



Vrije Universiteit Brussel

# Ontwerpproject

## Elektronische sensor- en actuatorsystemen

L. Standaert en Y. Verbelen



Vrije Universiteit Brussel  
Bachelor Industriële Wetenschappen  
Nijverheidskaai 170, B-1070 Anderlecht  
yannick.verbelen@vub.ac.be



This work is licensed under the Creative Commons  
Attribution-NonCommercial-ShareAlike 4.0 International License.  
To view a copy of this license, visit  
<http://creativecommons.org/licenses/by-nc-sa/4.0/>.

# Voorwoord

Vorig academiejaar heb je kennis gemaakt met de basisbeginselen van elektronisch ontwerp en prototyping toen je zelf een eenvoudige wekker ontworpen en gebouwd hebt. Hoewel de functionaliteiten van je eigen wekker enigszins gelimiteerd zijn in vergelijking met commerciële wekkers, zijn de fundamentele concepten van moderne elektronische systemen wel degelijk aanwezig. De wekker beschikt over knoppen waarmee de gebruiker de wekker kan besturen, een microcontroller (op een Arduino) om informatie te verwerken via software, en actuators zoals LED's, displays, buzzer e.d. waarmee iets nuttigs gedaan kan worden met die informatie. In feite berusten de meeste moderne elektronische applicaties op dezelfde principes: de verwerkingseenheid is bijna altijd een microcontroller waarop software wordt uitgevoerd, aangevuld met sensors en actuators specifiek aan de applicatie. De thermostaat van je centrale verwarming is bijvoorbeeld weinig meer dan een microcontroller waarop één of meerdere temperatuursensors zijn aangesloten, en die een relais stuurt om de verwarming aan of uit te zetten. Een mp3-speler is een krachtigere microcontroller die op basis van de invoer via zijn knoppen of touch screen de juiste mp3-bestanden decodeert en afspeelt via een luidspreker. Hetzelfde geldt voor wasmachines, microgolfovens, navi's, printers en talloze andere elektronische of mechatronische toestellen waar je dagelijks gebruik van maakt. Eens je het principe van invoer, verwerking en uitvoer van gegevens door hebt dan ben je niet alleen in staat om de werking van de meeste toestellen te begrijpen, maar ze ook aan te passen of extra functies toe te voegen ("hacken").

Waar de wekker beperkt was tot enkel elektronische componenten, wordt de scope dit jaar uitgebreid naar mechatronische systemen. Hiervoor wordt gebruik gemaakt van elektrische actuators zoals motors en solenoïdes om informatie vanuit microcontrollers om te zetten naar mechanische bewegingen zoals translaties of rotaties. Daarmee ligt de weg open naar mechatronische applicaties en, natuurlijk, robots. In deze cursus worden onder meer de verschillende actuators behandeld, hun fundamentele werking, toepassingsgebied, en de vermogenselektronica die nodig is om ze aan te sturen.

Naast actuators worden ook de knopjes op de wekker van vorig jaar uitgebreid met geavanceerdere sensors en transducers waarmee je applicatie fysische grootheden en omgevingstoestanden kan meten. Met deze informatie wordt je mechatronisch systeem zich bewust van zijn omgeving, en kan erop inspelen via software om zich nog intelligenter te gedragen. Sensors zijn ook de basis voor automatisering en controlesystemen, waarmee je in de 3<sup>e</sup> Bachelor kennis zal maken.

Het staat eenieder vrij om deze syllabus geheel of gedeeltelijk te repliceren en te verspreiden voor niet-commerciële doeleinden zoals vastgelegd in de Creative Commons BY-NC-SA 4.0 International licentie. Vrije toegang tot informatie behoort een basisrecht te zijn voor iedereen.

*Anderlecht, september 2015  
Y. Verbelen*



# Inhoudstafel

<b>Voorwoord</b>	<b>i</b>
<b>1 Apparatuur</b>	<b>1</b>
1.1 Inleiding . . . . .	1
1.2 De multimeter . . . . .	1
1.2.1 Voltmeter . . . . .	1
1.2.2 Amperemeter . . . . .	2
1.2.3 Ohmmeter . . . . .	2
1.2.4 Diodemeter . . . . .	2
1.3 De RLC-meter . . . . .	3
1.4 De oscilloscoop . . . . .	3
1.5 De regelbare voeding . . . . .	4
1.6 De signaalgenerator . . . . .	4
<b>2 Sensors</b>	<b>5</b>
2.1 Inleiding . . . . .	5
2.2 Sensors en transducers . . . . .	5
2.2.1 Rotatie en hoeken . . . . .	6
2.2.2 Beweging . . . . .	6
2.2.3 Afstand . . . . .	6
2.2.4 Krachten . . . . .	6
2.2.5 Contact . . . . .	8
2.2.6 Omgeving . . . . .	8
2.3 Digitaal of analoog . . . . .	8
2.3.1 I <sup>2</sup> C . . . . .	11
2.3.2 SPI . . . . .	12
<b>3 Actuators</b>	<b>13</b>
3.1 Inleiding . . . . .	13
3.2 Solenoïde . . . . .	13
3.3 Clutch . . . . .	14
3.4 Motors . . . . .	15
3.4.1 DC-motor . . . . .	15
3.4.2 Stepper . . . . .	16
3.4.2.1 Schematische werking . . . . .	16
3.4.2.2 Single coil / full step / half step / microstepping . . . . .	16
3.4.2.3 Aantal draden / types stepper . . . . .	17
3.4.3 Servo . . . . .	17
3.4.4 Brushless DC motor . . . . .	18
3.5 Bridge drivers . . . . .	20
3.5.1 Gepolariseerde sturing . . . . .	20
3.5.2 De H-brug . . . . .	21
3.5.2.1 Principe . . . . .	21
3.5.2.2 MOSFET H-brug . . . . .	24
3.5.2.3 Level shifting . . . . .	25
3.5.2.4 Flyback . . . . .	27
3.5.2.5 Clamping . . . . .	30
3.5.2.6 Monitoring . . . . .	31
3.5.2.7 Logische sturing . . . . .	31

3.5.2.8	Inschakelvertraging . . . . .	32
3.5.2.9	Galvanische isolatie . . . . .	34
3.5.2.10	Stroombegrenzing . . . . .	35
3.5.2.11	Geïntegreerde H-bruggen . . . . .	37
<b>4</b>	<b>Prototyping</b>	<b>39</b>
4.1	Inleiding . . . . .	39
4.2	De breadboardimplementatie . . . . .	40
<b>5</b>	<b>Debuggen</b>	<b>43</b>
5.1	Inleiding . . . . .	43
5.2	Basistechnieken . . . . .	43
5.2.1	Modulair debuggen . . . . .	44
5.2.2	Tracing . . . . .	44
5.2.3	Defecten identificeren . . . . .	44
5.2.4	Catastrophic failures . . . . .	46
5.3	PCB's debuggen . . . . .	47
5.4	Tips & Tricks . . . . .	47
5.4.1	LED's . . . . .	48
5.4.2	Testpunten . . . . .	48
5.4.3	Rubber Duck Debugging . . . . .	49
5.4.4	Thermisch debuggen . . . . .	49
5.4.5	Weerstand en jumpers . . . . .	50

# 1

## Apparatuur

“What is better, to be born smart, or to overcome your ignorant nature through great effort?”

---

Paarthurnax

### 1.1 Inleiding

In tegenstelling tot de mechanische constructie van je robot, spelen de processen die zorgen voor de logische sturing ervan zich af in de submicroscopische wereld van halfgeleiders en elektronen. Tenzij je schakeling dus een visueel waarneembaar resultaat heeft, zoals het inschakelen van een relais of het oplichten van een LED, zal je dus beroep moeten doen op gespecialiseerde apparatuur om de activiteiten van je schakeling zichtbaar te maken.

### 1.2 De multimeter

Multimeters zijn erg veelzijdige meettoestellen die, afhankelijk van het type, verschillende grootheden kunnen opmeten. In wat volgt bespreken we *digitale* multimeters. Analoge toestellen breng je bij voorkeur naar het dichtstbijzijnde museum.

#### 1.2.1 Voltmeter

De voltmeter is veruit de meest gebruikte configuratie van een multimeter, en laat je toe spanningen te meten. Hiertoe verbind je één probe met de common terminal (COM) en de andere met de terminal voor spanning ( $V/\Omega$ ). In deze configuratie is de interne impedantie van de multimeter groter dan  $1\text{ M}\Omega$  waardoor de probes over willekeurige meetpunten kunnen gezet worden zonder dat er stroom van betekenis gaat lopen.

De meeste multimeters beschikken over een stand om gelijkspanning (DC) en wisselspanning (AC) te meten. De AC meting is echter bedoeld voor de 50 Hz van het lichtnet, en geeft afwijkende resultaten bij andere frequenties. Wil je andere frequenties meten, maak dan gebruik van een oscilloscoop of een True RMS-meter.

Let erop nooit de maximale ingangsspanning van de multimeter te overschrijden. Dit staat meestal aangegeven tussen de respectievelijke terminals, en uiteraard ook in de handleiding. Als je niet zeker

bent van de grootte-orde van de spanning die je zal aantreffen, draai de meter dan in zijn grootste bereik en verfijn de nauwkeurigheid nadien desgewenst. Als de gemeten spanning hoger is dan het ingestelde bereik dan melden de meeste toestellen dit met een 1 in het meest linkse display of de letters OL (*overflow*).

Draagbare multimeters op batterijen bieden de mogelijkheid om ook spanningen die galvanisch in verbinding staan met de grond te meten zonder het risico te lopen aardlussen (*eng. ground loops*) te creëren. Op deze manier kan je bijvoorbeeld de spanning meten over componenten die op het lichtnet zijn aangesloten.

### **Opgelet:**

Let erop geschikte probes te gebruiken bij het meten van hoge spanningen zoals die van het lichtnet. Verifieer voor gebruik dat je probes niet beschadigd zijn.

## **1.2.2 Amperemeter**

De amperemeter laat je toe om stromen te meten. Net zoals bij voltmeters moet je een onderscheid maken tussen AC en DC. Ook hier is AC enkel accuraat bij 50 Hz, dus als je andere frequenties wil meten dan maak je beter gebruik van een oscilloscoop met een stroomtang.

De meeste multimeters hebben 2 terminals voor het meten van DC: eentje die gezeerd is tot 250 - 400 mA en eentje voor 10 A. Het kiezen van het juiste bereik is hier dus essentieel als je wilt vermijden om de zekering op de blazen! Ben je niet zeker welke grootte-orde van stroom je verwacht, plug de probes dan in de common (COM) en 10 A terminals in. Hiermee kan je stromen tot 10 A DC meten. Merk je dat de stroom lager dan het kleine bereik is, dan kan je de probe van de 10 A terminal verstoppen naar de  $\mu\text{A}/\text{mA}$  terminal om een nauwkeurigere aflezing te verkrijgen.

Merk je dat je steeds 0 mA meet, kijk dan de zekering binnenin de multimeter na en vervang deze als ze opgeblazen blijkt. Uiteraard mag de vervangende zekering de specificaties van de multimeter niet overschrijden!

## **1.2.3 Ohmmeter**

In de ohmmeterstand kan je de multimeter gebruiken om elektrische weerstand op te meten. Hiertoe sluit je de probes aan tussen de common (COM) en weerstand ( $V/\Omega$ ) terminals. In deze configuratie geeft de multimeter standaard een overflowconditie aan (weerstand tussen de probes valt buiten het bereik).

Om de weerstand te meten zal de multimeter een kleine teststroom door de aangelegde weerstand sturen. Hoe groter het bereik je instelt, hoe groter deze teststroom wordt. Let er dus op om het bereik steeds zo klein mogelijk in te stellen als je gevoelige componenten wil opmeten. Als je schakeling onder spanning staat terwijl je probeert weerstand te meten, dan wordt de aangelegde teststroom mogelijk beïnvloed en is de meting dus niet correct. Koppel de voeding dus steeds af en ontlad condensators.

## **1.2.4 Diode-meter**

Een laatste configuratie van multimeters is de diode-meter, die toelaat om diodejuncties uit te meten. Om diodes te meten sluit je de probes aan op de common (COM) en diode ( $V/\Omega$ ) terminals. Het getal op het display geeft de junctiespanning aan in millivolt. Siliciumdiodes zitten tussen 600 - 700 mV, Schottkydiodes tussen 200 - 400 mV. Bipolaire junctietransistors (BJT's) zijn eveneens uit te meten op deze manier. Meet je 0, dan is de diode stuk (doorgebrand). Controleer de polariteit van de diode als je een overflow meet. Meet je in alle richtingen overflow, dan is de diode waarschijnlijk stuk.

