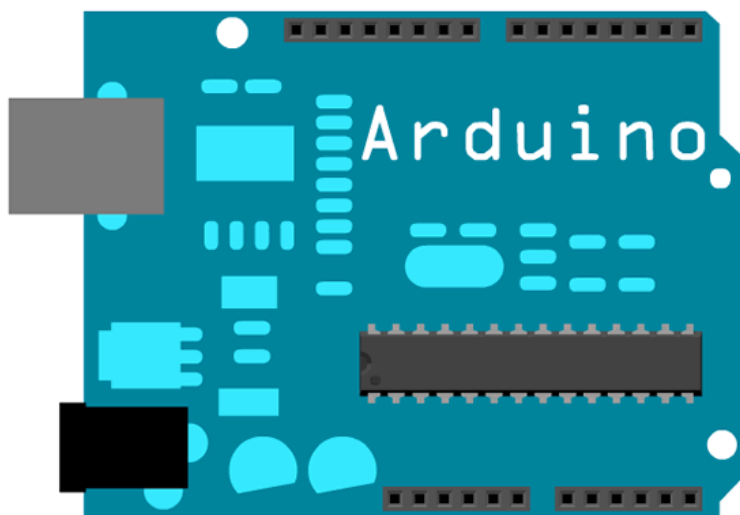




Programmieren



Op dit lesmateriaal is een Creative Commons licentie van toepassing.

© 2016-2019 Remie Woudt en Alborz Salimian Rizi

Infobroklede@gmail.com

De afbeelding op het voorblad is afkomstig van de website van Adafruit.

Adafruit was founded in 2005 by MIT hacker & engineer, Limor "Ladyada" Fried. Her goal was to create the best place online for learning electronics and making the best designed products for makers of all ages and skill levels. Adafruit has grown to over 100+ employees in the heart of NYC with a 50,000+ sq ft. factory.

Bron: <https://www.adafruit.com>

Delen van dit lesmateriaal is afkomstig van:
Brian W. Evans: "Arduino Programming Notebook"
Vertaling van A. Kompanje.

Inhoud

1

Physical computing

1.1 Zintuigen versus sensoren

Opdracht 1

Opdracht 2

Opdracht 3

1.2 Enkele veel gebruikte sensoren

Afbeelding 4

Afbeelding 5

Afbeelding 6

Afbeelding 7

Opdracht 4

Opdracht 5

Opdracht 6

Opdracht 7

2

Arduino (Genuino)

2.1 Wat is de Arduino?

Afbeelding 8: De Arduino Uno

Afbeelding 9: De Arduino Pro mini

2.2 Waar is het beeldscherm?

Afbeelding 10: De Arduino IDE

2.3 Aansluiten

Afbeelding 11: Het kiezen van het juiste board

Afbeelding 12: Het controleren en eventueel aanpassen van de poort in de software

2.4 Het breadboard

Afbeelding 13: Een breadboard

Afbeelding 14: Aansluitingen tussen de Arduino en het breadboard

Opdracht 8

2.5 Automaten

Afbeelding 15: een toestandsdiagram

2.5.1 Regels voor een toestandsdiagram

3

Programmeren

3.1 De programmastructuur

Opdracht 9

3.2 De programmeertaal

3.3 Aan de slag

Opdracht 10

Afbeelding 16: een toestandsdiagram voor een aanraakschakelaar

Opdracht 11

Opdracht 12

Afbeelding 17: een toestandsdiagram voor een toggle functie met een aanraakschakelaar

Opdracht 13

Opdracht 14

3.4 Aansluiten

3.4.1 Een LED aansluiten

3.4.2 Weerstanden

3.4.3 Een schakelaar aansluiten

Afbeelding 18: Pull-down weerstand

Afbeelding 19: Pull-up weerstand

3.5 Actuatoren

Afbeelding 20

Afbeelding 21

Afbeelding 22

Afbeelding 23

Afbeelding 24

Afbeelding 25

Afbeelding 26

Afbeelding 27

Opdracht 15

Opdracht 16

3.6 Terug naar de schakelingen

Opdracht 17

Opdracht 18

Opdracht 19

Opdracht 20

3.7 Motorshields

Afbeelding 28

Afbeelding 29

Afbeelding 30

Opdracht 21

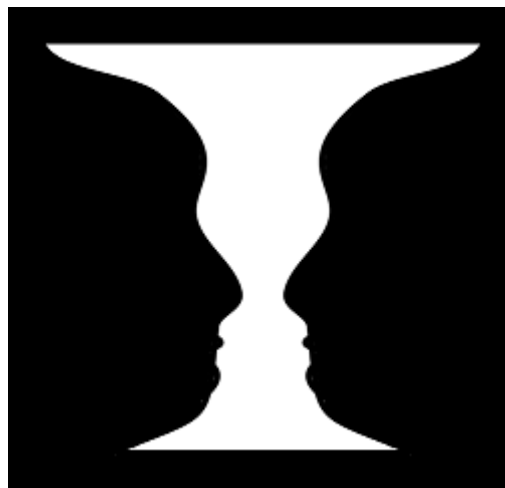
1

Physical computing

1.1 Zintuigen versus sensoren

Levende organismen, of het nu eencelligen of mensen zijn, nemen hun omgeving waar met behulp van zintuigen. Computersystemen zijn daarvoor uitgerust met sensoren. Er is een grote verscheidenheid aan sensoren beschikbaar, en sommige kunnen zelfs meer waarnemen dan de zintuigen van een mens of dier, zoals infrarood- of röntgenstraling. De meeste sensoren zijn echter beperkter dan onze zintuigen en soms zijn er verschillende soorten sensoren nodig om de werking van één van onze zintuigen te evenaren.

Computer sensoren werken om vele redenen meestal niet op identieke wijze als onze zintuigen; De afhandeling van ruwe data van een geluidssensor kan leiden tot de detectie van geheel andere verschijnselen dan mensen met hun oren zouden kunnen horen. Bovendien worden onze waarnemingen sterk beïnvloed door de associaties die ons brein aan deze waarnemingen koppelt. Het bekende “gezichtsbedrog” is daar bijvoorbeeld een gevolg van.



Afbeelding 1: De vaas van Rubin, figuur of achtergrond?

Twee sensoren met dezelfde functie kunnen onderling verschillen doordat tijdens het fabriceren een afwijking is ontstaan. Deze verschillen maken het noodzakelijk om de sensoren op elkaar af te stemmen. Als dit gebeurt naar referentie van een standaardwaarde dan noemen we dit een **ijking**. Als een sensor wordt afgeregeld om data waarden af te beelden op een bepaalde schaal (bijvoorbeeld van 0 tot 100) wordt dit **kalibreren** genoemd. Ook het aanpassen van niet-lineair gedrag van een

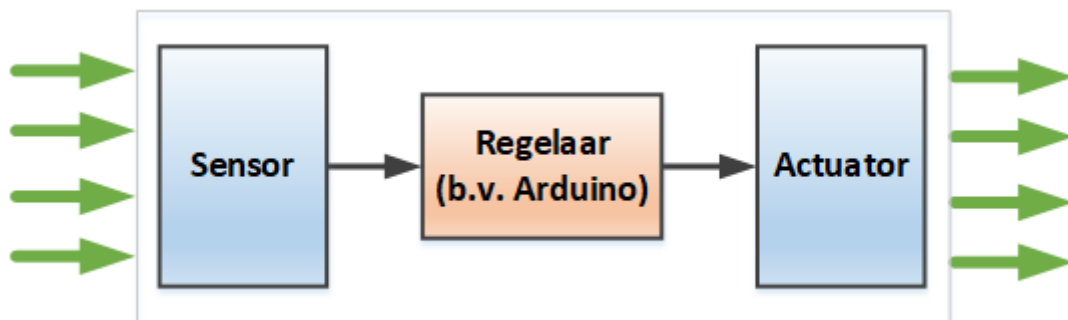
sensor naar lineair gedrag wordt kalibreren genoemd. Het kalibreren maakt het vergelijken van de output waarden van verschillende sensoren eenvoudiger.

Opdracht 1

Een mobiele telefoon heeft allerlei sensoren, bijvoorbeeld een microfoon, gyroscoop, of touchscreen. Deze sensoren nemen iets waar in de fysieke wereld. Bijvoorbeeld, een microfoon neemt geluidsgolven waar.

1. Welke sensoren zitten in moderne smartphones?
2. Geef bij elk van de sensoren aan welk fysiek verschijnsel wordt waargenomen en met welke nauwkeurigheid deze sensoren werken. Neem je eigen telefoon als uitgangspunt.

In physical computing wordt dus de echte (fysieke) wereld waargenomen met **sensoren**. Door middel van computersystemen wordt vervolgens bepaald om met **actuatoren** aanpassingen te maken.



Afbeelding 2: Sensoren en actuatoren

Opdracht 2

Een stofzuigerrobot maakt je kamer schoon terwijl je zelf op school zit of aan het werk bent. Er zijn diverse stofzuigerrobots verkrijgbaar.

In [dit filmpje](#) zie je een korte review van de iRobot Roomba 880.

Deze stofzuigerrobot is in staat te bepalen of er een object in de weg staat. Als de

stofzuiger naar beneden dreigt te vallen, bijvoorbeeld van de trap, draait hij zich om. Hij laadt zichzelf op bij het basisstation.

Beschrijf de sensoren en actuatoren van deze stofzuigerrobot.

1. Welke sensoren zijn hierbij betrokken?
2. Beschrijf de mogelijke functies van de robot in zoveel mogelijke onderdelen.

Vul het lijstje hieronder daarvoor aan:

- beweeg de robot vooruit
- detecteer obstakel
- etc ...

Voor de Arduino zijn er heel veel verschillende sensoren beschikbaar.

Hoe je ze kunt gebruiken is meestal wel even zoeken.

Als je ze bij gerenommeerde bedrijven koopt (bijvoorbeeld [Adafruit](#) of [Sparkfun](#)) krijg je de informatie over bijvoorbeeld hoe je hem moet aansluiten en welke code je moet gebruiken er vaak wel bij. Maar soms zijn de sensoren van Adafruit en Sparkfun net even anders en dan heb je bijvoorbeeld een ander getal (adres om de sensor te vinden) in jouw code nodig.

