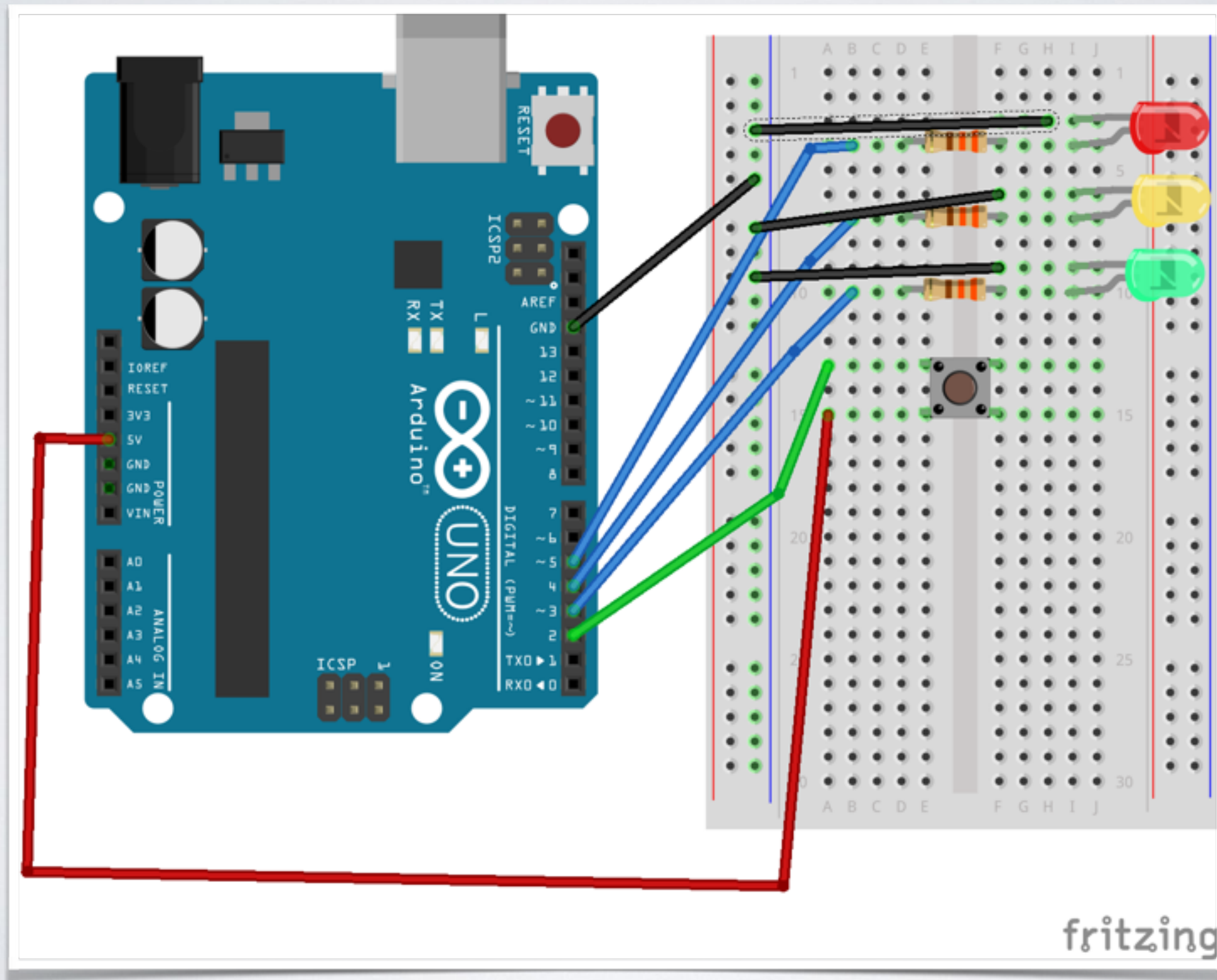
ZELFTINKEREN

Je hebt nu al heel wat basiskennis van de Arduino opgedaan. Tijd om zelf de vorige voorbeelden te combineren tot een eigen schakeling, met bijbehorende sketch.

De opdracht luidt. Maak een stoplicht. Als op de knop gedrukt wordt dan veranderd het licht van rood naar groen. Na 5 seconden gaat het licht op geel. En na 2 seconden gaat het licht op rood. De schakelaar wordt alleen uitgelezen als het licht op rood staat.



STOPLICHT SCHAKELING



STOPLICHT SKETCH

```
#include <Bounce2.h>
```

```
#define RED_LED 5
```

```
#define ORANGE_LED 4
```

```
#define GREEN_LED 3
```

```
#define SWITCH 2
```

```
#define RED_STATE 1
```

```
#define YELLOW_STATE 2
```

```
#define GREEN_STATE 3
```

```
#define GREENLIGHT_ACTIVE 5000
```

```
#define YELLOWLIGHT_ACTIVE 2000
```

```
int state = RED_STATE;
```

```
int state = RED_STATE;
```

```
Bounce debouncer = Bounce(); // Instantiate a Bounce object
```

```
void setup() {
```

```
    Serial.begin( 115200 );
```

```
    pinMode( RED_LED, OUTPUT );
```

```
    pinMode( ORANGE_LED, OUTPUT );
```

```
    pinMode( GREEN_LED, OUTPUT );
```

```
    pinMode( SWITCH, INPUT );
```

```
    digitalWrite( SWITCH,HIGH );
```

```
    // After setting up the button, setup the Bounce instance :
```

```
    debouncer.attach( SWITCH );
```

```
    debouncer.interval( 5 ); // interval in ms
```

```
    adjustLights();
```

```
}
```

```
void adjustLights() {

    digitalWrite( RED_LED, LOW );

    digitalWrite( ORANGE_LED, LOW );

    digitalWrite( GREEN_LED, LOW );

    switch ( state ) {

        case RED_STATE :

            digitalWrite( RED_LED, HIGH );

            break;

        case YELLOW_STATE :

            for ( int i = 0; i < 4; i++ ) {

                digitalWrite( ORANGE_LED, HIGH );

                delay( YELLOWLIGHT_ACTIVE / 8 );

                digitalWrite( ORANGE_LED, LOW );

                delay( YELLOWLIGHT_ACTIVE / 8 );

            }

            break;

        case GREEN_STATE :

            digitalWrite( GREEN_LED, HIGH );

            delay( GREENLIGHT_ACTIVE );

            break;

    }

}
```

```
void loop() {

    boolean stateChanged = debouncer.update(); // Update the Bounce instance :

    int value = debouncer.read(); // Get the updated value :

    if ( stateChanged && value == LOW ) { // remember pullup

        state = GREEN_STATE;

        adjustLights();

        state = YELLOW_STATE;

        adjustLights();

        state = RED_STATE;

        adjustLights();

    }

}
```

LEUK OM ZELF UIT TE ZOEKEN

- Geluid maken via de Arduino (met een speaker of een piezo element)
- Zie ook <http://www.arduino.cc> voor voorbeelden en library beschrijvingen

HUISWERK

YouTube kanaal Hobbyelektronica

- Aflevering 1 Gelijkspanning
- Aflevering 2 Wisselspanning

Aflevering 3 Stroom en weerstand

Aflevering 4 Stroom en weerstand Update

Aflevering 5 Vermogen