De maximaal meetbare junctiespanning is meestal 1000 mV. Componenten met een hogere junctiespanning, zoals LED's, zijn gewoonlijk dus niet meetbaar.

Eenvoudige multimeters die spanning, stroom en weerstand kunnen opmeten, hoeven helemaal niet duur te zijn. Vanaf 15 EUR kan je al een degelijk exemplaar op de kop tikken, en online vind je ze op sites als Ebay of AliExpress zelfs nog strakker geprijsd. Harddiscounters zoals Lidl en Aldi zetten ze bovendien regelmatig in de aanbieding.

### 1.3 De RLC-meter

De RLC-meter is een toestel waarmee je weerstanden, inductors (spoelen), condensators en combinaties van deze 3 kan opmeten. Een RLC-meter kan je vertellen of je schakeling een negatieve of positieve faseverschuiving (bij resp. een capacitief of inductief gedrag) tussen spanning en stroom zal vertonen, en soms ook de hoek ( $\cos \phi$ ) tussen beiden.

RLC-meters zijn vooral handig voor het opmeten van zelf gewikkelde spoelen, transformators e.d., en om de parasitaire capaciteit tussen tracks op je PCB's te bepalen.

Net zoals een multimeter in Ohmmeterconfiguratie zal een RLC-meter enkel een accurate waarde kunnen geven als de gemeten deelschakeling spanningsloos is. Schakel dus steeds de voeding af en ontlad condensators voor je je schakeling aan de klemmen van de RLC-meter hangt.

### 1.4 De oscilloscoop

Waar een multimeter ideaal is om het DC-gedrag van een schakeling op te meten, is een oscilloscoop essentieel om het AC-gedrag van een schakeling te bestuderen. Een oscilloscoop beschikt daartoe over een LCD-display waarop variërende signalen zoals sinusoïdale spanningen of blokgolven kunnen afgelezen worden.

Om een oscilloscoop aan te sluiten heb je een probe met een BNC- (coax) stekker nodig. Duw deze probe over de BNC-connector van de oscilloscoop en draai hem een kwart slag in wijzerzin om hem te vergrendelen. Je kan 2 signalen tegelijk bemonsteren, deze heten respectievelijk kanaal 1 en 2 (CH1 en CH2). Kanaal 1 wordt steeds weergegeven in geel, kanaal 2 in cyaan. In XY-modus vertegenwoordigt kanaal 1 steeds X, en kanaal 2 steeds Y.

#### Opgelet:

Bij een oscilloscoop zijn de negatieve terminals van beide kanalen intern met elkaar en met de aarde doorverbonden. Je hoeft dus slechts één probe met de grond van je schakeling te verbinden. Door de interne aarding kan je met een oscilloscoop **geen** signalen meten die galvanisch met de netspanning verbonden zijn!

Het display van een oscilloscoop is opgedeeld in een horizontaal en vertikaal raster, en de afstand tussen 2 opeenvolgende rasterlijnen heet een *divisie* (eng. *division*). De oscilloscoop geeft onderaan het scherm aan met welke spanning 1 divisie vertikaal overeen komt, en met welke tijdspanne 1 divisie horizontaal correspondeert. De amplitude van een signaal meet je dus door het aantal verticale divisies te tellen boven de oorsprong en te vermenigvuldigen met de spanning per divisie. Voor de peak-to-peak spanning ( $V_{pp}$ ) tel je alle divisies van de laagste tot de hoogste spanning binnen een periode. Hetzelfde geldt voor het meten van de periodelengte, hiervoor tel je het aantal divisies binnen één periode en vermenigvuldig je dit met de tijdspanne van 1 divisie. Door de meest linkse knop te verdraaien kan je de spanning per verticale divisie instellen, de middelste draaiknop stelt de tijd per horizontale divisie in.

Via de functie *Measure* kan je bij periodieke functies ook automatisch de gemiddelde spanning, peak-to-peak spanning, periode en frequentie laten berekenen en in real time weergeven op het display. Merk op dat dit enkel accuraat is voor perfect periodieke functies. Veranderende signalen zoals digitale



communicatielijnen, muziek enz. kunnen op deze manier niet nauwkeurig gemeten worden.

Als je niet weet welk signaal je verwacht, maak dan gebruik van de functie *Auto* rechtsbovenaan het bedieningspaneel. De oscilloscoop gaat dan zelf op zoek naar signalen op de aangesloten kanalen en zal deze trachten zo goed mogelijk weer te geven op het display.

## 1.5 De regelbare voeding

Vooraleer je je robot test op batterijen, verifieer je uiteraard eerst of je elektronica wel werkt op een geregleerde DC-spanning. Om deze spanning op te wekken, maak je gebruik van een regelbare voeding. Deze voeding laat je toe om een willekeurige DC-spanning van 0 V tot ca. 30 V in te stellen. De meeste types beschikken over een display waarop de actuele spanning op de uitgang wordt weergegeven.

Regelbare voedingen voor elektronica zijn gestabiliseerd, dit betekent dat hun uitgangsspanning niet wijzigt als je de belasting wijzigt.

Hoewel voedingen kortsluitvast zijn, en dus niet beschadigd worden als je de uitgang per ongeluk kortsluit, is de schakeling die je voedt ongetwijfeld minder resistent tegen kortsluitingen. Maak daarom gebruik van de stroombegrenzing op de voeding om een maximale uitgangsstroom in te stellen. Wanneer deze stroom overschreden wordt door een fout in je schakeling, dan zal de uitgangsspanning automatisch worden teruggeregeld zodanig dat de defecte componenten in je schakeling niet meteen beginnen te smeulen.

Een voeding is galvanisch gescheiden van het lichtnet, en je mag schakelingen die hiermee gevoed worden dus opmeten met een oscilloscoop. Voedingen gedragen zich in de meeste opzichten als batterijen die nooit leeg geraken, en waarvan je de spanning kan kiezen. Net zoals batterijen kan je voedingen dus ook in serie schakelen, bijvoorbeeld om een symmetrische voeding te creëren of een hogere uitgangsspanning te bereiken.

## 1.6 De signaalgenerator

Om het DC-gedrag van een schakeling uit te meten, volstaat een regelbare voeding. Wil je echter het AC-gedrag van een schakeling bepalen, bijvoorbeeld hoe die reageert op een ingangssignaal zoals een sinusoïde of blokgolf, dan kan je dit signaal opwekken met een *signaalgenerator*.

Een signaalgenerator laat je toe om willekeurige periodieke signalen zoals sinusoïdale spanningen en blokgolven met instelbare duty cycle, frequentie en amplitude te produceren. Let erop dat de meeste signaalgenerators een toggle switch op hun uitgang hebben zitten waarmee je de uitgang expliciet moet aanzetten. Dit verhindert dat je de BNC-connector steeds weer moet losmaken. Een servo-motor kan je bijvoorbeeld eenvoudig testen met een signaalgenerator door de duty cycle van het signaal te wijzigen, en te checken of de servo meedraait.

Signaalgenerators kunnen geen complexe signalen zoals digitale communicatiepakketten genereren. Hiervoor heb je een *Logic generator* voor nodig, waar we in het 3<sup>e</sup> jaar op terugkomen.

# 2

## Sensors

“Byrne’s Law: in any electrical circuit, components and wiring will burn out to protect fuses.”

---

Robert Byrne

### 2.1 Inleiding

Sensors zijn de zintuigen van een mechatronisch systeem: ze laten je robot toe om te zien, te voelen, te horen, ... Met elektronische sensors kan je zelfs fenomenen waarnemen die de meeste zoogdieren (waaronder mensen) niet kunnen detecteren. Er bestaan bijvoorbeeld sensors voor magneetvelden, ioniserende straling, infrarood licht, ultrasoon geluid, elektrische velden enzovoort.

In de 3<sup>e</sup> Bachelor wordt in het vak *Geïntegreerde Elektronische Systemen* dieper ingegaan op sensorsystemen, hun beperking en stuelelektronica ervoor. Gelukkig hoeft je al deze technische details niet te kennen om de meeste sensors te kunnen gebruiken dankzij de vereenvoudigde interface die Arduino je biedt. De Arduino kan dienst doen als busdecoder voor I<sup>2</sup>C of SPI, de twee meest gebruikte interfaces voor digitale sensors, en de geïntegreerde analoog naar digitaalconverter van Arduino (ADC) laat toe om rechtstreeks analoge sensors in te lezen, eventueel in combinatie met een eenvoudige analoge voorversterker. Dit hoofdstuk is er bijgevolg op gericht om je een overzicht te geven van de voornaamste types sensors, waarvoor je ze kan gebruiken, en hoe je ze moet aansturen vanuit Arduino. Voor de details wordt verwezen naar de cursus *Geïntegreerde Elektronische Systemen*.

Praktische informatie met voorbeeldcode vind je op de Rapptorwebsite<sup>1</sup> onder “Education”.

### 2.2 Sensors en transducers

Het overzicht hieronder heeft zeker niet tot doel je een complete lijst te geven met alle bestaande sensors, dat is quasi onmogelijk en zou ook weinig zinvol zijn. Onbekend is echter onbemand, en door je enkele veelvoorkomende sensors voor te stellen, zal je gemakkelijker applicaties herkennen waarin je ze kan gebruiken. Door de juiste sensor te kiezen voor een bepaalde situatie, kan je jezelf heel wat ellende besparen. Hieronder vindt je de meest voorkomende sensors met een breed toepassingsgebied, die ook betaalbaar zijn voor praktische applicaties.

---

<sup>1</sup><http://rapptor.vub.ac.be>

### 2.2.1 Rotatie en hoeken

Omdat motors de meest voorkomende actuators zijn en ze arbeid leveren in de vorm van een roterende as, zijn sensors die toerentallen en hoeken kunnen meten van groot belang. Je kan ze namelijk gebruiken als feedback bij motortypes die zelf geen ingebouwde feedback hebben zoals DC-motors. Tabel 2.1 toont een overzicht met de meest voorkomende hoek- en rotatiesensors.

Naam	Gebruik	Interface	Voorbeeld
Potentiometer	Een circulaire regelbare weerstand met middenafstakking op verdraaibare as. Meestal beperkt tot 360° of minder. Potentiometers met meerdere toeren bestaan maar zijn zeer duur. Toegepast als feedback in hobby-servomotors.	Analoog	WH148
Rotationale encoder	Een lichtsluis die gaten in een geperforeerde draaiende schijf waarneemt. Onbeperkt aantal toeren. Duurdere types hebben 2 uitgangen zodat ook de rotatiezin van de as kan bepaald worden. Toegepast als toerenteller.	Digitaal	25LB22-H
Tiltsensor	Een metalen kogel die afhankelijk van de hoek al dan niet contact maakt tussen metalen behuizing en centreelektrode. Typisch gebruikt in de transportsector om te achterhalen of goederen niet omgevallen zijn tijdens vervoer.	Digitaal	AT407
Accelerometer	MEMS-sensor die versnellingen in 3D kan waarnemen, en dus hoek t.o.v. gravitatievector kan bepalen. Gebruikt in drones om horizontaal te vliegen.	Analoog, I <sup>2</sup> C	ADXL345
Kwikschakelaar	Variant op tiltsensor met een druppel kwik in plaats van een metalen kogel. Tegenwoordig weinig gebruikt omdat het toxische kwik vrijkomt als de glazen buis breekt.	Digitaal	KY-017

Tabel 2.1: Overzicht van hoek- en rotatiesensors.

### 2.2.2 Beweging

Naast rotaties komen ook lineaire bewegingen (in hetzelfde vlak) veel voor. Om die te meten kan de lineaire beweging ofwel omgezet worden naar een rotatie, ofwel kunnen lineaire beweging- en positie-sensors gebruikt worden. Tabel 2.2 toont de voornaamste sensors voor lineaire bewegingen.

### 2.2.3 Afstand

Afstanden opmeten is een sporadisch weerkerend probleem in mechatronische systemen. Denk bijvoorbeeld aan autonome voertuigen die obstakels moeten detecteren, een digitale camera die zichzelf wil focussen, een parkeerhulp in personenwagens enzovoort. Afhankelijk van gewenste minimale en maximale afstand enerzijds en de precisie van de meting anderzijds, zijn er verschillende mogelijkheden om afstanden te meten. Tabel 2.3 geeft een overzicht van veel voorkomende afstandssensors (*eng. range finder*).