Als je ze rechtstreeks in China koopt zul je meer moeite moeten doen om de juiste code te vinden..

Je krijgt dan vaak een klein plastic zakje met een zeer onduidelijke sticker erop (afbeelding 3). En verder staan er alleen maar wat codes op de sensor geprint.



Afbeelding 3: Een sensor. Van links naar rechts: de verpakking, de achterkant en de aansluitkant

Opdracht 3

- Op de sensor van afbeelding 12 is de code CJMCU-811. Zoek op wat voor sensor dit is. De eigenlijke chip waar het hier om gaat is de CSS811 maar dat is wel wat moeilijk te lezen.
- Zoek op waarvoor de aansluitingen op de rechterfoto bedoeld zijn
- Kijk of je arduino code voor deze sensor kunt vinden

Voor de code zul je vermoedelijk een zogenaamde library nodig hebben. Een library is een stukje kant en klare code die je in jouw programma voor het laten werken van de sensor nodig hebt.

Hoe dat werkt met zo'n library? Kijk eens [hier](#).

1.2 Enkele veel gebruikte sensoren

We kennen een aantal sensoren die bijvoorbeeld beweging, warmte of aanraking kunnen waarnemen, zoals:

PIR sensor (Passive Infrared Sensor)



Afbeelding 4

De sensor reageert op beweging en doet dat met behulp van een infrarood sensor en een lens (het witte bolletje is de lens, de sensor zit daaronder).

Obstacle Avoidance Sensor



Afbeelding 5

Ook dit is een infrarood sensor. Hij zendt infrarood licht uit. Als die lichtstraal tegen een voorwerp of een persoon aan komt wordt deze teruggekaatst en opgevangen door de sensor.

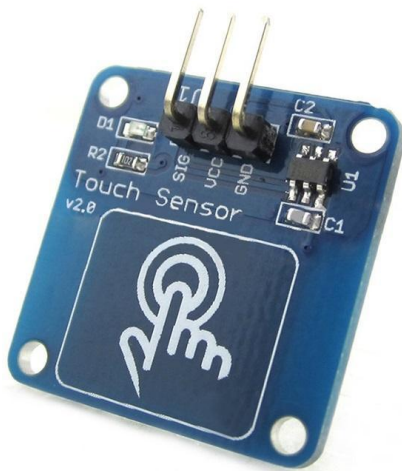
Ultrasonic Module Distance Measuring Transducer Sensor



Afbeelding 6

Deze sensor zendt ultrasoon geluid uit en ontvangt dat ook weer. Daaruit bepaalt hij de tijd tussen het moment van uitzenden en het weer ontvangen van dat geluid en dat is een maatstaf voor de afstand tussen de sensor en het voorwerp, een (vloei-) stof of persoon.

Capacitive Touch Switch Sensor



Afbeelding 7

Dit is een zogenaamde aanraaksensor. Hij geeft een signaal af (beter gezegd, een aansluiting wordt HIGH) als je hem aanraakt.

HIGH betekent in dit geval dat er op de aansluiting SIG een spanning komt te staan van ongeveer 5 volt. Als je hem niet aanraakt is de spanning op de SIG aansluiting ongeveer 0 volt.

Opdracht 4

- Bekijk [dit filmpje](#) eens om te zien wat je allemaal met een ultrasone sensor kunt doen.

Opdracht 5

Een datasheet van de ultrasone sensor vind je [hier](#). Maar [deze](#) of [deze](#) geven de informatie misschien net iets duidelijker weer. Kun je de volgende informatie vinden? Zo ja, noteer die.

1. Wat is de minimum en maximum afstand die je kunt meten?
2. Hoe breed is de bundel (graden van de hoek) van de meting?
3. Is de temperatuur van invloed?
4. Wat zijn de aansluitingen?
5. Op welke spanning moet je de sensor aansluiten?
6. Met welke spanning werken de logische signalen (kijk bij TTL level, informatie daarover vind je [hier](#))?

Opdracht 6

Zoek ook van de overige sensoren die hier genoemd worden de datasheets. En misschien zijn er wel meerdere varianten van te vinden waardoor het gemakkelijker is om onderstaande vragen te beantwoorden.

Onderzoek dus met behulp van de datasheets (noteer de links ervan) van al deze sensoren per sensor de volgende punten (indien van toepassing):

1. Wat is de minimum en maximum afstand die je kunt meten?
2. Hoe breed is de bundel (graden van de hoek) van de meting?
3. Is de temperatuur van invloed?
4. Wat zijn de aansluitingen?
5. Op welke spanning moet je de sensor aansluiten en met welke spanning werken de logische signalen?

Opdracht 7

Veronderstel dat je een (speelgoed) zelfrijdende auto wilt bouwen. Uiteraard mag die, als hij loopt (of rijdt) nergens tegenaan botsen. En hij mag ook niet van de tafel vallen. De hierboven genoemde sensoren hebben diverse mogelijkheden objecten.

1. Beredeneer welke sensor(en) jou daar het geschiktst voor lijkt.
2. Bedenk ook hoe je die sensoren gaat plaatsen op of aan de auto.

2

Arduino (Genuino)

2.1 Wat is de Arduino?

De **Arduino** is een minicomputer.

Om ongetwijfeld goede maar ook duistere redenen heet tegenwoordig de Arduino zoals die in landen buiten de Verenigde staten verkocht wordt, de Genuino.

Maar in deze reader zullen we gewoon spreken over de Arduino.

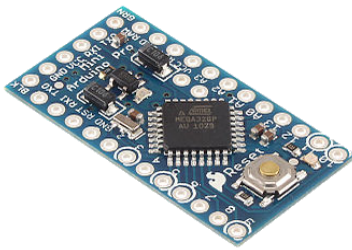


Afbeelding 8: De Arduino Uno

De Arduino kennen we in diverse varianten. In afbeelding 8 zie je de **Arduino Uno**. Deze zie je het meest.

Maar heb je meer aansluitingen nodig dan kun je bijvoorbeeld een Arduino Mega gebruiken.

Het ontwerp van de Arduino is **open source** en dat betekent dat iedereen ze na mag maken. Het gevolg is dus dat je veel klonen ziet. Daar staat dan niet de naam Arduino op maar wel vaak Uno en die is dan compatible met de Arduino Uno. Om ze zo goedkoop mogelijk te maken worden vaak andere chips gebruikt zoals de CH340 chip om via de USB aansluiting met een computer te kunnen communiceren. Dat werkt dan prima maar je hebt er wel een andere driver voor nodig. Even opletten dus als je een Arduino wilt kopen.



De Arduino Uno is handig omdat je, in ieder geval als je gebruik maakt van een **breadboard**, er helemaal niet bij hoeft te solderen om er wat op aan te sluiten. Er zijn ook kleinere Arduino varianten (zoals in afbeelding 9) en dat is handig om ergens in te bouwen maar dan moet je de aansluitingen vaak wel solderen.

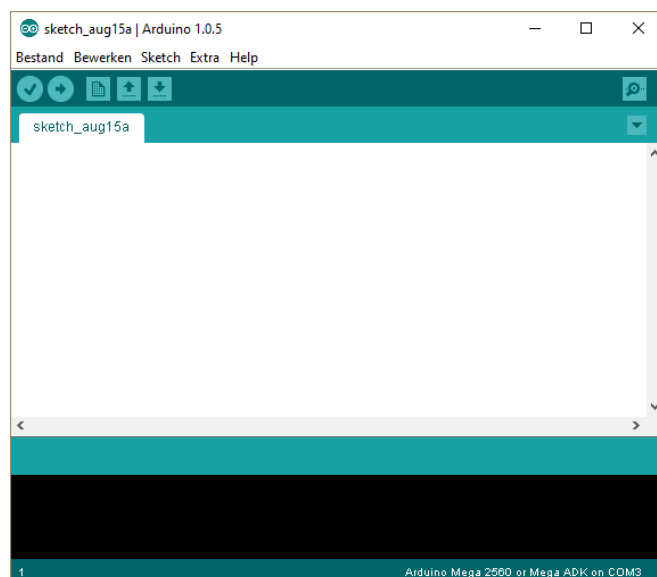
Afbeelding 9: De Arduino Pro mini

2.2 Waar is het beeldscherm?

Zoals je ziet is de Arduino een computer zonder toetsenbord, beeldscherm of muis.

Om die reden sluiten we de Arduino aan op een gewone computer om hem te programmeren. Er is speciale software om de Arduino te programmeren.