### 2.2.4 Krachten

Toepassingen waarin de exacte grootte van uitgeoefende krachten of koppels van belang is vereisen aangepaste krachtsensors. Een robothand die een ei moet kunnen optillen zonder het plat te nijpen kan daar een voorbeeld van zijn, maar het kan even nuttig toegepast worden om de krachtenwerking op gevoelige

Naam	Gebruik	Interface	Voorbeeld
Schuifpotentiometer	Potentiometer met een lineair <i>slider</i> . Gebruikt als sensor waar een absolute meting van korte afstanden vereist is, of als interface-element voor volume- en snelheidsregelingen e.d.	Analoog	PS100
Rekstrookje	Een serpentine van fijne koperdraad op rekbaar materiaal waarvan de diameter en dus weerstand verandert in functie van de uitrekking. Toegepast om de uitrekking van stangen en andere mechanische onderdelen te meten. De kleine weerstandsveranderingen vereisen nauwkeurige versterkerelektronica.	Analoog	MMF003129
Membraanpotentiometer	Variant op schuifpotentiometer met een membraan in plaats van slider. Duurder dan schuifpotentiometers maar dunner en duurzamer.	Analoog	TSP-L-0170-103

Tabel 2.2: Overzicht van beweging- en positie-sensors.

Naam	Gebruik	Interface	Voorbeeld
Sonar	Transducer die de afstand tot een obstakel meet als tijd die een uitgezonden puls ultrason geluid nodig heeft om terug te kaatsen. Toegepast als autofocus voor macrolenzen en als parkeerhulp. Werkt tot ongeveer 4 meter. Meerdere sonars op dezelfde frequentie zullen met elkaar interfereren.	Analoog, I <sup>2</sup> C, SPI	HC-SR04
IR-sensor	Zelfde principe als sonar, maar met infrarood licht in plaats van geluid. Grotere afstand dan sonar, maar werkt pas boven een minimale afstand. Toegepast voor navigatie van stofzuigerrobots.	Analoog, I <sup>2</sup> C, SPI	GP2Y0A41SK0F
LIDAR	Lichtradar ( <i>eng. Light raDAR</i> ), zelfde principe als sonar maar met laser in plaats van geluid. Duur, maar zeer lange afstanden mogelijk. Toegepast voor hoogtemetingen bij drones.	I <sup>2</sup> C, PWM	LIDAR-Lite
Reflectiesensor	LED en fotodiode die nabijheid van obstakels meet als functie van hoeveelheid teruggekaatst licht. Typisch toegepast in autonome voertuigen die lijnen moeten volgen, bv. in de logistieke sector. Vereist contrasterende oppervlakken (met variabele lichtabsorptiecoëfficiënten) en op korte afstanden (cm range).	Analoog, digitaal	QRE1113

Tabel 2.3: Overzicht van afstandssensors.

mechanische onderdelen te meten, of op het gewicht van een last te bepalen in een weegschaal. Tabel 2.4 toont een overzicht van enkele krachtsensors.

Naam	Gebruik	Interface	Voorbeeld
Flexsensor	Rekstrookje op een buigbare polymeer dat verbuigt onder invloed van een externe kracht.	Analoog	FS-L-0095-103-ST
Accelerometer	MEMS-sensor die versnellingen en dus ook trillingen kan meten in 3D.	Analoog, I <sup>2</sup> C	ADXL345
Vibratiesensor	Variant op accelerometer die trillingen over een breed frequentiebereik kan meten	Analoog	Minisense 100
Schoksensor	Variant op vibratiesensor die geactiveerd wordt door grote trillingen. Gebruikt als “warranty void” sensor in draagbare gadgets en magnetische harde schijven.	Analoog	801S
Krachtsensor	Weerstand waarvan de impedantie verandert in functie van de aangelegde kracht, FSR ( <i>eng. force sensitive resistor</i> ). Toegepast in reservatiesystemen om vrije treinzetels te vinden.	Analoog	ZFLEX A201-1
Load cell	Rekstrookje aangebracht op een verbuigbare aluminium cantilever. Typisch toegepast in weegschalen.	Analoog	TAL201
Piëzosensor	Keramisch piëzo-elektrisch materiaal dat een spanning genereert als de kristalstructuur wordt samengedrukt door een externe kracht.	Analoog	LDT0-028K

Tabel 2.4: Overzicht van kracht- en trillingsensors.

### 2.2.5 Contact

Vanuit een functioneel perspectief spelen contactsensors veruit de belangrijkste rol in mechatronische systemen. Ze worden gebruikt als eenvoudiger alternatief voor potentiometers in situaties waar geen absolute positionering vereist is, of waar slechts een digitale status nodig is (of er wel of geen contact is). De voorbeelden zijn eindeloos: deurbeveiliging in de metro, papiersensor in printers, eindloopschakelaar (“end stop”) in CNC-machines, enzovoort. Ze spelen ook een belangrijke rol bij de hercalibratie van deze systemen, om te kunnen terugkeren naar een gekende uitgangspositie als bijvoorbeeld de stroom uitvalt of om één of andere reden de actuators blokkeren en absolute positie-informatie niet voor handen is. Tabel 2.5 somt de belangrijkste contactsensors op.

### 2.2.6 Omgeving

Tot slot zijn er nog sensors die onder geen van bovenstaande categorieën vallen, voornamelijk sensors die omgevingsvariabelen of fysische grootheden zoals licht en temperatuur meten. Ze kennen tal van toepassingen: temperatuursensors bewijzen bijvoorbeeld hun nut in de afregeling van je oven als je pizza bakt, de extrusiekop van 3D-printers op temperatuur houden, en je processor beschermen tegen oververhitting als je aan het gamen bent. Tabel 2.6 geeft een overzicht.

## 2.3 Digitaal of analoog

Op enkele uitzonderingen zoals drukknoppen na zijn quasi alle sensors intrinsiek analoog. De manier waarop sensors de fenomenen waarvoor ze bedoeld zijn meten is echter onafhankelijk van hun elektronische *interface*, m.a.w. de manier waarop je data uit de sensor leest. Er zijn 3 mogelijkheden:

Naam	Gebruik	Interface	Voorbeeld
Druknop	Simpelste vorm van schakelaar bestaande uit 2 contacten die door indrukking kortgesloten worden. In elektronica toegepast als <i>tactile switch</i> .	Digitaal	FSM2JRT
Lichtsluis	U-vormige sensor met in het ene been een LED en andere been een fotodiode. Toegepast als contactloze eindeloopschakelaar of in rotationele encoders. Ook bekend als foto-interrupters.	Digitaal	GP1A57HRJ00F
Reed relais	Twee parallelle metalen contacten die worden kortgesloten door een extern magneetveld aan te leggen.	Digitaal	MKA-14103
Hallsensor	Solid state sensor die externe magneetvelden meet als spanningen. Kan in tegenstelling tot reed relais noordpolen van zuidpolen onderscheiden.	Analoog	SS49E

Tabel 2.5: Overzicht van contactsensors.

een analoge interface, een digitale interface of een businterface.

De eenvoudigste sensors hebben een analoge interface. Dit is het geval voor de meeste sensors die grootheden waarnemen via een weerstand die daarvoor gevoelig is. Zulke sensors hebben een tweedraadsinterface en gedragen zich als een weerstand waarvan de impedantie (= weerstandswaarde) verandert in functie van de grootte. Dergelijke sensors gaan steeds vergezeld van een grafiek in de datasheet die het verband tussen de weerstandswaarde en de gemeten grootte geeft. Als dit verband niet te ingewikkeld is (bv. lineair, kwadratisch, logaritmisch, ...) dan wordt de grafiek soms ook vervangen door een formule met bijhorende fout (afwijking). Voorbeelden van dit soort sensors zijn LDR's (waarvan de weerstand verandert in functie van het invallend licht), NTC's (waarvan de weerstand verandert in functie van de temperatuur) en rekstrookjes (waarvan de weerstand verandert in functie van de uitrekking). Met een Arduino kan je niet rechtstreeks weerstanden meten, maar door de veranderende weerstand om te zetten in een spanning via een spanningsdeler kan je toch rechtstreeks de ADC (analoog naar digitaalconverter) van de Arduino gebruiken. Hiertoe sluit je deze sensors dus aan op de analoge pinnen A0 tot A5.

Als het uitgangssignaal van de sensor erg klein is, bijvoorbeeld in het geval van een rekstrookje, dan kan voorversterking noodzakelijk zijn. In de cursus *Elektronisch Ontwerpen* vind je talrijke opampschakelingen die zich daar uitstekend toe lenen, zoals bijvoorbeeld niet-inverterende versterkers of buffers. Keuze van de opamp is essentieel voor de performantie.

Analoge signalen over lange afstand transporteren leidt onherroepelijk tot contaminatie met ruis, wat zeker bij erg kleine signalen een probleem is omdat je dan zelf nog moet gaan filteren. De verwerking wordt dan een stuk ingewikkelder omdat de schakeling complexer wordt en/of meer software nodig is. Om dit probleem te elimineren wordt in analoge sensors met lage uitgangssignalen soms al een versterkerschakeling geïntegreerd. Deze sensors geven dan een uitgangsspanning die tussen 0 en bijvoorbeeld 5 V ligt zodat je ze meteen kan inlezen. Hallsensors zoals de SS49E zijn typische voorbeelden hiervan. Je kan sensors met geïntegreerde verwerking- en versterkingslogica gemakkelijk herkennen doordat ze een extra aansluiting voor voedingsspanning hebben. Deze actieve sensors hebben dus 3 pinnen in plaats van de gebruikelijke 2.

In sensors met een ingewikkelde verwerkingseenheid, zoals bijvoorbeeld infraroodontvangers voor TV's zoals de TSOP1736, gaan fabrikanten nog een stap verder. Om het gebruiksgemak te verhogen wordt alle analoge signaalverwerking (filteren, versterken en analoog naar digitaalconversie) in de sensor zelf afgehandeld. Het resultaat van deze bewerkingen is een digitaal signaal, en de uitgang van deze

Naam	Gebruik	Interface	Voorbeeld
Lichtsensoren	Lichtgevoelige weerstand (LDR, <i>eng. light dependent resistor</i> ), meestal CdS.	Analoog	GL5528
Temperatuursensoren	Temperatuurgevoelige weerstand. Bestaat in 2 versies: weerstand evenredig met temperatuur (PTC) en weerstand omgekeerd evenredig met temperatuur (NTC).	Analoog	LM35
Microfoon	Transducer die trillingen in de lucht omzet in elektrische signalen met dezelfde frequentie.	Analoog	Electret
Vochtigheidssensoren	Transducer waarvan de weerstand verandert in functie van de relatieve luchtvochtigheid. Meestal gecombineerd met interne versterker en/of controller.	I <sup>2</sup> C, SPI	DHT11
Kleurensensoren	Lichtsensoren met RGB-LED die kleur van een object bepaalt door uit te zoeken welke golflengtes worden geabsorbeerd en welke worden teruggekaatst. Werkt enkel op korte afstand en bij weinig of geen omgevingslicht.	Analoog, I <sup>2</sup> C, SPI	TCS230
Camera	CCD ( <i>eng. charge-coupled device</i> ) die afbeeldingen kan waarnemen in zwart-wit, grijschaal of kleur. Steeds gecombineerd met interne versterker en digitale controller. Vrij moeilijk aan te sturen, vereist veel rekenkracht van de microcontroller.	Parallel, I <sup>2</sup> C, SPI, LVDS	OV3640
PIR-sensoren	Fotodiode die gevoelig is voor infrarood licht. Toegepast om warmtebronnen op afstand te detecteren, zoals in inbraakalarm-systemen. Vaak verkocht als module met 180° lens.	Analoog	HC-SR501
Gassensoren	Elektrochemische transducer die vluchtige chemicaliën kan onderscheiden. Bestaan voor uiteenlopende gassen en combinaties.	Analoog	MQ-136
Geigerteller	Buis gevuld met inert gas waarvan de geleiding wordt beïnvloed door ioniserende ("radioactieve") straling. Sturing analoog aan vacuümbuizen, vereist hoge DC bi-sspanning (> 200 V).	Analoog	SBM-20
Rooksensoren	Gevoelige variant op een lichtsluis die aerosols, rook en stof kan detecteren in de lucht. Gebruikt in ventilatiesystemen, rookmelders, luchtfilters e.d.	Analoog	GP2Y1010AU0F
Barometer	MEMS-sensoren die luchtdruk meet als een kracht op een membraan van gekende oppervlakte. Toegepast voor hoogtebepaling in de luchtvaart of voor weersvoorspellingen.	I <sup>2</sup> C	BMP180
pH-sensoren	Platinum en iridium contact waartussen een potentiaal (elektrochemische cel) ontstaat wanneer beide ondergedompeld worden in dezelfde vloeistof. Lage uitgangsspanning vergt nauwkeurige externe versterker. Toegepast voor monitoring van corrosie of chemische processen.	Analoog	CT-1111

Tabel 2.6: Overzicht van omgevingsensors.

sensors is dus ook digitaal. Infraroodontvangers zoals de TSOP1736 produceren bijvoorbeeld een actief digitaal “lage” uitgang als ze een infraroodsignaal ontvangen, en een digitaal “hoge” uitgang als ze niks ontvangen. Dergelijke sensors hoeven dus niet naar de analoge pinnen van de Arduino geleid te worden, maar kunnen direct via de digitale IO-pinnen 2 tot 13 ingelezen worden. Soms, zoals bij de TSOP1736, is wel een pullupweerstand vereist.