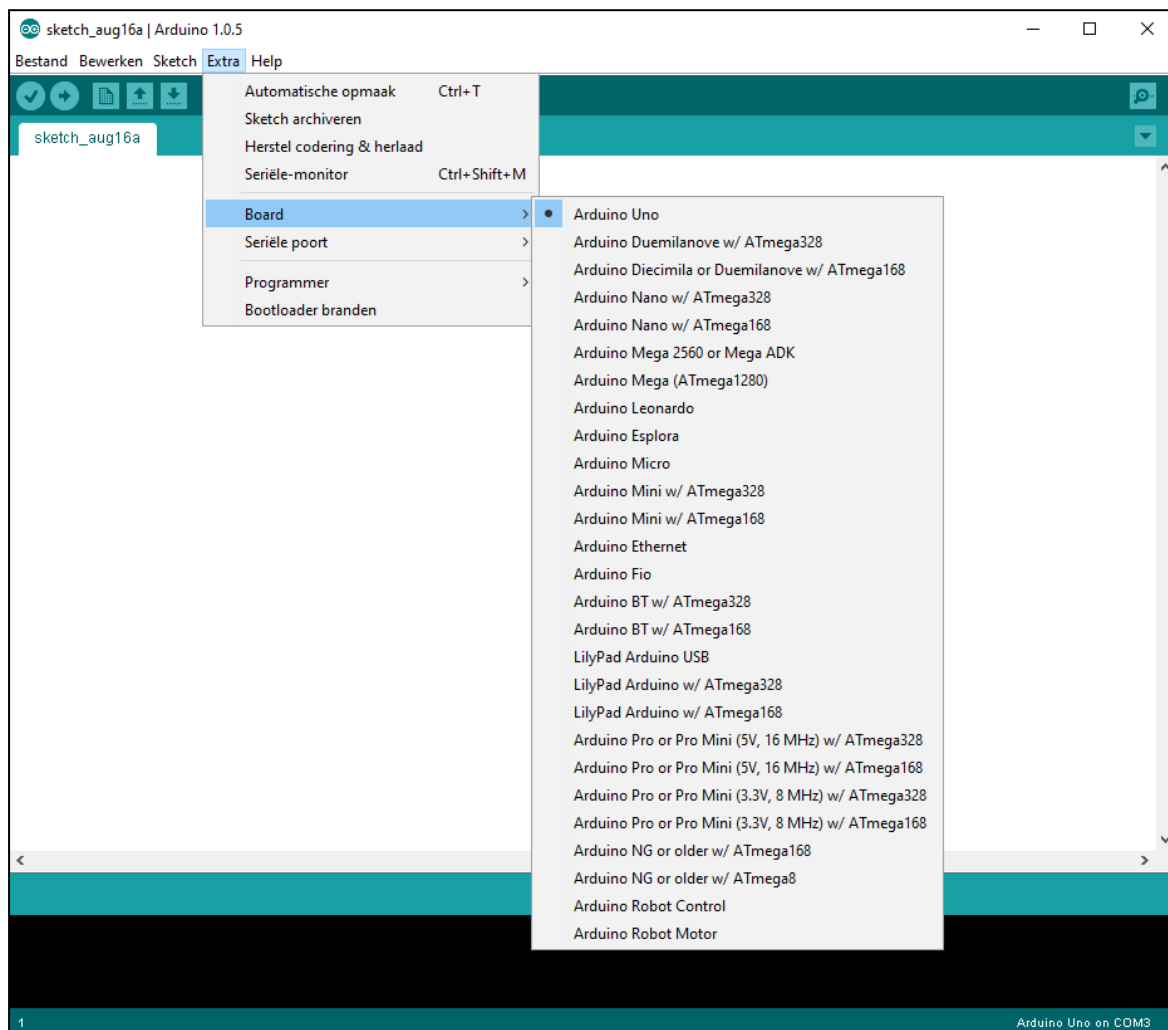
Mogelijk staat die software al op jouw computer en anders kun je hem downloaden vanaf de website www.arduino.cc



Afbeelding 10: De Arduino IDE

2.3 Aansluiten

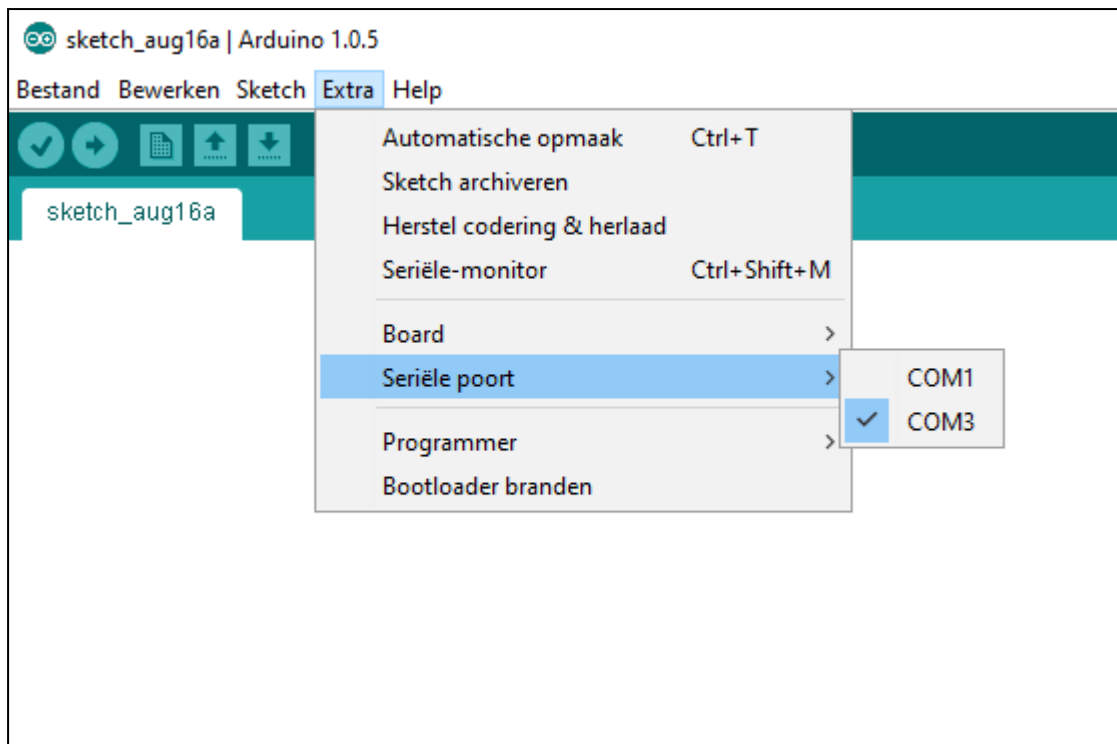
Als je de Arduino op de computer aansluit moet je goed opletten. In de software moet je namelijk de juiste Arduino aangeven en ook moet de poort, waarop de Arduino is aangesloten kloppen.



Afbeelding 11: Het kiezen van het juiste board

Rechtsonder in de IDE zie je waarop de software nu staat ingesteld. In afbeelding 9 is dat **Arduino Uno on COM3**.

De juiste poort zie je pas als de Arduino is aangesloten.



Afbeelding 12: Het controleren en eventueel aanpassen van de poort in de software

Ga er maar vanuit dat de Arduino vrijwel altijd op de hoogste poort zit.

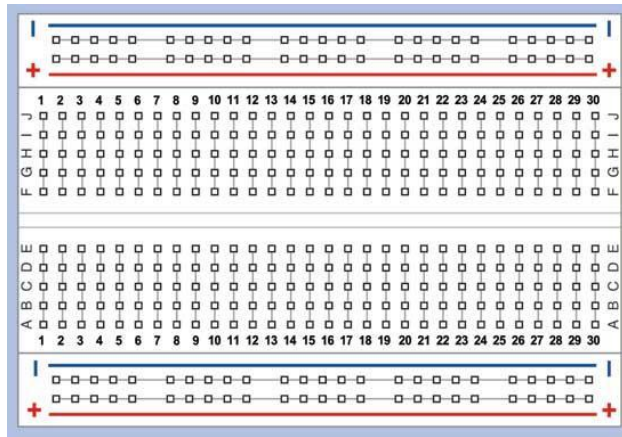
Als de poort niet verschijnt kan je een aantal dingen doen:

1. Sluit de Arduino opnieuw aan.
2. Herstart de Arduino software.
3. Gebruik een andere usb kabel.
4. Start je computer opnieuw.

2.4 Het breadboard

Je hebt natuurlijk niet een Arduino om er alleen maar naar te kijken. Je wilt er ook “dingen” op aan kunnen sluiten. Die “dingen” kunnen van alles zijn: lampjes, drukknopjes, sensoren enz.

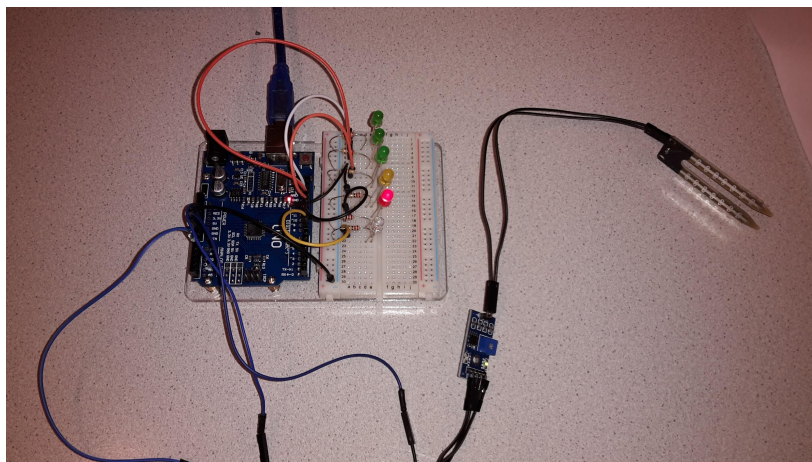
Zeker als we het hebben over een proefopstelling is het erg gemakkelijk om daar een zogenaamd **breadboard** voor te gebruiken.



Afbeelding 131: Een breadboard

Een breadboard is dus een soort van bord met gaatjes waar je draadjes en andere onderdelen in kunt steken en zo verbindingen kunt maken.

In afbeelding 12 zie je hoe je zo'n breadboard kunt gebruiken.



Afbeelding 14: Aansluitingen tussen de Arduino en het breadboard

Meer over breadboards vind je [hier](#).

Een leuk filmpje over het gebruik van een breadboard samen met een Arduino vind je [hier](#).

Opdracht 8

- Sluit nu de Arduino aan op de computer
- Open de Arduino IDE en stel die op de juiste manier in (juiste Arduino en juiste poort)

- Kies nu via Bestand > Voorbeelden > 01.Basics > Blink

Er opent zich nu een nieuw venster met daarin het programma Blink.

- Ga dit programma met  uploaden naar de Arduino en bekijk het resultaat (er gaat op de Arduino een ledje knipperen)

Zo, dat was het begin.

Je hebt een programma, kunt dat uploaden en je ziet het resultaat.

Nu tijd om te leren de Arduino te programmeren en er op de juiste manier onderdelen aan te koppelen.

Maar eerst nog even een belangrijk onderdeel: het **toestandsdiagram**.

2.5 Automaten

Je kent ze allemaal wel, die deuren bij openbare gebouwen. Als je er voor staat gaat de deur open en kun je naar binnen. Maar de tweede deur gaat pas open als de eerste deur weer dicht is.

Een dergelijke deur zouden we wel kunnen programmeren maar is best nog wel ingewikkeld. Dus laten we eerst wat eenvoudiger beginnen.

Veronderstel dat je voor een deur komt te staan die automatisch open gaat. Jij staat buiten en je wilt naar binnen.

Zoiets is redelijk eenvoudig te maken met behulp van een kleine computer en enkele sensoren maar het is wel aan te raden hier eerst een model van te maken.

Dat kunnen we doen door het tekenen van een toestandsdiagram.

Een toestandsdiagram is een manier waarmee we de verschillende toestanden die zich voor kunnen doen gaan tekenen.

Simpel voorgesteld is dat bij zo'n automatische deur:

- A. de deur is dicht**
- B. de deur is open**

Dit is overigens wel erg simpel voorgesteld want er zijn wel meer toestanden te bedenken. Bijvoorbeeld:

A. de deur gaat dicht

B. de deur gaat open

Maar bij een toestandsdiagram noemen we dat geen **toestanden (state)** maar een **gebeurtenis (event)**.

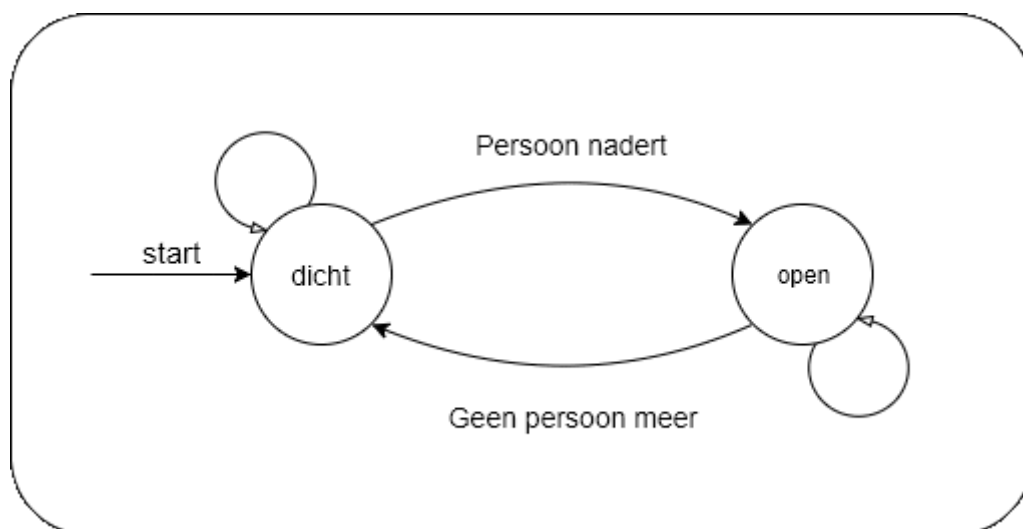
En die events geven aan hoe je van de ene toestand in de andere komt. Dit noemen we ook wel de **transitie**.

De toestanden laten we zien door middel van een **cirkel**.

Tussen de toestanden tekenen we **pijlen** die de transities aangeven.

Een pijl kan ook naar de toestand zelf verwijzen. In dat geval vindt er geen transitie plaats.

Ingewikkelde woorden? Kijk maar eens naar afbeelding 13.



Afbeelding 15: een toestandsdiagram

We beginnen bij start.

Je komt dan eerst bij de toestand **dicht**. Dus de deur is dicht. Er is niemand bij de deur.

Een toestandsdiagram werkt met een klokpuls of een loop. Dat betekent dat hij tijdens iedere klokpuls gaat kijken of er iets veranderd is. Als er niets veranderd is blijft de toestand dicht. Hij gaat dan als het ware langs de ronde pijl en komt weer bij zichzelf uit.

Als er wel iets veranderd is, dus als er een persoon voor de deur staat, vindt er een

transitie plaats naar de toestand **open** (de kromme pijl van dicht naar open). Zolang de sensoren een persoon “voelen” blijft de toestand open. Als er niemand meer is gaat hij van de toestand open weer over de pijl naar de toestand dicht.

Uiteraard is dit een heel eenvoudig voorbeeld. Maar we gaan, bij het programmeren van de Arduino, ook wel wat ingewikkeldere automaten tekenen.

Als we zo’n toestandsdiagram vertalen naar code dan kun je ruwweg zeggen dat iedere pijl een `if` opdracht is. We gaan dat bij de Arduino opdrachten verder uit werken.

2.5.1 Regels voor een toestandsdiagram

- Een toestandsdiagram bestaat uit toestanden (cirkels) en toestandsovergangen (pijlen).
- Alle toestanden worden weergegeven als cirkels. In de cirkel staat een korte beschrijving van de toestand.
- Er is altijd precies één start toestand, die herken je aan de pijl die niet verbonden is aan een andere toestand.
- Alle toestandsovergangen worden weergegeven als een pijl, die één kant op wijst. Bij de pijl staat een korte beschrijving van de toestandsovergang.
- Een toestandsovergang gaat altijd van precies één toestand naar precies één andere toestand.
- Een toestandsovergang mag dezelfde toestand als begin en eind hebben.
- Bij een toestandsovergang mogen één of meerdere acties staan.
- Eerder zeiden we al: Een toestandsovergang wordt uitgevoerd op basis van een event: informatie uit de sensoren en soms ook op basis van een klok of timer.
- Voor het uitvoeren van acties worden vaak actuatoren gebruikt, zoals een lampje, display of motor

3

Programmeren

3.1 De programmastructuur

Een Arduino programma noemt men ook een sketch. Een sketch heeft altijd dezelfde structuur. En die is erg simpel. Hij bestaat uit minstens twee gedeelten of functies. Deze twee functies zijn `setup()` en `loop()`. Het ziet er dan zo uit:

```
void setup() {  
    // Hier komen opdrachten te staan die te maken hebben met  
    // de voorbereiding  
    // Bijvoorbeeld hoe de poorten (aansluitingen) moeten werken  
}  
  
void loop() {  
    // En hier komen opdrachten te staan die continu herhaald  
    // worden zoals kijken of er al een knop wordt ingedrukt of  
    // het meten van de temperatuur;  
}
```

Beide functies zijn nodig om een Arduino programma te maken.

`setup()` wordt als eerste uitgevoerd en daarna wordt `loop()` telkens herhaald.

`setup()` wordt gebruikt om variabelen te declareren en er wordt bepaald welke pinnen ingang of uitgang worden.

`loop()` kan bijvoorbeeld gebruikt worden om de invoerpoorten te “lezen” en dan met die informatie te beslissen wat er met de uitvoerpoorten moeten gebeuren.

Je kan `loop()` eigenlijk zien als de motor van een Arduino programma, dus waar al het werk moet gebeuren.

Het woordje `void` voor deze functies betekent dat je deze functies wel aan kunt roepen maar dat je geen waarde of iets dergelijks terug krijgt. Dit in tegenstelling tot bijvoorbeeld een functie die iets moet berekenen. Dan verwacht je de uitkomst van de berekening terug.

Hieronder de code van het programma Blink dat je in opdracht 8 al gebruikt hebt.

Maar nu met de commentaarregels in het Nederlands.

Kijk maar even of je het kunt volgen:

Opdracht 9

Bestudeer de onderstaande code:

- Probeer het programma helemaal te begrijpen
- Wijzig nu de tijd dat het LEDje aan is naar 3 seconden

```
/*
  Blink
  Dit programma zet steeds gedurende 1 seconde een LED aan en daarna weer
  een seconde uit.

  De meeste Arduinos hebben op het board een LED die je aan en uit kunt
  zetten. Op de Uno en de Leonardo zit die gekoppeld aan de digitale pin 13.
  Als je niet zeker weet met welke pin de on-board LED op jouw Arduino is
  gekoppeld, lees dan de documentatie op http://arduino.cc

  Deze code is in het publieke domein. D.w.z. vrij te gebruiken.

  Gewijzigd op 8 May 2014 door Scott Fitzgerald
*/

// De setup functie loopt eenmalig als je op reset drukt
// of als je de Arduino aan zet.
void setup() {
  pinMode(13, OUTPUT); // initialiseer de digitale pin 13 voor uitvoer
}

// De loop functie wordt regel voor regel doorlopen en
// begint daarna weer van voren af aan.
void loop() {
  digitalWrite(13, HIGH); // Zet spanning op poort 13 op hoog
  delay(1000);            // Wacht een seconde (1000 msec)
  digitalWrite(13, LOW);  // Zet de spanning op poort 13 op laag
  delay(1000);            // Wacht weer een seconde
}
```

Omdat IO-pinnen zowel invoer (INPUT) als uitvoer (OUTPUT) kunnen zijn moeten we dus eerst in de `setup()` functie aangeven dat pin 13 als uitvoer wordt gebruikt:

```
pinMode(13, OUTPUT);
```

In de `loop()` functie kunnen we het LED laten branden door pin 13 hoog (**HIGH**) te maken:

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

Daarna moeten we een seconde (1000 msec) wachten:

```
delay(1000);
```

Vervolgens kunnen we de LED weer uitzetten door pin 13 laag (**LOW**) te maken:

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

Naast deze functies kunnen nog meer zaken aan de orde komen zoals het declareren van variabelen en het inlezen van zogenaamde libraries. Ook kun je zelf functies aanmaken die je dan aanroept vanuit de `loop()` functie.

3.2 De programmeertaal

De programmeertaal zoals die bij de Arduino gebruikt wordt is een afgeleide van de **programmeertaal C**. Als je al enige programmeerervaring hebt dan zul je de meeste elementen wel herkennen.

Elementen zoals de voorwaarden (`if ...`, `if ... else ...`) en de loops (`for ...`, `while ...`, `do ... while`, `switch case`).

Maar al doende leert men dus ga, als je wat wilt programmeren, op zoek naar de juiste code.

Er is heel veel te vinden op het internet, ook al is het vaak niet fraaie code; zorg er dus voor dat, als je programmeert, je vanaf het begin nette en begrijpbare code maakt.

Teken ook altijd eerst een toestandsdiagram voor je gaat programmeren.