Wanneer er meer dan één bit informatie moet overgedragen worden, worden bij digitale sensors gebruikelijk extra digitale uitgangspinnen toegevoegd, die al dan niet aangesloten kunnen worden. Een interruptpin, die aangeeft of en wanneer er informatie binnenkomt, is daar een voorbeeld van. Bij ingewikkelde sensors zoals de TCD1304 CCD van Toshiba wordt zelfs een volledig digitale bus van 8 bits breed voorzien waarmee per kloktik een volledige byte parallel kan overgedragen worden. Dergelijke oplossingen bezetten dus snel een vrij groot aantal pinnen op je Arduino. Andere sensors dragen informatie over via een gecodeerd signaal zoals PWM (pulsbreedtemodulatie), PCM (pulscode-modulatie) of PDM (pulsdichtheidmodulatie). Anderen gebruiken zelfs specifieke protocollen zoals de DHT11 vochtigheidssensor. De code die je op Arduino zal moeten schrijven om deze sensors uit te lezen zal dus behoorlijk complex zijn. Gelukkig kan je online op bv. Github of Sourceforge driverbibliotheken voor de meeste van deze sensors downloaden waardoor je niet meer zelf de datasheet in detail moet lezen om de juiste driversoftware te schrijven.

Om de wildgroei aan componentspecifieke digitale protocollen in te perken werden er standaardinterfaces gedefiniëerd. Deze interfaces maken gebruik van een klein aantal pinnen om informatie over te dragen via een zeer ingewikkelde techniek die een digitale *bus* wordt genoemd. Deze meest gebruikte busprotocollen zoals I<sup>2</sup>C of SPI zijn gelukkig goed gedocumenteerd en vastgelegd in standaarden, en alle componenten die ze ondersteunen kunnen rechtstreeks met elkaar communiceren zonder dat daar bijkomende elektronica voor nodig is. De *state machines* die het werk achter de schermen uitvoeren, zitten in de sensors en in de ATmega328p op de Arduino Uno geïntegreerd.

In de 3<sup>e</sup> Bachelor worden I<sup>2</sup>C en SPI in het vak *Geïntegreerde Elektronische Systemen* uitvoerig besproken. Hieronder wordt enkel een samenvatting gegeven om je er nu al mee op weg te helpen.

### 2.3.1 I<sup>2</sup>C

I<sup>2</sup>C (afkorting van IIC of *Inter IC Communication*) is het meest gebruikte digitale protocol om digitale sensors met microcontrollers te verbinden. I<sup>2</sup>C is flexibel en compact omdat het slechts 2 datalijnen vereist: SDA (seriële data en adressen) en SCL (seriële kloklijn). De 2 lijnen SDA en SCL samen vormen de I<sup>2</sup>C-bus. De spanning op SDA en SCL is variabel en wordt bepaald door 2 pullupweerstand, één voor elke lijn. Pullupweerstand naar bijvoorbeeld 5 V zorgen ervoor dat de bus op 5 V draait. De exacte waarde van deze pullupweerstand is niet van belang, typisch wordt een waarde tussen 4.7 k $\Omega$  en 10 k $\Omega$  gekozen afhankelijk van het aantal devices op de bus en de fysieke lengte ervan.

Om een sensor met I<sup>2</sup>C-interface aan te sluiten op een Arduino Uno verbindt je de SDA-lijn van de sensor met de SDA-lijn van de ATmega328p, en ook de SCL-lijn van de sensor met de SCL-lijn van de ATmega328p. Welke pinnen dit precies zijn staat in de datasheet van de Arduino, of je vindt ze ook in de cursus *Ingenieursvaardigheden* uit de 1<sup>e</sup> Bachelor. Vergeet de pullupweerstand niet!

Eens je de sensor verbonden hebt, dan kan je via de `Wire` library van Arduino digitale data naar de sensor sturen en weer uitlezen. De Arduino handelt het ingewikkelde busprotocol automatisch af.

Je kan meerdere sensors met I<sup>2</sup>C-interface aansluiten op dezelfde bus op voorwaarde dat elke sensor een uniek adres bezit. Het adres van elke sensor staat in zijn datasheet, en geavanceerdere types laten zelfs toe om één of meerdere adresbits in te stellen via digitale IO-pinnen om meerdere sensors van hetzelfde type op één bus te kunnen combineren. I<sup>2</sup>C-adressen zijn 7 bits breed, en dus kan je maximaal  $2^7 = 128$  verschillende sensors (of andere devices) aansturen via één bus. Ongeacht het aantal devices op de bus heb je maar één paar pullupweerstand nodig.



### 2.3.2 SPI

Het voornaamste nadeel van I<sup>2</sup>C is dat de verbinding *half duplex* is, dat betekent dat je niet tegelijk data kan verzenden én ontvangen. Dat is bij I<sup>2</sup>C ook logisch vermits zowel input als output over dezelfde fysieke lijn (SDA) moeten. Om die reden is I<sup>2</sup>C niet erg geschikt voor applicaties waarbij data tegelijk in 2 richtingen moet getransporteerd worden, zoals bv. een SD-kaart die tegelijk bestanden moet kunnen lezen en schrijven.

SPI (*eng. Serial Peripheral Interface*) lost dit probleem op door de gecombineerde SDA-lijn van I<sup>2</sup>C te ontdebellen in een aparte lijn voor invoer en een andere voor uitvoer. De lijn van die data van de sensor naar de microcontroller transporteert heet MISO (Master Input, Slave Output) en omgekeerd heet de lijn die data van de microcontroller naar de sensor voert MOSI (Master Output, Slave Input). Op sensors heet dit dan SDO (Serial Data Output) en SDI (Serial Data Input) respectievelijk. Net zoals bij I<sup>2</sup>C heeft ook SPI een kloklijn, die SCLK heet (Serial Clock).

In plaats van devices te adresseren zoals I<sup>2</sup>C doet, maakt SPI gebruik van een extra verbinding om devices te selecteren. Deze extra lijn heet SS (Slave Select) en is gebruikelijk actief laag. Als er meerdere devices op de SPI-bus hangen dan zijn zijn alle SDO-lijnen met elkaar verbonden en ook alle SDI-lijnen met elkaar verbonden, maar enkel het device waarvan de SS-lijn laag is zal antwoorden op informatie van de microcontroller (master). Omdat er geen adressering meer nodig is dankzij de afzonderlijke SS-lijn en er afzonderlijke lijnen voor input en output voorzien zijn, kan SPI zeer grote datasnelheden bereiken. De state machine wordt er ook veel eenvoudiger mee, en doordat er slechts één slave tegelijk op de bus kan zijn er ook geen pullupweerstand meer nodig. De prijs die hiervoor betaald wordt is de 2 extra lijnen in vergelijking met I<sup>2</sup>C.

Sensors die SPI ondersteunen kan je aansturen vanuit Arduino via de SPI library. Bij sensors die zowel SPI als I<sup>2</sup>C ondersteunen heb je de keuze welke interface je kiest. Je kan op elk moment slechts één interface gebruiken, en vaak deelt de kloklijn voor beide interfaces (SCL respectievelijk SCLK) een gemeenschappelijke pin op de sensor. Welke interface je kiest hangt er dan vanaf of er al een bus aanwezig is in je design, hoeveel en welke pinnen op de Arduino je beschikbaar hebt, en hoeveel data je wil doorsturen.

Denk eraan dat de ATmega328p op je Arduino Uno een *single core* processor is die alle instructies sequentiëel uitvoert. Zelfs als je zowel een SPI-bus als een I<sup>2</sup>C-bus aan je Arduino hebt gekoppeld, zal je nooit beide tegelijk kunnen besturen.

# 3

## Actuators

“The fewer moving parts, the better. No truer words were ever spoken in the context of engineering.”

---

Christian Cantrell

### 3.1 Inleiding

In het vorige hoofdstuk werden sensors besproken als besturingselementen van de robot: ze verzamelen informatie over de omgeving, de toestand van de robot en commando's van de gebruiker. Deze informatie wordt vervolgens verwerkt door de ATmega328p microcontroller op de Arduino met als doel de toestand van de robot te veranderen en bij te sturen. In het wekkerproject van vorig jaar heb je reeds enkele actuators gebruikt, waaronder zevenssegmentdisplays om getallen weer te geven, LED's om toestanden te signaleren naar de gebruiker, en een buzzer om audiosignalen te genereren.

Om robots te laten bewegen zijn er actuators nodig die elektrische signalen omzetten naar mechanische bewegingen. De 2 voornaamste types actuators zijn solenoïdes voor lineaire bewegingen en motors voor roterende bewegingen.

### 3.2 Solenoïde

Een solenoïde is een spoel waarvan de middenkern ontbreekt. In plaats van deze middenkern wordt een beweegbaar anker toegepast als kern. Dit anker, gemaakt van een ferromagnetisch materiaal zoals staal of ferriet, wordt gebruikelijk half uit de spoel getrokken door een veer.

Wanneer de spoel wordt bekrachtigd, dan zal de elektrische stroom die erdoor vloeit een magnetisch veld opwekken dat het beweegbare anker in de spoel trekt. Dat anker doet nu dienst als kern, waardoor een stabiele toestand wordt gevormd die behouden blijft zolang er stroom door de spoel loopt.

Van zodra de spoel niet langer bekrachtigd wordt dan zal de veer het anker terug uit de spoel trekken waardoor de initiële toestand hersteld wordt.

Een solenoïde wordt in vrijwel alle toepassingen gebruikt als digitale actuator: ofwel bekrachtigd (anker aangetrokken) ofwel niet bekrachtigd (anker uitgeduwd). Solenoïdes worden zo toegepast om

andere onderdelen te vergrendelen of te ontgrendelen. Een typische toepassing is bijvoorbeeld het vergrendelen van een microgolfoven terwijl de microgolfbron actief is.

Ferromagnetische materialen worden aangetrokken door magnetische noordpolen zowel als zuidpolen. Het maakt dus niet uit in welke richting de stroom door de solenoïde loopt. Dat betekent ook dat je het anker er niet actief kan uitduwen, hiervoor heb je een andere solenoïde nodig die in de andere richting trekt, of een passieve kracht zoals een veer of contragewicht.

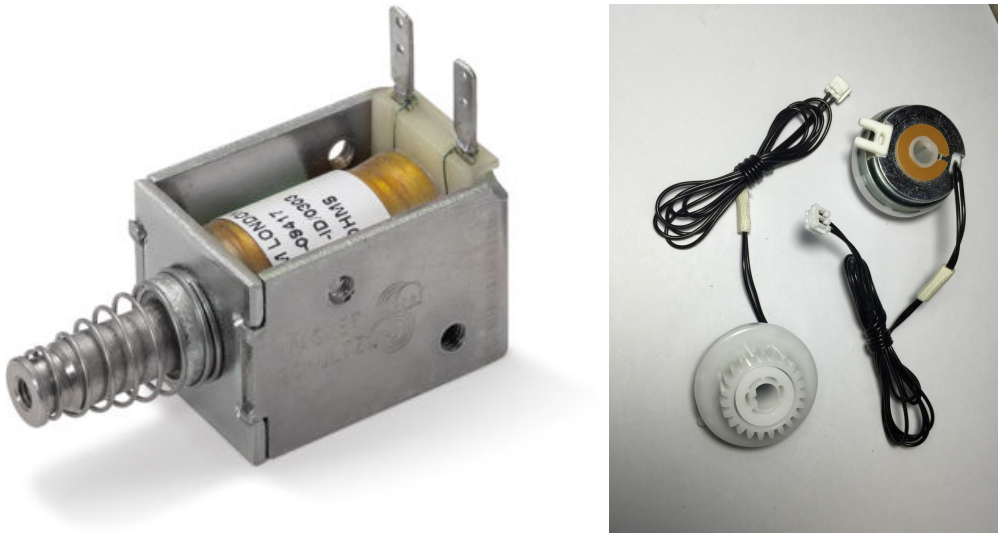
Als je feedback (terugkoppeling) gebruikt kan je de solenoïde ook gebruiken als lineaire actuator. Hiervoor moet je de gemiddelde stroom door de solenoïde nauwkeurig kunnen regelen, bijvoorbeeld via PWM. Het verband tussen de stroom en de positie volgt rechtstreeks uit de krachtenvergelijking op het anker. In een stilstaande positie is de nettokracht 0, en dus

$$kx - \frac{(NI)^2 \mu_0 A}{2g^2} = 0 \quad (3.1)$$

waaruit het verband tussen  $x$  en  $I$  volgt. Omdat  $A$  en  $g$  echter niet nauwkeurig gekend zijn voor commerciële solenoïdes zal je in de praktijk liever met feedback van een afstandssensor werken om de PWM duty cycle, en dus de positie, te regelen.

### 3.3 Clutch

Een clutch (of koppeling in het Nederlands) bestaat uit twee schijven op dezelfde as die vrij ten opzichte van elkaar kunnen bewegen. Op één van de schijven zijn permanente magneten aangebracht, op de andere spoelen. Zolang de spoelen niet bekrachtigd worden, oefenen ze geen kracht uit op de magneten en draaien de schijven met weinig weerstand. Wanneer er stroom door de spoelen loopt dan zullen ze zich aligneren met de magneten, en de uitgeoefende aantrekkingskracht verhindert dat de schijven nog langer onafhankelijk kunnen draaien. Om te verhinderen dat de aansluitingsdraden voor de schijf met de elektromagneten meedraaien (en dus opgerold geraken) worden ze meestal uitgevoerd met sleepringen zodat beide schijven kunnen draaien en de aansluitingen stationair blijven.



*Figuur 3.1: Typische solenoïde voor mechatronische toepassingen (links). De stang (vooraan) wordt actief in de spoel getrokken, en er weer uitgeduwd met een veer die langs de buitenzijde over de stang aangebracht is. Rechts: 2 clutches zoals deze in printers worden toegepast.*

Clutches voor mechatronische toepassingen hebben een perforatie in het center zodat ze over een as kunnen geschoven worden. De buitenkant is opgehangen rond de binnenkant via een rollager, en op de buitenkant zijn meestal tanden aangebracht om in een tandwieloverbrenging geïntegreerd te worden.

Ze zijn dus bijzonder handig om assen dynamisch met elkaar te verbinden, en zo roterende lasten aan- of af te koppelen. De fysieke grootte en dus ook de prijs hangt af van het koppel dat kan overgedragen worden.

Clutches zijn in tegenstelling to solenoïdes minder courant verkrijgbaar, maar vinden brede toepassingen in mechatronische apparaten zoals printers. Zoek je enkel een manier om een as te blokkeren, dan is een stappenmotor een gemakkelijker verkrijgbare en dus wellicht goedkopere oplossing.

### 3.4 Motors

De 4 motortypes waarmee je kennis zal maken in het Ontwerpproject zijn de stappenmotor (*eng. stepper motor* of kortweg *stepper*), de servomotor, de DC-motor, en de brushless DC motor (BLDC). Elk motortype heeft specifieke eigenschappen, voor- en nadelen, en vereist verschillende stuelelektronica. Je vindt de 4 types motors ook in je onderdelendoos.

#### 3.4.1 DC-motor

DC-motors bestaan uit een stator en een rotor. Op de rotor zijn koperwindingen aangebracht die een spoel vormen. Wanneer de motor wordt bekrachtigd, wordt de spoel gecommuteerd waardoor een magnetisch veld wordt opgewekt. Dit magnetisch veld stoot zich af tegen permanente magneten in de stator, waardoor de rotor begint te draaien. Doordat de commutatie via de as gebeurt en die van positie verandert, zal ook de polarisatie van het veld veranderen waardoor de motor steeds blijft verderdraaien en nooit een stabiele positie bereikt. In sommige uitvoeringen zijn de permanente statormagneten vervangen door elektromagneten om de prijs naar beneden te halen (koper is minder duur dan neodymi-ummagneten).

Omdat met de draairichting afhangt van de polarisatie van het magnetische veld, en de zin van dat veld wordt bepaald door de stroomzin, kan de draairichting van een DC-motor worden omgekeerd door de stroomzin om te keren.

Het voornaamste nadeel van een DC-motor is dat je die niet zomaar nauwkeurig kan positioneren. Je kan een robot met DC-motoren niet zeggen “rijd nu 10.5 cm rechtdoor en draai dan 90°”. Om dit te doen moet je sensoren aanbrengen op de as van je motor (of je wiel of de as van wat je maar aandrijft) om de controlelogica van feedback over de positie te voorzien. Dit is precies wat er gerealiseerd is binnenin een servo door een potmeter op de as te plaatsen.



*Figuur 3.2: Encoderschijf met light gate. De light gate bevat in één been een IR-LED en in het ander been een fototransistor (transistor die in geleiding gaat onder invloed van licht) of een LDR (weerstand waarvan de waarde afneemt onder invloed van licht (links) en een volledige assembly incl. lichtsensor en connector op PCB (midden en rechts).*

In het vorige hoofdstuk worden verschillende technieken besproken waarmee je dergelijke feedback zelf kan realiseren, bijvoorbeeld door een potmeter op de as te plaatsen. Je kan dus van een DC-motor een servo maken, door een potmeter op de as te monteren. Potmeters kunnen door hun bouw echter maar een beperkt aantal toeren draaien (normaal 1 rotatie, er bestaan er ook van 10 toeren). Een betere

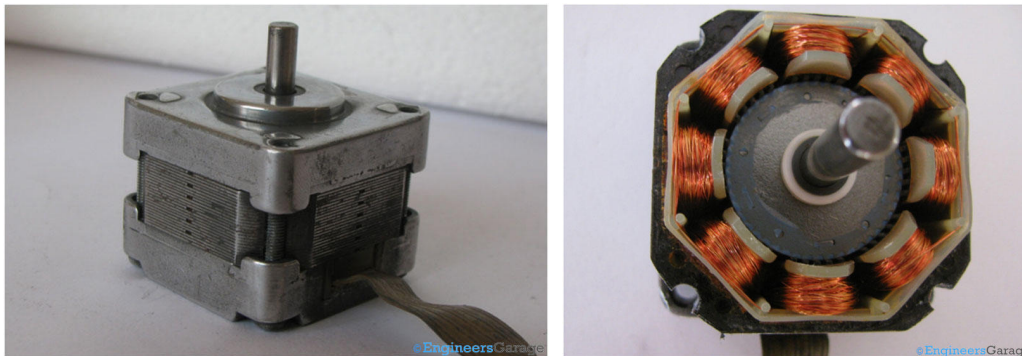
manier is om een sensor te gebruiken die eindelijk kan doordraaien. Een betere optie is daarom om rotationele encoders te gebruiken (zie figuur 3.2), en ken je misschien als je ooit al eens een printer opengegooid hebt.

Professionele servomotoren zien er dan ook niet uit als de hobby-servos die in de materiaaldoos van het Ontwerpproject zitten, ze kunnen wel meerdere rotaties doen en hebben rotationele encoders op hun as zitten. Je vindt die terug in duurdere CNC-machines omdat ze sneller/krachtiger zijn dan stappenmotoren. De CNC-machines in het fablab werken met stappenmotoren, maar de zware, groene Hartford-frees in labo Mechatronica gebruikt servodrives.

## 3.4.2 Stepper

### 3.4.2.1 Schematische werking

De rotor is een combinatie van permanente magneten rond een as. De stator (het stilstaande deel van de motor) bestaat uit een aantal elektromagneten. Elke keer één van deze elektromagneten bekrachtigd wordt (door er stroom doorheen te sturen) wordt de rotor aangetrokken door deze elektromagneet en draait de motor **één stap** verder. De volgende elektromagneet wordt bekrachtigd, en de volgende, ... en zo draait de motor rond.



Figuur 3.3: Uitzicht van een typische stepper (links) en opengewerkt exemplaar met 8 spoelen zichtbaar. Bron: Engineersgarage.com (<http://www.engineersgarage.com/insight/how-stepper-motor-works>)

In werkelijkheid zijn er in de rotor véél meer magnetische polen aanwezig, en zijn de elektromagneten in de stator onderverdeeld in meerdere spoelen. De meeste steppers hebben 200 onderverdelingen per rotatie, **dus 200 stappen / omwenteling**.

Dit laat toe om de motor bv. nauwkeurig een kwart slag te laten draaien (50 stappen).

Het nadeel is dat je er niet gewoon een batterij aan kan hangen om de motor te laten draaien zoals bij een DC. Je moet de verschillende spoelen van de motor gecontroleerd pulsen.

### 3.4.2.2 Single coil / full step / half step / microstepping

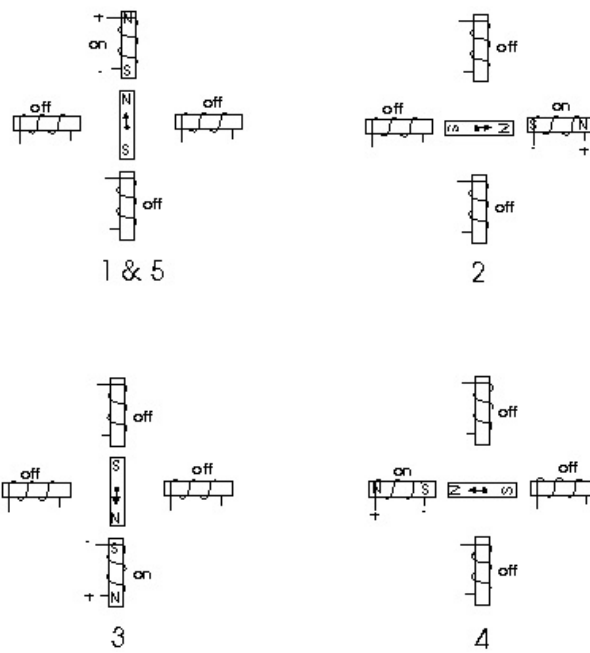
Door de spoelen in verschillende combinaties aan te sturen, kan je de nauwkeurigheid/het koppel van een stepper beïnvloeden:

**Wave drive:** er wordt één spoel er stap bekrachtigd.

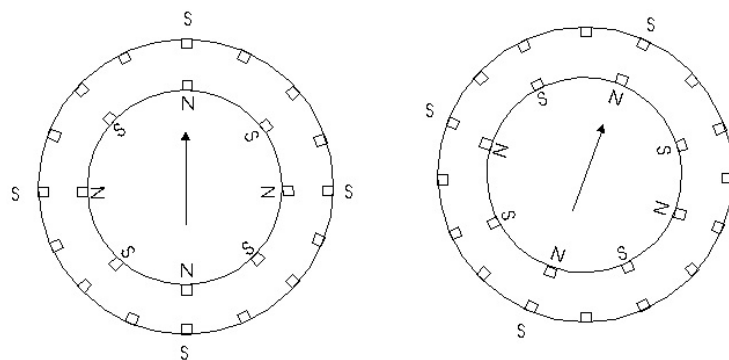
**Full drive:** door 2 spoelen per keer te bekrachtigen krijgt de motor meer koppel (bij steppers met 4 draden heb je niet deze optie, of eigenlijk altijd deze optie)

**Half drive:** door combinaties van 2 spoelen te bekrachtigen kan je het aantal stappen per rotatie van een stepper doen toenemen. Industriële stepper drivers doen tot  $1/16^e$  stap. Dit heet **microstepping**.

Meer info over verschillende manieren om steppermotoren aan te sturen: (single coil, full step, half step, microstepping) [http://www.pcbheaven.com/wikipages/How\\_Stepper\\_Motors\\_Work](http://www.pcbheaven.com/wikipages/How_Stepper_Motors_Work)



Figuur 3.4: Rotoatieprincipe van een stepper.



Figuur 3.5: Magneetparen zijn telkens een minimale afstand verschoven om een offset tussen magneten en elektromagneten in stand te houden.

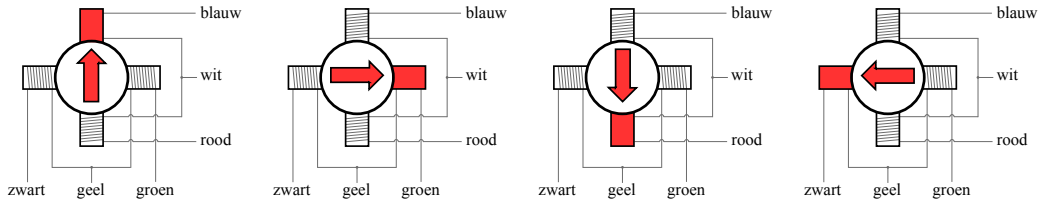
### 3.4.2.3 Aantal draden / types stepper

Sommige steppers hebben acht draden. Dit komt overeen met de 4 spoelen die je in de schemas op de vorige pagina's zag. (3e van links) Soms worden de uiteinden van elk paar spoelen samengenomen zoals in de linkse tekening. Je hebt dan 6 draden. Soms worden de spoelen per 2 samengenomen, zoals in de rechtse tekening. Je stuurt dan steeds 2 spoelen tegelijk aan = bipolar. Dit komt overeen met full-drive stepping.