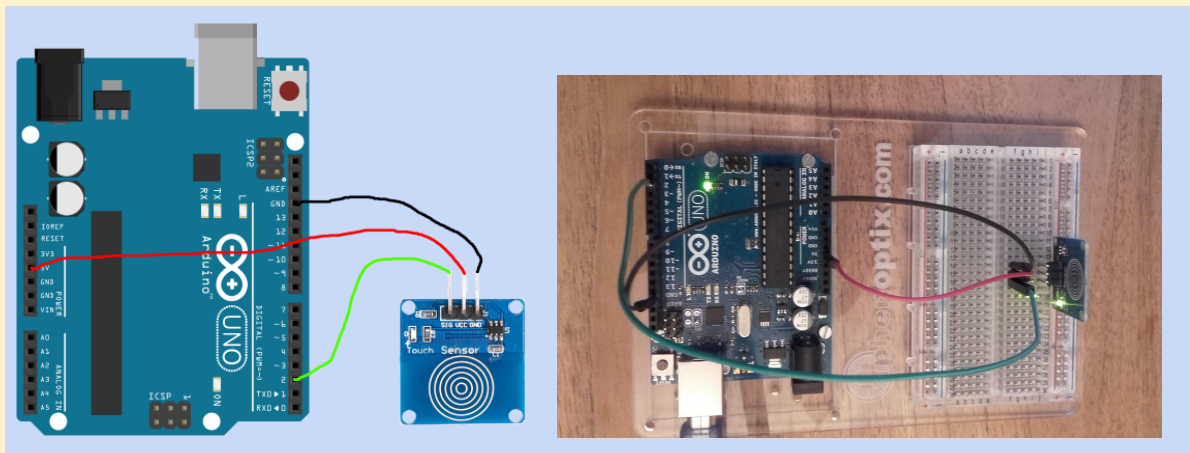
De belangrijkste bronnen vind je op de website van Arduino, bijvoorbeeld [hier](#). En [hier](#) vind je tutorials.

3.3 Aan de slag

Laten we kijken hoe je verschillende onderdelen kan gebruiken met je Arduino.

Opdracht 10

- Sluit de aanraakschakelaar aan volgens het schema hieronder. Doe dat met een breadboard waar je de aanraakschakelaar in prikt. Ook de draadjes prik je in in het breadboard.



- Via de rode draad sluiten we de VCC (volts) pen van de aanraakschakelaar aan op 5 volt van de Arduino.
- De GND (ground) pen van de aanraakschakelaar komt aan de gnd van de Arduino.
- De groene draad (SIG/signal) sluiten we aan op poort 2 van de Arduino.

De aanraakschakelaar (**Capacitive Touch Switch Sensor**) is een simpele sensor. Je raakt hem aan en hij reageert (maakt de SIG-poort HIGH) en zodra je hem niet meer aanraakt gaat hij weer terug naar de oorspronkelijke toestand (de SIG-poort wordt dan weer LOW).



Let op: er zijn ook aanraakschakelaars die een zogenaamde toggle functie hebben. Als bij een dergelijke schakelaar de uitgang LOW is wordt hij na aanraken HIGH en dat blijft hij tot je hem weer aanraakt. Dus de schakelaar houdt de toestand vast als je hem niet meer aanraakt. Dat kan handig zijn, maar de functies zijn daardoor wel beperkter.

We gaan de aanraakschakelaar gebruiken om een ledje aan- en uit te zetten. We gebruiken hiervoor het ledje dat al op de Arduino zit dus vandaar dat we alleen maar de aanraakschakelaar hoeven aan te sluiten.

Maar hoe kunnen we die aanraakschakelaar gebruiken?

In de volgende opdrachten gaan we er mee aan de slag. Maar eerst tekenen we de toestandsdiagram om helemaal duidelijk te maken wat we eigenlijk willen programmeren.

We hebben hierbij te maken met twee toestanden:

A. ledAan

B. ledUit

En twee transities:

A. aangeraakt

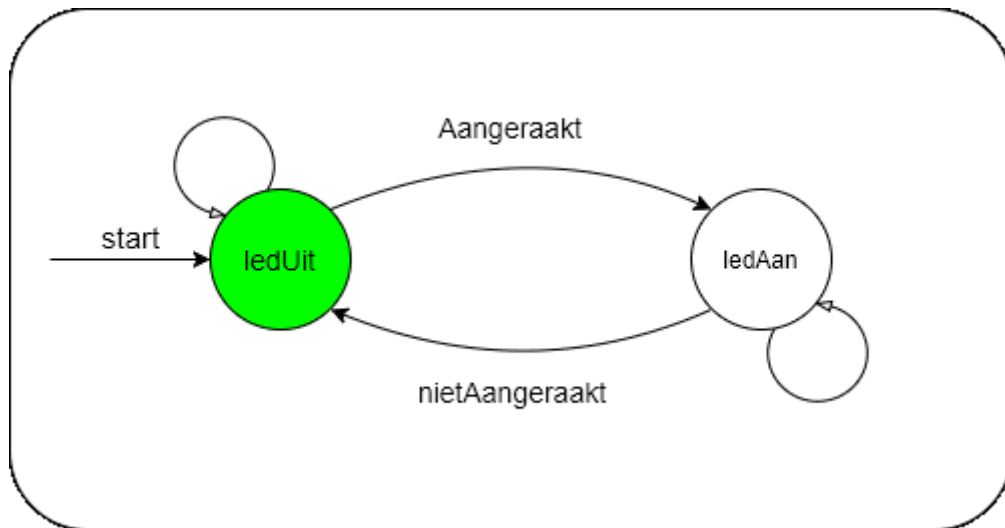
B. nietAangeraakt

Bij **nietAangeraakt** vanuit toestand **ledUit** blijft de led uit.

Bij **aangeraakt** vanuit toestand **ledAan** blijft de led aan.

Het toestandsdiagram ziet er dan uit zoals in afbeelding 14.

Ga na of deze klopt.



Afbeelding 16: een toestandsdiagram voor een aanraakschakelaar

Opdracht II

Het toestandsdiagram die we in afbeelding 14 getekend hebben gaan we nu gebruiken om de code te schrijven.

Maar omdat het de eerste keer is krijg je de code er hier bij.

- Voer nu de volgende code in:

```

//*****//
//                                                    //
// Het programmeren van een aanraaksensor 1           //
//                                                    //
// Schakelt de ingebouwde led op de Arduino aan en uit //
// met behulp van een capacitieve aanraakschakelaar  //
//                                                    //
//*****//

// *****
// Pin toewijzing
// *****
//
// Het ledje op de Arduino
const int ledPin = 13;
// De aansluiting van de sensor

```

```

const int sensorPin = 2;

// *****
// Variabelen
// *****
//
// De variabele aanraken geeft aan of de sensor aangeraakt is.
// Is de waarde 0 dan wordt hij niet aangeraakt.
// Is de waarde 1 dan wordt de sensor aangeraakt.
int aanraken = 0;
//
// In het toestandsdiagram zien we 2 toestanden:
//
// 1 Het ledje is uit
// 2 Het ledje is aan
//
// We beginnen bij toestand 1
int toestand = 1;

void setup() {
    // Initialiseert de sensorPin als een input.
    // Daarmee geven we aan dat de sensorPin (pin 2) gebruikt
    // wordt om een signaal aan de computer door te geven.
    pinMode(sensorPin, INPUT);
    // Initialiseert de ledPin als een output.
    // Daarmee geven we aan dat de ledPin (pin 13) gebruikt
    // wordt om een signaal vanuit de computer aan
    // een ander onderdeel door te geven.
    pinMode(ledPin, OUTPUT);
}

void loop() {
    // Lees de waarde die de aanraakschakelaar afgeeft:
    aanraken = digitalRead(sensorPin);

    if (toestand == 1) {
        if (aanraken == HIGH) {
            // Ga nu naar toestand 2
            toestand = 2;
            // en zet het ledje aan
            digitalWrite(ledPin, HIGH);
        }
    }

    if (toestand == 2) {

```

```

    if (aanraken == LOW) {
        // Ga nu naar toestand 1
        toestand = 1;
        // en zet het ledje uit
        digitalWrite(ledPin, LOW);
    }
}
}

```

- Sla het bestand op
- Upload het naar de Arduino
- Bekijk het resultaat

Opmerking:

Op alle plaatsen in de code waar **HIGH** of **LOW** gebruikt is mag je daarvoor in de plaats ook **1** of **0** gebruiken.

Hopelijk zie je de relatie tussen de toestandsdiagram van afbeelding 14 en de code.

Opdracht 12

Voor deze opdracht gebruik je dezelfde schakeling want met deze serie opdrachten is het juist de bedoeling dat je ziet dat je met de computer kunt bepalen hoe er op de sensor gereageerd wordt.

- Het is de bedoeling dat je hier in feite het programma dusdanig aanpast dat het ledje brandt als je de sensor NIET aanraakt en dat het ledje uitgaat als je hem WEL aanraakt.
Teken daarvoor eerst het toestandsdiagram.
- Pas nu de code aan.

Wordt het al wat duidelijker?

Vooraf het tekenen van het toestandsdiagram is erg belangrijk. Op die manier wordt het schrijven van het programma eigenlijk een soort van stappenplan voor de computer. En dat kunnen we wel gebruiken als de opdrachten ingewikkelder worden.

Bij de volgende opdracht proberen we de code dusdanig te wijzigen dat het ledje bij de eerste keer aanraken aangaat en bij een volgende keer aanraken weer uit gaat. We programmeren dus de toggle functie.

Maar uiteraard beginnen we met het toestandsdiagram.

Dit is nogal wat lastiger want het gaat niet alleen om het aanraken maar ook om het loslaten.

Want als je de sensor aanraakt, moet het ledje aan gaan. En als je dan de sensor weer loslaat moet hij aanblijven.

Het probleem hier is dat tussen het aanraken van de sensor en het loslaten meerdere klokpulsen en dus meerdere loops zijn verstreken. En tijdens die klokpulsen blijft die sensor aangeraakt.

Maar om het ledje uit te laten gaan heb je ook een aanraking nodig. Kun je je voorstellen dat dat voor de computer verwarrend is?

Uiteraard niet wanneer wij hem gaan programmeren.

Dus je raakt hem aan en dan gaat het ledje aan.