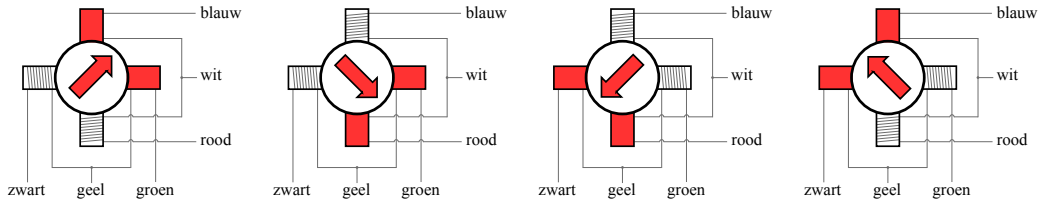
Wanneer je je steppers hebt, kijk op de behuizing voor het typenummer. Zoek aan de hand hiervan de datasheet op op de website van de fabrikant. Hier zal je een dergelijke schematische tekening van jouw type motor vinden.

### 3.4.3 Servo

Als je één van de servo's openmaakt vind je binnenin een DC-motor met een gearbox en een stuk elektronica. Als je verder graaft, vindt je een kleine potmeter ingebouwd in de as van de servo. Wanneer de servo aangestuurd wordt (tussen 0 en 180°) dan bekrachtigt de elektronica de DC-motor tot de as in de juiste positie staat (tot de **potmeter** de correcte waarde rapporteert). Op de as is meestal ook een



Figuur 3.6: Full wave sturing



Figuur 3.7: Half wave sturing

**mechanische stop** aangebracht, waardoor je ook met de hand niet verder dan  $180^\circ$  kan draaien.

Een servo is niks anders dan een DC-motor met opties. Dit betekent dat je van je servo's normale, continue DC-motoren kan maken. Een andere mogelijkheid is om de interne potmeter te vervangen door een externe, analoge sensor.

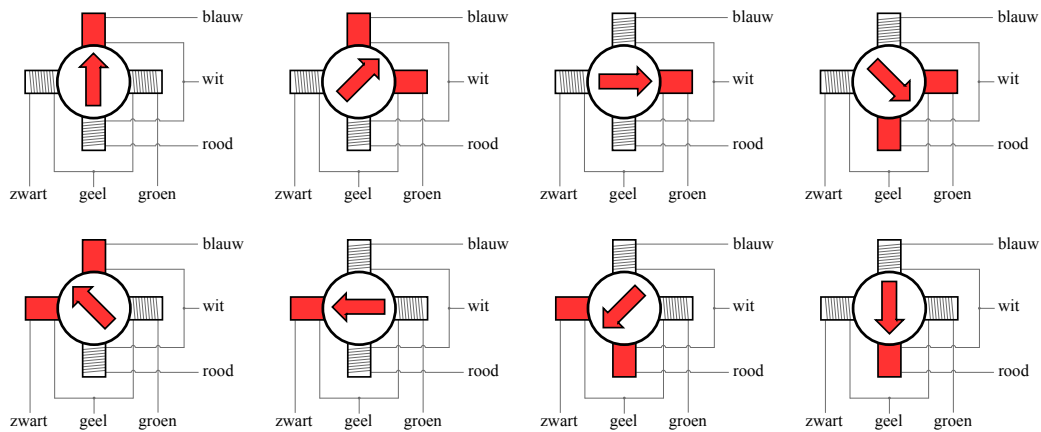
Servo's hebben 3 connecties die naar buiten gebracht worden in een header-compatibele connector: een aansluiting voor voeding (Vcc), grond (GND) en de sturingang die een pulsbreedtegemoduleerd signaal is (PWM). Arduino ondersteunt reeds een bibliotheek voor servo's die een PMW-sigitaal met de juiste frequentie en duty cycle kan genereren voor een integer input tussen 0 en 180.

Servo's verwachten een vrij nauwkeurige DC voedingsspanning tussen ca. 4.5 V en 5.5 V. Eén servo kan je dus nog net voeden uit een USB-poort samen met je Arduino, maar als je meerdere servo's wil gebruiken dan zal je hiervoor een afzonderlijke voeding moeten voorzien. Let erop dat elke servo zijn eigen buffercondensator (minstens  $100 \mu\text{F}$  @ 16 V) krijgt om storingen te onderdrukken.

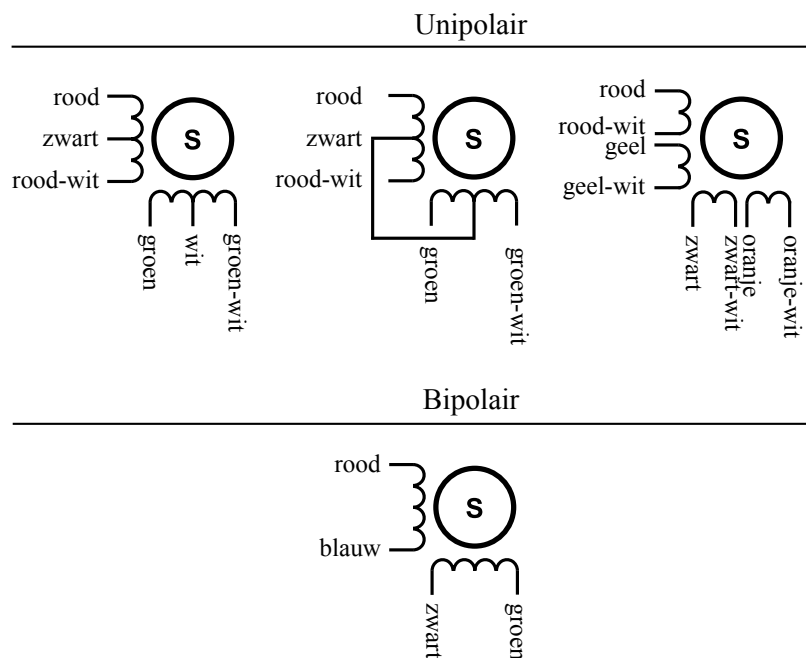
### 3.4.4 Brushless DC motor

Brushless DC motors, afgekort tot BLDC's, zijn het krachtigste motortype in je onderdelendoos. In tegenstelling tot wat de naam doet vermoeden, hebben ze qua opbouw meer gelijkenissen met steppers dan met DC-motoren. Ze hebben typisch echter veel minder polen en wikkelingen dan steppers, en zijn dus ook niet bedoeld om één stap met precies bepaalde hoek te verdraaien per puls. Doordat de polen fysiek verder uit elkaar zitten, is de hoekverdraaiing per puls veel groter dan bij steppers, en deze motortypes zijn dan ook bedoeld om continu te draaien. Een BLDC wordt net als steppers in beweging gebracht door een wisselend magnetisch veld aan te leggen op de polen, en hoe minder polen de motor heeft, hoe hoger de snelheden die hij zal kunnen bereiken. BLDC's met toerentallen tussen 10.000 en 30.000 RPM zijn geen uitzonderingen.

BLDC's zijn synchrone motoren waarbij de rotor bestaat uit permanente magneten (soms ook via variabele reluctantie of inductie), en de driverlogica ervoor is dus zeer vergelijkbaar met die van steppers. In feite zijn steppers een speciaal type BLDC. Aangezien bij BLDC's de precieze positie niet van belang is, worden de polen gegroepeerd in 3 fasen. Deze worden ofwel in ster of in driehoek aan elkaar gekoppeld om 3 contacten naar buiten te brengen. De configuratie in ster is veruit te meest voorkomende.



Figuur 3.8: Half drive sturing

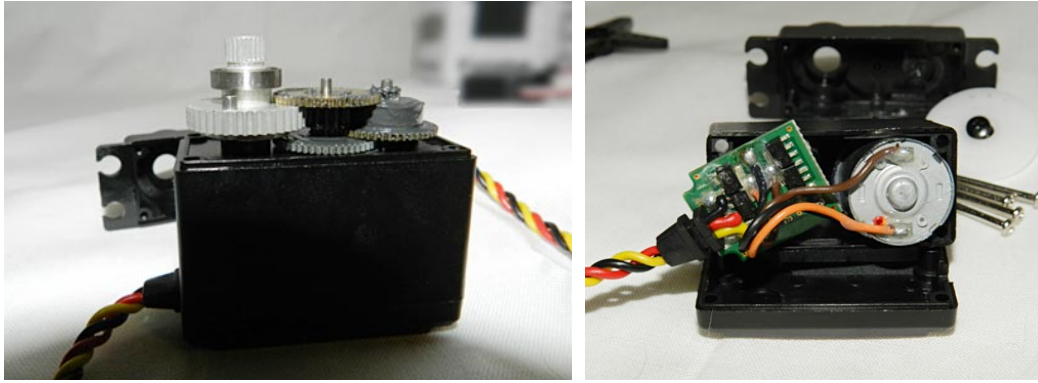


Figuur 3.9: Diverse types stappenmotors met 4 draden (bipolair) en 5, 6 of 8 draden (unipolair).

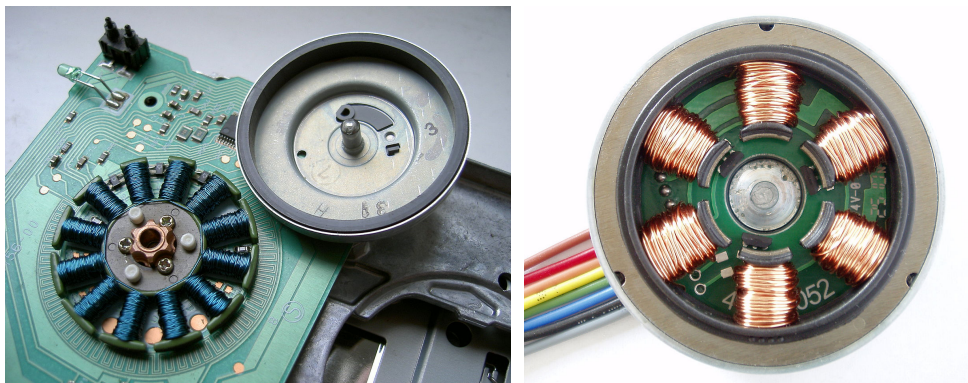
De stuursequentie van een stepper is relatief eenvoudig te implementeren omdat steppers stap voor stap roteren, en de rotatiehoek kan gemakkelijk berekend worden uit het aantal gemaakte stappen, vermits de hoekverdraaiing per stap precies gekend is. Na elke stap komt de stepper kort tot stilstand, waarna de volgende fasecombinatie wordt aangelegd om de volgende stap te maken. Daarom is het bij steppers gewoonlijk niet mogelijk om hoge rotatiesnelheden te halen. Bij BLDC's wordt dit opgelost door het aantal polen te verminderen, en de velden reeds te wisselen (= volgende fase bekrachtigen) nog voor de rotor zijn finale positie heeft bereikt. Om de velden op het juiste ogenblik te wisselen, moet de positie van de rotor echter gekend zijn. Het bepalen van deze positie kan via 2 manieren: ofwel op basis van Hallsensors die de statische velden van de rotor oppikken, ofwel via BEMF. Niet alle BLDC's hebben echter Hallsensors ingebouwd, en bij de types zonder geïntegreerde Hallsensors ben je wel verplicht om met BEMF te werken.

Net zoals voor steppers bestaan er ook voor BLDC's kant-en-klare controllers (drivers), waarmee je de kans krijgt te experimenteren in de 3<sup>e</sup> Bachelor. In de context van het Ontwepproject eisen we echter





Figuur 3.10: Gearbox van een hobby servo (links) en stuelelektronica met transistors van de H-brug zichtbaar (rechts).



Figuur 3.11: Brushless DC motor in een DVD-lezer (links) en van een hobby-quadcopter (rechts).  
Sebastian Koppehel, CC BY 3.0

dat je alle motorstuurelektronica zelf bouwt.

## 3.5 Bridge drivers

Vanuit een objectief perspectief zijn de DC-motors in hobby servo's eigenlijk zwak in vergelijking met de krachtige 12 V DC-motors die je in je Ontwerpprojectdoos aantreft. Deze motors zijn weliswaar iets groter dan servo's, maar het koppel dat ze kunnen leveren is een veelvoud van wat hobby servo's kunnen leveren. Wat maakt hobby servo's dan zo succesvol? Het antwoord schuilt in de combinatie van een flexibel mechanisch design en de eenvoudige sturing via PWM. Elke microcontroller kan PWM-signalen genereren, en dus hobby servo's aansturen. Om grotere DC-motors efficiënt te kunnen aansturen is er dus gelijkaardige driverlektronica nodig als in de hobby servo's zit, maar weliswaar op grotere vermogens berekend.