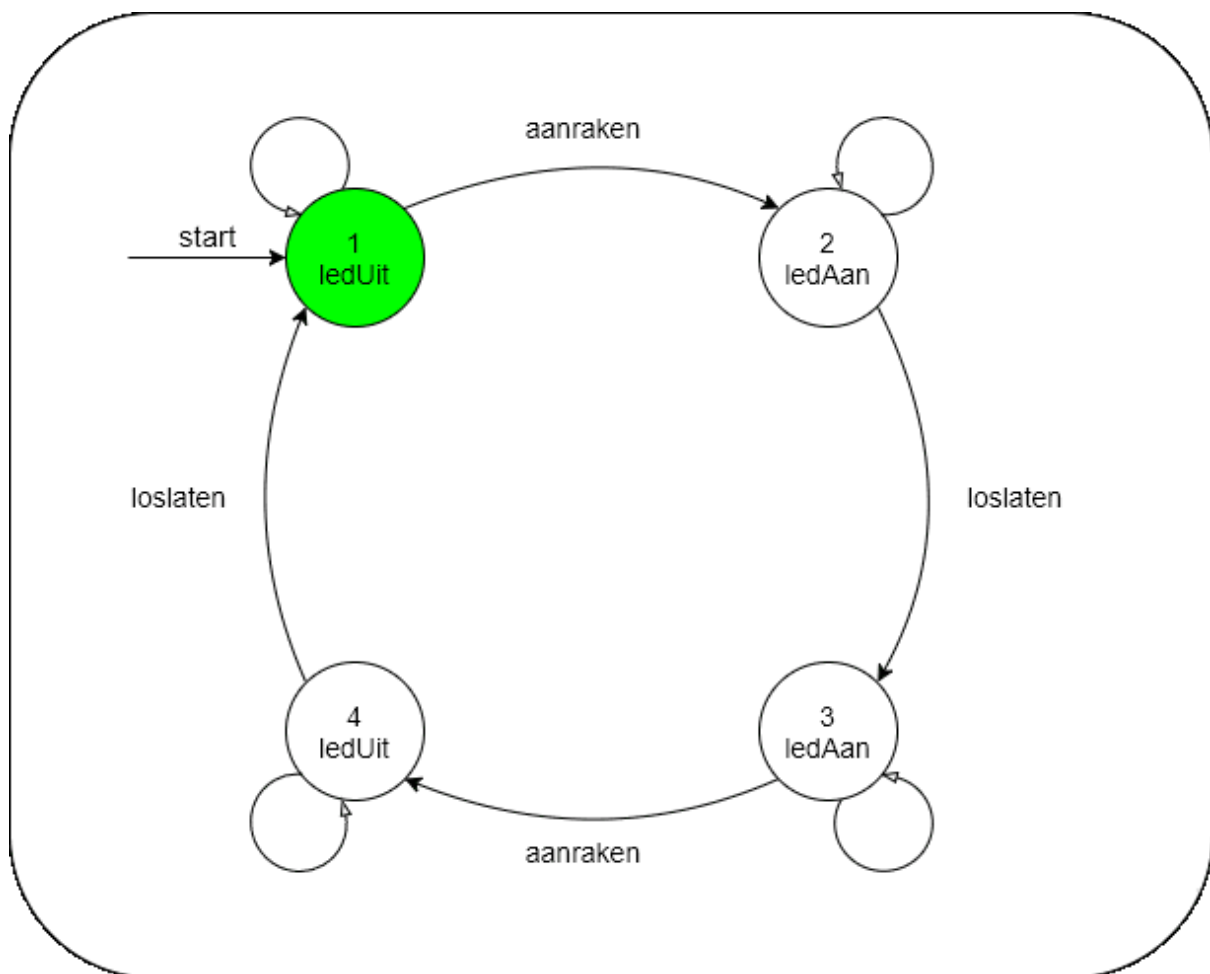
Daarna laat je de sensor weer los.

Met het ledje gebeurt dan niets maar dat is voor de computer het signaal dat bij de volgende aanraking het ledje weer uit mag gaan.

We hebben dus 4 toestanden:

1. Het ledje is uit bij start
2. Het ledje gaat aan na een aanraking
3. Het ledje blijft aan na het weer loslaten
4. Het ledje gaat uit na een aanraking

Het toestandsdiagram ziet er dan als volgt uit:



Afbeelding 17: een toestandsdiagram voor een toggle functie met een aanraakschakelaar

Kun je hem volgen?

Nu de kunst om hiervan code te maken.

Onthoud: iedere pijl is een if in de betreffende toestand.

Opdracht 13

We gebruiken dus weer dezelfde schakeling

- Programmeer nu de code zoals in afbeelding 15 met het toestandsdiagram wordt aangegeven.

Opdracht 14

- Veronderstel dat je niet de beschikking hebt over een aanraaksensor maar wel een PIR sensor. Moet je dan ook iets aan de code veranderen als je die gaat gebruiken?
- Denk je dat de schakeling dan anders gaat reageren? Zo ja, op welke manier merk je dat?

3.4 Aansluiten

Een standaard LED op de Arduino laten knipperen is nog iets anders dan er allerlei onderdelen aan te koppelen.

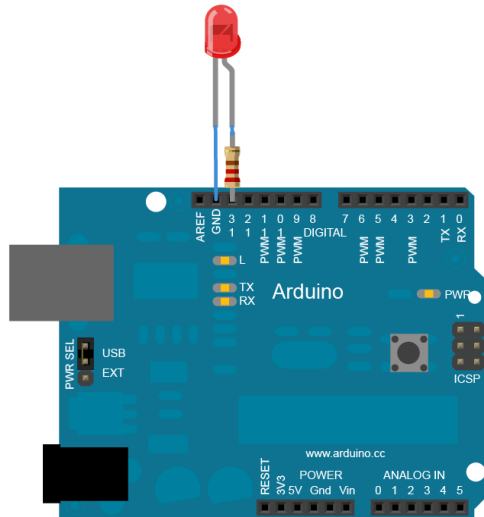
Dus daar moeten we nog wel even mee bezig.

3.4.1 Een LED aansluiten

Om te beginnen het aansluiten van een **LED** (Light Emitting Diode) op de Arduino. Een led heeft twee aansluitingen; de kathode (negatief) en de anode (positief). Zorg ervoor dat de **langste** poot (anode) altijd aan de **plus** van de spanning komt.

De meeste Arduino's werken met een spanning van 5 volt. En voor de meeste LEDs is dat een te hoge spanning. Maar dat kun je oplossen door in serie met een LED een **weerstand** te plaatsen.

Meestal is een weerstand tussen 200 Ω en 500 Ω een goede waarde. Begin met 500 Ω en als je het niet fel genoeg vindt kun je eventueel de weerstand iets verlagen.



Afbeelding 18: LED koppelen aan Arduino met weerstand

3.4.2 Weerstanden

Een weerstand zorgt ervoor dat de stroom die er doorheen gaat, afhankelijk van de weerstandswaarde, getemperd wordt. Of beter gezegd, dat de energie omgezet wordt in warmte.

De waarde van een weerstand kun je aflezen aan de kleuren die erop staan:

-2	-1	0	1	2	3	4	5	6	7	8	9
----	----	---	---	---	---	---	---	---	---	---	---

$([1e \text{ kleur}] \times 10 + [2e \text{ kleur}]) \times 10^{[3e \text{ kleur}]} = [\text{weerstand}]$

zwart, groen, rood, goud is dus: $5 \times 10^2 = \underline{500 \Omega}$. $((0 \times 10 + 5) \times 10^2)$

3.4.3 Een schakelaar aansluiten

Als je iets van elektrotechniek herinnert zoals je ongetwijfeld bij natuurkunde hebt gehad dan weet je hoe je een schakelaar in serie met een lampje aan moet sluiten om het lampje te laten branden.

Maar een computer kun je programmeren nietwaar?

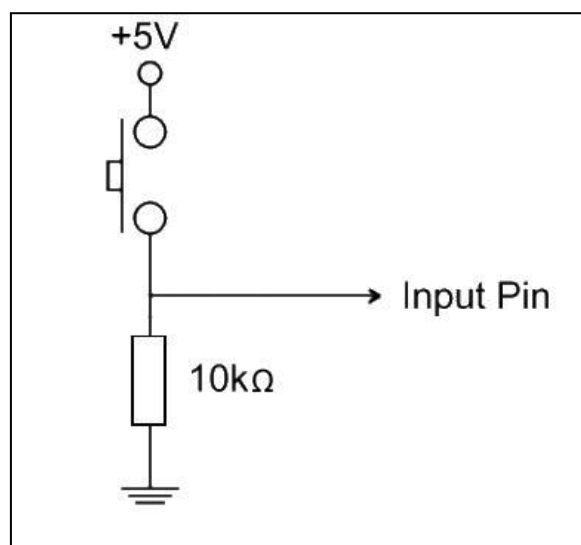
Dus sluit je een schakelaar aan op een computer dan ga je ook het gedrag van die schakelaar programmeren.

Je schakelt dus een LED of iets anders door tussenkomst van de Arduino.

Het grappige hiervan is dat je maar één type schakelaar nodig hebt en die kun je op allerlei manieren gebruiken.

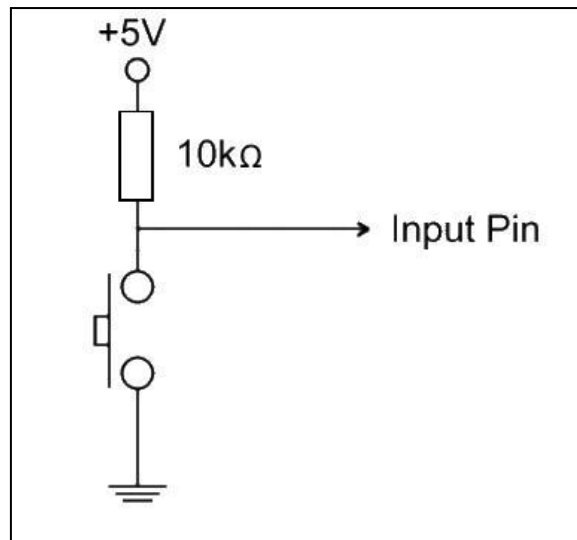
Bij het gebruik van een schakelaar maak je ook altijd gebruik van een **pull-up of pull-down weerstand** om omgevingsruis te voorkomen.

Door tussen de digitale IN en GND een grote weerstand te plaatsen zal deze verbonden zijn met de aarde als de drukknop niet ingedrukt is en dus nul volt (LOW) meten, maar HIGH als de drukknop wel ingedrukt is (zie afbeelding 18).



Afbeelding 18: Pull-down weerstand

Door een grote weerstand te plaatsen tussen de digitale IN en de 5V zal de Arduino altijd HIGH meten als de drukknop niet ingedrukt is (zie afbeelding 19).



Afbeelding 19: Pull-up weerstand

3.5 Actuatoren

Een computer kan reageren op de signalen die een sensor afgeeft en dat reageren gebeurt meestal door middel van het aansturen van een actuator.

Zo is een ledje, zoals we dat eerder gebruikt hebben, een actuator. Maar er zijn er veel meer op te noemen.

Als een temperatuursensor een signaal afgeeft, en in de computer is ingesteld (geprogrammeerd) dat bij een temperatuur gelijk aan of hoger dan een bepaalde waarde een uitgangspoort van waarde verandert (bijvoorbeeld van low naar high), dan kan daarmee een actuator zoals een airco-installatie aangestuurd worden. Maar ook zou je een zonnescherm kunnen laten zakken. Of simpelweg de kachel een stand lager zetten. Allemaal voorbeelden van actuatoren. Of in ieder geval van apparaten waar actuatoren ingebouwd zitten.

Er zijn heel veel verschillende soorten actuatoren zoals:

Relais



Afbeelding 20

Een relais is een schakelaar die reageert op een spanning aan de ingang. Er hoeft dan maar een heel klein stroompje door die ingang te lopen om hem te laten schakelen. Dus een kleine stroom vanuit de Arduino is voldoende en daarmee kun je dus apparaten mee schakelen die een grote stroom nodig hebben.

Led/lamp/display/beeldscherm



Afbeelding 21

Een led of een display (bestaat vaak uit leds) is bekend genoeg. Heeft vaak slechts een signaleringsfunctie of om berichten over te brengen.

Motor



Afbeelding 22

Met een elektromotor maak je van elektriciteit een ronddraaiende beweging.

Dat klinkt simpel maar er zijn nog wel verschillen. Zo heb je gewone ronddraaiende motortjes zoals in afbeelding 22. Door de aansluitspanning om te keren kun je zijn draairichting veranderen.



Afbeelding 23

Een ander type zie je in afbeelding 23. Dit is een zogenaamde stappenmotor of stepper motor.

Met een stappenmotor kun je de as heel nauwkeurig laten draaien bijvoorbeeld in stappen van 1,8 graden.

Tenslotte kennen we ook nog de servomotor. Daarmee kunnen we de as ook heel nauwkeurig laten draaien maar niet in vooraf vastgestelde stappen maar voornamelijk tot een bepaalde toestand is bereikt.

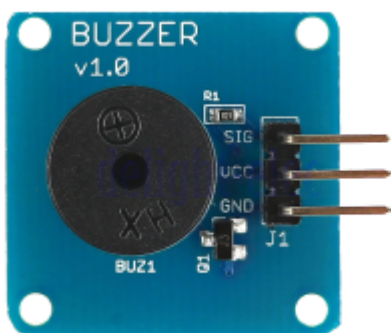
Trilelement in een telefoon



Afbeelding 24

Een trilelement is vaak ook een kleine motor die, als hij draait, voor een voelbare trilling zorgt. Hiernaast een afbeelding van een trilmotor uit een iPhone 4.

Luidspreker of buzzer



Afbeelding 25

Een buzzer is vaak prima te gebruiken voor signalering. Bijvoorbeeld in een auto als je dichtbij een muur komt. Of als er door lekkage ergens water staat.

Met een actieve buzzer zoals hiernaast (deze buzzer heeft ook een spanningsaansluiting) kun je één toon maken. Maar wel heel hard.

Met een passieve buzzer (meestal heeft die maar twee aansluitingen) kun je een melodie afspelen.

Met een luidspreker is het ook mogelijk muziek af te spelen zoals een ringtone.

Verwarmingselement



Afbeelding 26

Er zijn natuurlijk genoeg voorbeelden te bedenken van een verwarmingselement.

Hiernaast een vloermat. Sluit hem aan op jouw Home Automatiseringssysteem en je kunt er voor zorgen dat, zodra jij aan het huiswerk moet, de vloer lekker warm is.

Pomp



Afbeelding 27

Last van hoogwater in de kelder?

Of op vakantie maar de plantjes moeten wel water krijgen?

Meet de waterhoogte (of de droogte) en schakel een pomp aan.

Hiernaast een kleine pomp met een capaciteit van maximaal 2 liter/min.

De maximum stroom die hij dan verbruikt is 1 ampère bij een spanning van 12 volt.

Aangezien de Arduino niet echt veel vermogen kan leveren - per (digitale) poort maximaal 40 mA - kun je daar in de meeste gevallen geen actuator mee aansturen en moet je gebruik maken van een relais, een transistor of een apart (motor)shield.

Opdracht 15

- Zoek van ieder van de actuatoren hierboven een voorbeeld en probeer, voor zover mogelijk, te achterhalen wat hun eigenschappen zijn zoals:
 - Welke spanning hebben ze nodig?
 - Welke stroom verbruiken ze?
 - Wat hebben ze nodig om ze te laten werken (bijvoorbeeld een digitaal signaal (high of low) of een analoog signaal)?
 - Beschrijf hoe ze reageren (dus hoe je ze kunt gebruiken)

Opdracht 16

- Kijk nog eens naar de stofzuigerrobot van opdracht 2.
Welke actuatoren zijn hierbij betrokken?

3.6 Terug naar de schakelingen

Opdracht 17

Je kent vast wel de wisselschakeling of hotelschakeling. In vrijwel ieder huis komt een dergelijke schakeling voor bij de trap.

Zowel beneden als boven kun je de lamp aan- en uitschakelen.

- Teken eerst het toestandsdiagram
- Gebruik nu twee aanraakschakelaars en programmeer daarmee een wisselschakeling. (Sluit de signaal pinnen van de sensoren bijvoorbeeld aan op de poorten 2 en 3.)
- Maak bij deze (en de volgende) schakeling nu wel gebruik van echte ledjes. Denk daarbij om de voorschakelweerstand (zie 1.6.1, tenminste 220 Ω , kleurcode rood-rood-bruin)

De wisselschakeling is vrij normaal in huis. Maar minder vaak zie je een dergelijke schakeling waarbij je de lamp vanaf drie plekken kunt bedienen.

- Teken ook hier weer eerst het toestandsdiagram en ga dan met een derde aanraakschakelaar aan de slag.

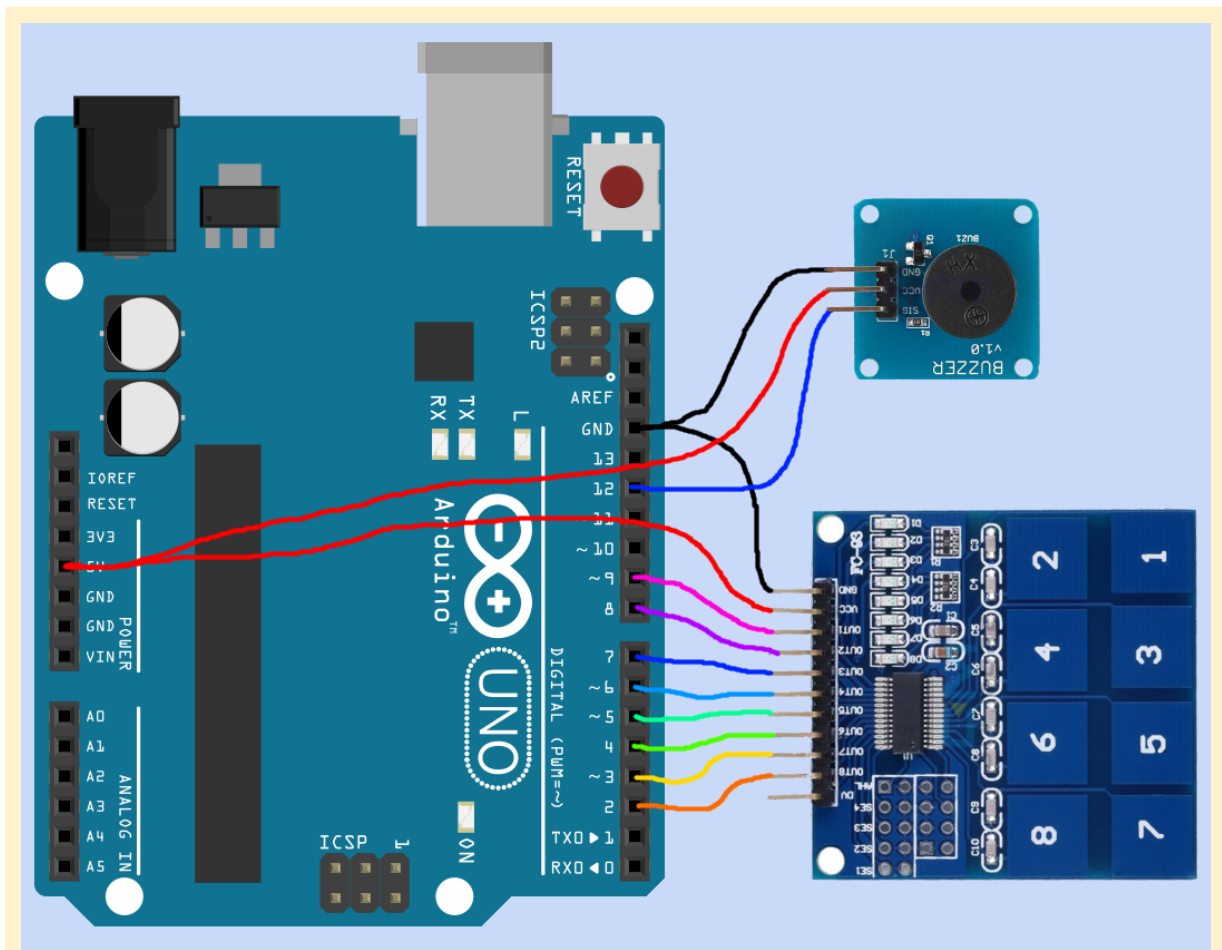
Opdracht 18

- Gebruik dezelfde opstelling als in opdracht 17 (met twee aanraakschakelaars).
- Lees deze opdracht door en teken de toestandsdiagram.
- Zorg er nu voor dat de led pas aangaat als je eerst de linker sensor aanraakt en daarna de rechter.
- Zorg er daarna voor dat de led pas weer uit wil als je eerst de rechter sensor aanraakt en daarna de linker.