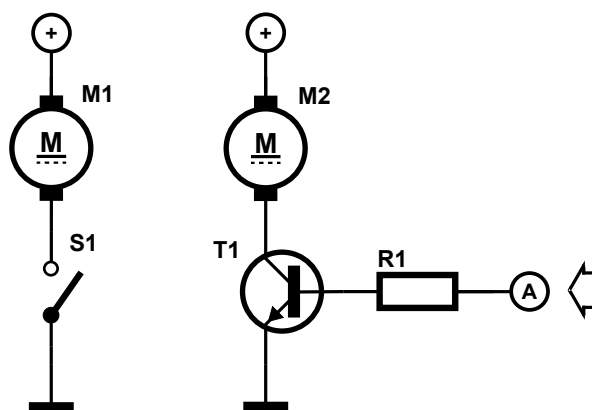
### 3.5.1 Gepolariseerde sturing

In het allereenvoudigste geval kan een DC-motor beschouwd worden als een invariante belasting  $R_L$ . Verbinding van die motor met een voedingsrail zorgt voor een spanning over de motor waardoor er een stroom door de windingen loopt en het resulterende elektromagnetische veld de motor laat draaien. Op een enkele uitzondering na, bijvoorbeeld een ventilator die continu actief moet zijn, is het ook wenselijk dat de motor kan aan- en uitgeschakeld worden. Dat kan mechanisch middels een schakelaar al dan niet gestuurd door een relais, maar interessanter is om de sturing op een elektronische manier te doen met componenten die geen mechanische onderdelen bevatten, zogenaamde *solid state* elektronica zoals transistors, MOSFET's, thyristors en triacs. De keuze voor deze solid state schakelaars heeft verschillende

voordelen:

- de schakelsnelheid kan veel hoger liggen dan bij mechanische schakelaars;
- er is geen slijtage van bewegende delen en dus langere levensduur;
- transitie tussen verschillende staten veroorzaken minder ruis en EMI.

Veronderstel dat je een motor aan- en uit wilt kunnen schakelen vanuit de Arduino. De ATmega328p microcontroller heeft een maximale uitgangstroom van 30 mA dus ongeacht interferentie en het bijhorende gevaar voor de ATmega328p is het niet mogelijk om motors rechtstreeks vanuit de ATmega328p aan te sturen. Zoals vorig jaar werd besproken in de practica *Basiselektronica* kunnen transistors uitstekend dienst doen als schakelementen: bipolaire junctietransistors zijn stroomversterkers en MOSFET's zijn spanningsgestuurde schakelaars. Beide zijn dus perfect geschikt om een belasting zoals een motor aan of uit te schakelen. Figuur 3.12 toont een het principe van solid state schakelaars in een voorbeeld waarbij een klassieke mechanische schakelaar vervangen wordt door een bipolaire junctietransistor. Merk op dat er een basisweerstand R1 nodig is om de stroom in de basis te begrenzen aangezien bipolaire junctietransistors stroomgestuurd zijn. Flybackdiodes zijn in dit voorbeeld weggelaten voor de duidelijkheid.



Figuur 3.12: Gepolariseerde sturing van een DC-motor: principe met schakelaar (links) en bipolaire junctietransistor als solid state schakelaar (rechts).

### 3.5.2 De H-brug

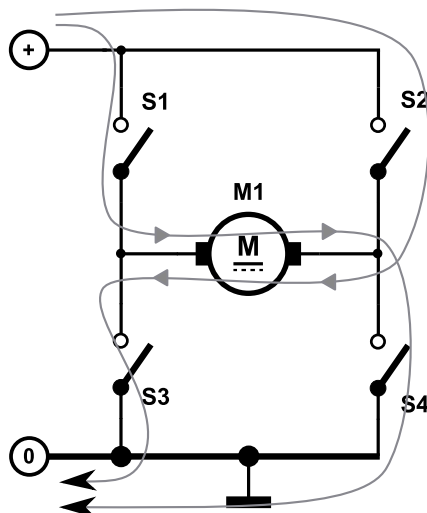
Om spoelen zowel als noordpolen en als zuidpolen te kunnen magnetiseren moet de stroomzin en dus ook de polarisatie van de spanning kunnen omgekeerd worden. Bij een DC-motor met aansluitingspunten A en B komt dat erop neer dat voor wijzerzin A aan VCC moet liggen en B aan de grond, en voor tegenwijzerzin net omgekeerd. Geen van beide verbindingpunten mag dus vast op VCC of grond liggen, want dat zou tot kortsluiting leiden. De enige mogelijke oplossing is gebruik te maken van 4 schakelaars, 2 langs elke kant die elk aansluitingspunt respectievelijk met VCC of GND kunnen verbinden, zoals in figuur 3.13 wordt getoond.

Met wat verbeelding lijkt de configuratie van de 4 schakelaars en de belasting op een hoofdletter “H”, vanwaar deze topologie de benaming “H-brug” haalt. De “H” is dus geen afkorting, maar slaat louter op de vorm van het schema.

#### 3.5.2.1 Principe

De 4 schakelaars zijn digitale componenten omdat ze maar 2 stabiele toestanden kennen, open of gesloten. In open toestand vormen schakelaars een open keten en kunnen geen stroom geleiden, in gesloten toestand gedragen ze zich als kortsluiting waarover geen spanning valt. Met 4 schakelaars die elk 2 toestanden hebben kan de H-brug dus in  $2^4 = 16$  verschillende toestanden worden geschakeld. Tabel 3.1 geeft de waarheidstabel voor deze H-brug.

Je merkt dat een aantal toestanden steeds terugkeert.



Figuur 3.13: Principe van een H-brug: 4 schakelaars die symmetrisch tegenover de voedingsrails rondom een belasting zijn geplaatst.

S1	S2	S3	S4	resultaat
0	0	0	0	motor in vrijloop
0	0	0	1	motor in vrijloop
0	0	1	0	motor in vrijloop
0	0	1	1	motor wordt geremd
0	1	0	0	motor in vrijloop
0	1	0	1	voedingsrail kortgesloten
0	1	1	0	motor bekrachtigd in tegenwoordse zin
0	1	1	1	motor geremd, voedingsrail kortgesloten
1	0	0	0	motor in vrijloop
1	0	0	1	motor bekrachtigd in voorwaardse zin
1	0	1	0	voedingsrail kortgesloten
1	0	1	1	motor geremd, voedingsrail kortgesloten
1	1	0	0	motor geremd
1	1	0	1	motor geremd, voedingsrail kortgesloten
1	1	1	0	motor geremd, voedingsrail kortgesloten
1	1	1	1	voedingsrail kortgesloten

Tabel 3.1: Waarheidstabel voor ingangscombinaties van een H-brug.

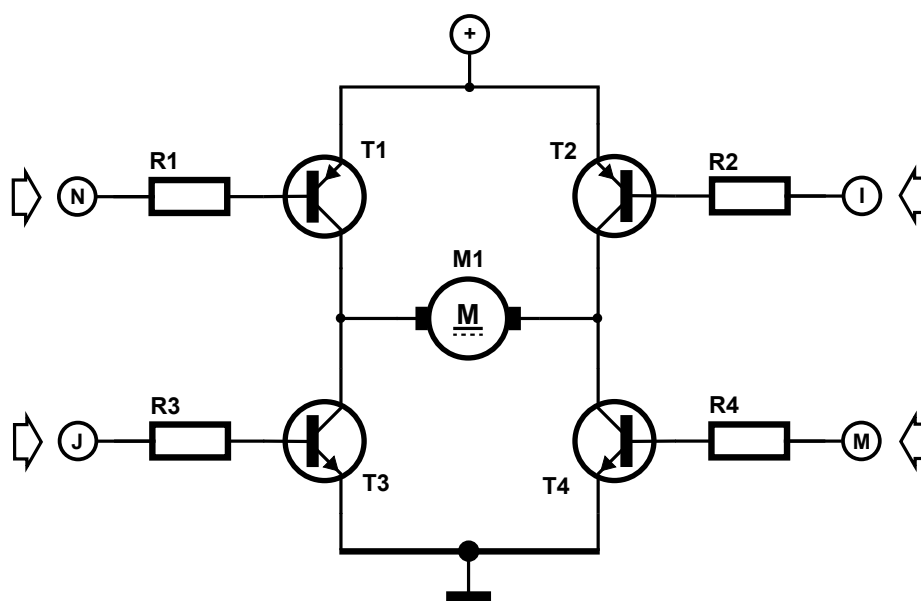
- **1 schakelaar of minder gesloten:** de motor is niet bekrachtigd en loopt vrij. Een draaiende motor zal blijven draaien door inertie.
- **bovenste 2 of onderste 2 schakelaars gesloten:** de motor wordt kortgesloten en daardoor geremd.
- **linkse 2 of rechtse 2 schakelaars gesloten:** de voedingsrail wordt kortgesloten en de motor wordt dus niet bekrachtigd.
- **overstaande schakelaars gesloten:** de motor wordt bekrachtigd.

Alle toestanden zijn nuttig behalve die waarbij de voedingsrail wordt kortgesloten. Er moet dus ten allen tijden vermeden worden dat op hetzelfde moment S1 en S3 samen gesloten worden, of S2 en S4. In vrijloop is de H-brug niet actief. Een onbelaste motor zal dus blijven draaien, en enkel afremmen door mechanische verliezen via wrijving. Een belaste motor zal uiteraard sneller tot stilstand komen dan een onbelaste motor. Als de motor wordt kortgesloten dan zal een draaiende motor geremd worden omwille van het generatoreffect: in onbekrachte toestand gedraagt de draaiende motor zich als generator en wekt daarbij een spanning op. Deze spanning wordt kortgesloten, en de resulterende kortsluitstroom

wekt in de spoelen een magnetische velden op die tegengesteld gepolariseerd zijn aan de velden die de motor in bekrachtigde toestand doet draaien. Dit type magnetische rem wordt op basis van Foucaultstromen (*eng. Eddy currents*) wordt o.a. toegepast als rem in rollercoasters.

Het is natuurlijk mogelijk om een H-brug te construeren met discrete mechanische schakelaars zoals in figuur 3.13 wordt weergegeven, en voor DC-motors kan dat in sommige gevallen een gemakkelijke oplossing zijn. De werking van stappenmotors berust echter op veranderende magneetpolarisaties, en bij elke stap moet de configuratie gewisseld worden. Mechanische besturing zou dus onbruikbaar traag zijn. Het is weliswaar mogelijk om de schakelaars te vervangen door relais, maar het constante geklik zou snel irritant worden en de levensduur van het relais verlaagt aanzienlijk. De enige haalbare optie is dus om de mechanische schakelaars te vervangen door solid state schakelaars, een principe dat hierboven al werd aangehaald bij gepolariseerde sturing.

Er zijn diverse solid state schakelcomponenten die in aanmerking komen: bipolaire junctietransistors, MOSFET's, thyristors (*eng. SCR, Silicon Controlled Rectifier*), triacs, IGBT's enzovoort. Thyristors en triacs bewijzen vooral hun nut bij AC-motorsturingen en vallen daardoor buiten de scope van deze cursus. IGBT's zijn veruit het duurst omdat ze bedoeld zijn voor zeer hoge spanningen, en behoren tot de vermogenselektronica waarover je de volgende jaren meer leert. In het Ontwerpproject wordt de scope beperkt tot bipolaire junctietransistors en MOSFETs, twee halfgeleidercomponenten die vorig jaar in de practica Basiselektronica al uitvoering besproken werden.



Figuur 3.14: H-brug opgebouwd uit 4 bipolaire junctietransistors: PNP-transistors aan high side en NPN-transistors aan low side.

Een H-brug opbouwen met bipolaire junctietransistors is nauwelijks moeilijker dan de aansturing van de ventilator in de practica Basiselektronica vorig jaar. Elke schakelaar in figuur 3.13 wordt vervangen door een transistor met bijhorende basisweerstand om de basisstroom te begrenzen. Je kiest deze basisweerstand afhankelijk van de applicatie zodanig dat de transistor gesatureerd wordt en er dus de minste spanningsval tussen collector en emitter optreedt (bij saturatie is  $V_{CE}$  minimaal). Figuur 3.14 geeft de resulterende schakeling weer.