Opdracht 19

In deze opdracht gaan we gebruik maken van een 8 kanaals touchpad en een buzzer als actuator.

- Sluit het touchpad aan volgens het schema hieronder. Sluit ook de buzzer aan. Het zal duidelijk zijn dat twee draadjes in een poort niet erg handig is dus maak hier gebruik van het breadboard.



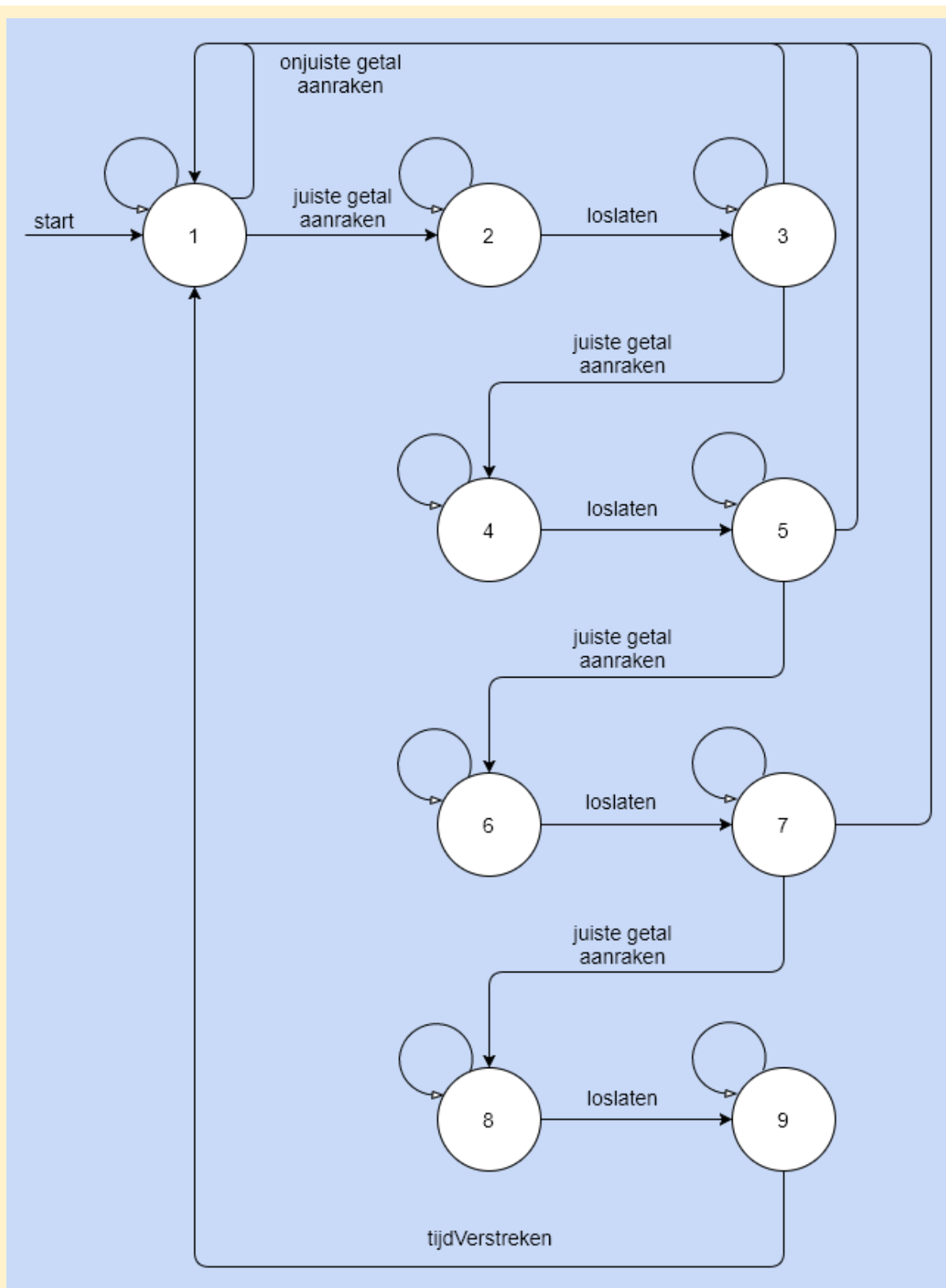
Je zult inmiddels gemerkt hebben dat de logica van het programmeren niet altijd eenvoudig is maar dat het door het tekenen van een goed toestandsdiagram een stuk gemakkelijker is om het programma te schrijven.

Ook de schakeling die we nu gaan maken is niet zo eenvoudig.

Het is de bedoeling dat we het ledje aan laten gaan als je de cijfers 1458 (exact in die volgorde) aanraakt zonder tussendoor andere cijfers aan te raken.

Let op: bij een verkeerde code moet je dus iedere keer weer opnieuw beginnen.

En dus beginnen we weer met het toestandsdiagram.



- Maak nu het programma waarmee je de buzzer een duidelijke piep laat geven als je de cijfers 1458 in die volgorde aanraakt zonder tussendoor andere cijfers aan te raken. Let op: bij een verkeerd ingegeven getal moet je dus iedere keer weer opnieuw beginnen.

- Als de code juist is gaat de buzzer kort piepen. Zolang hij piept heeft het aanraken van het touchpad geen effect.
- Zorg ervoor dat de buzzer na 2 seconden weer uit gaat.
- Daarna heeft het aanraken van het touchpad weer effect en kun je dus opnieuw de code intypen.

Een buzzer kun je ook gebruiken als waarschuwingssignaal. In een moderne (iets duurdere) auto zitten tegenwoordig aan de voor- en achterkant zogenaamde parkeersensoren. Meestal zijn dat sensoren die werken met een ultrasoon geluid (Ultrasonic Module Distance Measuring Transducer Sensor).

Deze sensor heeft als het ware twee onderdelen, de ene zendt een geluid uit en de andere vangt dat geluid weer op. Uitgaande van de snelheid van het geluid in lucht (340 m/s) kun je dan de afstand bepalen.

Opdracht 20

Je kent het vast wel, de parkeersensor.

Kom je met de auto te dicht bij een andere auto of een muur dan gaat de auto piepen. Dat is heel eenvoudig te maken met een ultrasone sensor.

- Bouw een opstelling met een ultrasone sensor en een buzzer.
- Zorg er met jouw programma voor dat de buzzer bij een afstand van 75 cm 1 keer per seconde gaat piepen met een korte piep.
- Zorg er met jouw programma voor dat de buzzer bij een afstand van 50 cm 2 keer per seconde gaat piepen met een korte piep.
- Zorg er met jouw programma voor dat de buzzer bij een afstand van 25 cm een continue piep laat horen.

3.7 Motorshields

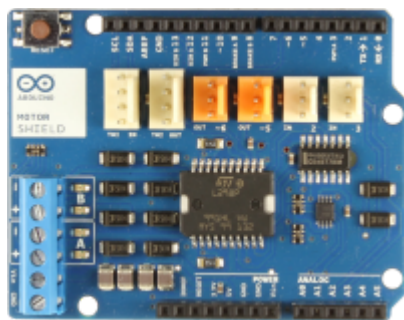
Zoals al eerder gemeld kun je een motor (behalve piepkleine servomotortjes) niet zonder meer rechtstreeks aansluiten op een Arduino.

Je kunt dan gebruik maken van een motorshield.

Voor de sensoren en de kleine actuators die je op de Arduino aan sluit maak je gebruik van de stroom die de Arduino kan leveren. Maar heb je wat meer vermogen nodig dan kun je dat via een motorshield doen. Op zo'n motorshield kun je een aparte stroombron aansluiten en daarmee hou je de stroom van de Arduino en die van (bijvoorbeeld) jouw motortjes gescheiden.

Er zijn veel verschillende motorshields verkrijgbaar. De bekendste zijn:

Arduino Motor Shield



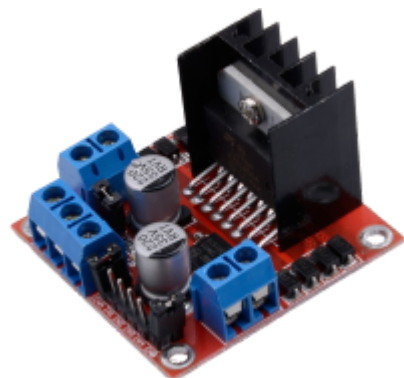
Afbeelding 28

Het officiële stackable motor shield van Arduino.

Stackable betekent dat je hem lekker simpel op een Arduino Uno kunt prikken. Dus weinig aan te sluiten.

Officieel betekent in verhouding tot de Chinese producten: duur.

Motor Shield op basis van de L298N



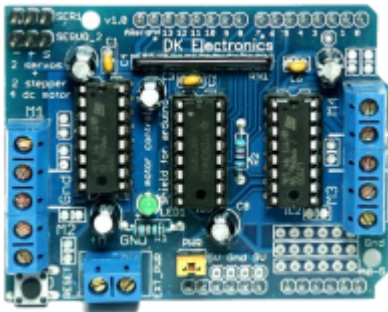
Afbeelding 29

Een goedkoop motor shield.

Hierover is veel informatie te vinden, ook over hoe je hem aan met sluiten.

Bijvoorbeeld [hier](#).

Motor Shield van Adafruit of DK Electronics



Afbeelding 30

Nog een stackable shield. Oorspronkelijk uitgebracht door Adafruit maar later ook door DK Electronics.
Ook hiervan is veel informatie te vinden o.a. [hier](#).

Opdracht 21

- Bouw een opstelling met een motor shield, een motor en twee schakelaars of aanraakschakelaars.
- Zorg dat de motor linksom draait als je op de ene schakelaar drukt en rechtsom als je op de andere schakelaar drukt.