Een H-brug wordt quasi altijd symmetrisch opgebouwd omdat het ontwerp daardoor aanzienlijk wordt vereenvoudigd. Met ontwerp wordt bedoeld dat de twee bovenste schakelementen, in figuur 3.14 transistors T1 en T2, hetzelfde type hebben. Analoog wordt ook voor de onderste twee steeds hetzelfde type gekozen. De schakelementen die met de positieve voedingsrail verbonden worden, en

dus de hoogste spanningen voeren, worden gerefereerd als de *high side* van de brug. Volgens dezelfde redenering heten de schakelelementen die met de grond of negatieve voedingsrail verbonden zijn de *low side*. Bij een symmetrische topologie zijn alle *high side* schakelelementen dus identiek en ook alle *low side* schakelelementen identiek.

Je merkt in figuur 3.14 dat voor de *high side* transistors PNP-types gekozen werden en voor de *low side* NPN-types. Dit is louter een kwestie van voorkeur tijdens het ontwerp, bij een H-brug met bipolaire junctietransistors is het ook perfect mogelijk om zowel voor *high side* als voor *low side* NPN- of PNP-transistors te kiezen. PNP-transistors aan *high side* en NPN-transistors aan *low side* vereenvoudigen de sturing wel enigszins omdat inputs N en J langs de linkerzijde en inputs I en M aan de rechterzijde telkens met elkaar verbonden kunnen worden. Met een logisch "laag" signaal zal de PNP-transistor aan *high side* in geleiding komen en zal de NPN-transistor aan *low side* sperren omdat zijn drempelspanning niet bereikt wordt. Bij een logisch "hoog" signaal is het net omgekeerd. De transistors aan *high side* en *low side* kunnen dus (theoretisch) nooit gelijktijdig in geleiding zijn waardoor de onwenselijke situatie uit tabel 3.1 – kortsluiting van de voedingsrails – vermeden wordt. Verder in dit hoofdstuk volgen nog verbeteringen op dit ontwerp.

Een H-brug met 4 bipolaire junctietransistors is betrouwbaar omdat geen enkele transistor in geleiding kan zijn als er geen basisstroom wordt geleverd. De 4 inputs kunnen dus zonder probleem open gelaten worden zonder dat er onvoorspelbare situaties (races) optreden. Dat is een voordeel als de brug bijvoorbeeld rechtstreeks gestuurd wordt uit een Arduino Uno, omdat de brug gewoon uitvalt als de Arduino wordt gereset of geheerprogrammeerd.

Het voornaamste nadeel van een H-brug met bipolaire junctietransistors is dat  $V_{CE}$  zelfs in saturatie een relatief grote spanning behoudt. Een stroom die van collector naar emitter loopt (of omgekeerd) zal dus een vermogen dissiperen in de transistor dat lineair afhankelijk is van  $V_E$  vermits  $P_T \approx P_{CE} = V_{CE} * I_{CE}$  waarbij  $I_{CE}$  ook de stroom is die door de motor loopt.

Rekenvoorbeeld: veronderstel een H-brug die is opgebouwd uit TIP31 NPN-transistors aan *low side* en TIP32-transistors aan *high side*. De karakteristieken van TIP31 en TIP32 liggen erg dicht bij elkaar hoewel ze elkaars tegengestelde zijn. Ze worden daarom *complementaire* transistors genoemd. Omdat de berekeningen voor *high side* dus ook op *low side* van toepassing zullen zijn en omgekeerd, wordt het ontwerp sterk vereenvoudigd. Volgens de datasheet van de TIP31 is  $V_{CE(sat)}$ , de collector-emitter saturatiespanning gelijk aan maximaal 1.2 V. Het absolute maximale vermogen dat gedissipeerd mag worden is eveneens volgens die datasheet 2 W. Omrekenen naar  $I_{CE}$  levert met een veiligheidsfactor van 0.3 dat  $I_{CE(max)} \approx 1.3$  A. Een stroom van 1.3 A is relatief veel voor DC-motors, maar weinig voor stappenmotors zonder stroombegrenzing. Bij 1.3 A zullen de transistors echt heet worden en volgens de datasheet een junctietemperatuur van ca. 125 °C halen. Koeling van dergelijk H-brug is dus een uitdaging op zich.

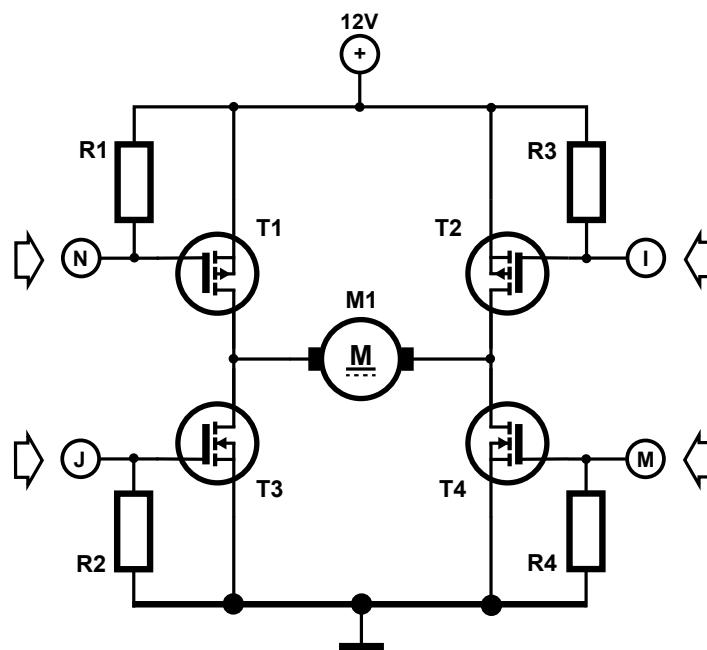
Daarnaast vermindert de spanningsval over de beide transistors ook de spanning over de motor, en dus het nuttige vermogen. Bij een voedingsspanning van 12 V valt er maximaal 1.2 V over elke transistor. Voor *high side* en *low side* samen is dat dus ongeveer 2.4 V. De spanning over de motor wordt dus gereduceerd tot 80 % van de waarde van een H-brug met ideale schakelelementen.

Een oplossing voor de excessieve spanningsval over de transistors kan gevonden worden door ze te vervangen door schakelelementen die geen minimale spanningsval hebben, namelijk MOSFET's.

### 3.5.2.2 MOSFET H-brug

Om een H-brug met bipolaire junctietransistors om te vormen naar een ontwerp met MOSFET's als schakelelementen is er weinig inspanning nodig: elke transistor kan afzonderlijk vervangen worden, waarbij ook de basisweerstand kunnen geëlimineerd worden vermits MOSFET's een hoogimpedante gate hebben waar geen noemenswaardige stroom in loopt. Als de gates opengelaten worden dan hebben ze helaas de neiging om lading afkomstig van elektromagnetisch ruis te accumuleren waardoor de

MOSFET's langzaam in geleiding komen. Dat zou betekenen dat de H-brug onvoorspelbaar gedrag vertoont als de gates niet verbonden worden met stuurlogica, bijvoorbeeld als de Arduino afgekoppeld wordt. Bij grote lasten kan dit potentieel gevaarlijk zijn omdat een motor opeens kan beginnen draaien zonder dat daar een duidelijke aanleiding voor is. Omdat te vermijden moet de gate naar een constante potentiaal getrokken worden, zonder dat sturing daardoor onmogelijk wordt. Hiervoor worden pullup of pulldown weerstanden met een grote waarde, d.w.z. hoger dan  $47\text{ k}\Omega$  gekozen die een zwakke pullup of pulldown vormen. De resulterende schakeling is getoond in figuur 3.15.



Figuur 3.15: H-brug opgebouwd uit 4 MOSFET's: P-MOSFET's aan de high side en N-MOSFET's aan de low side.

Anders dan bij bipolaire junctietransistors hebben MOSFET's geen minimale  $V_{CE(sat)}$ , zoals je weet uit de practica Basiselektronica wordt de geleiding van MOSFET's en JFET's uitgedrukt als een impedantie in  $\Omega$ . Als de MOSFET in geleiding is, dan is de weerstand tussen drain en source gelijk aan  $R_{DS(on)}$ . Levert dit een verbetering op?

Rekenvoorbeeld: beschouw een H-brug opgebouwd met complementaire MOSFET's in de vorm van IRF630 N-MOSFET's aan *low side* en complementaire IRF9630 P-MOSFET's aan *high side*. Volgens de datasheet is  $R_{DS(on)}=0.4\ \Omega$  bij  $V_{GS} = 10\text{ V}$ . Voor hetzelfde maximumvermogen van  $2\text{ W}$  bedraagt de maximale stroom met veiligheidsfactor 0.3 nu  $0.7\sqrt{\frac{2}{0.4}} = 1.7\text{ A}$ . Dit is dus een verbetering van 30% in vergelijking met het ontwerp met bipolaire junctietransistors van figuur 3.14. MOSFET's zijn bijgevolg dus de beste keuze als schakelementen voor een H-brug, maar ze hebben als nadeel dat ze veel gevoeliger zijn aan allerlei storingen en overbelastingen dan bipolaire junctietransistors.

### 3.5.2.3 Level shifting

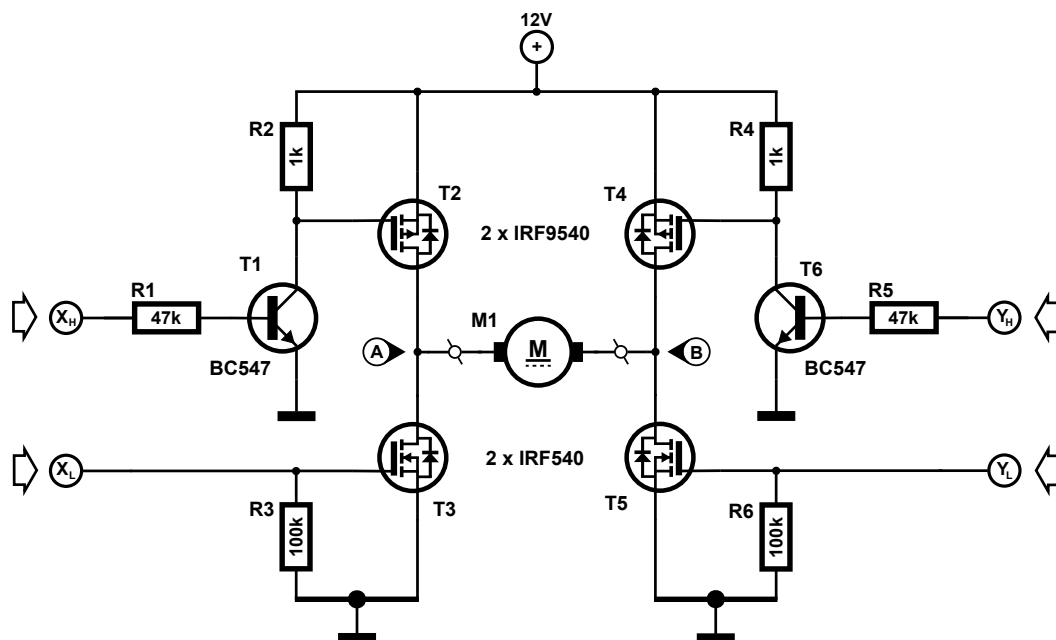
Bekijk de schakeling in figuur 3.15 aandachtig. T3 is een N-MOSFET en kan zonder probleem rechtstreeks aangestuurd worden vanuit digitale stuurlogica op 5 V. Een logische "0" op sturingang J betekent  $V_{GS}=0$  en dus zal de MOSFET sperren, een logische "1" betekent resulteert in  $V_{GS}=5\text{ V}$  wat voor alle moderne MOSFET's hoger is dan  $V_{GS(th)}$  (de drempelspanning), en dus wordt de MOSFET in geleiding gebracht.

Aan de *high side* is de situatie complexer. P-MOSFET T1 gedraagt zich op een identieke manier

als T3, en kan in geleiding gebracht worden als  $V_{GS} \geq V_{GS(th)}$ . De source zit echter op voedingsspanning, in het voorbeeld 12 V. Bij een logische “0” op sturingang N resulteert in  $V_{GS} = -12$  V en de MOSFET zal dus in geleiding gebracht worden. Dat is omgekeerd aan het gedrag van een N-MOSFET omdat alle logica omgekeerd is, en dus te verwachten. Bij een logische “1” op sturingang N is de spanning op de gate 5 V. In dat geval is  $V_{GS} = (5 - 12) = -7$  V. Dat is voor moderne MOSFET’s nog steeds ruim meer dan  $V_{GS(th)}$  die gewoonlijk tussen 2 V en 4 V ligt voor N-MOSFET’s en -2 V en -4 V voor P-MOSFET’s. Ongeacht welk signaal er op sturingang N wordt aangelegd zal T1 dus altijd in geleiding zijn, wat uiteraard niet de bedoeling is.

Idealiter moet de spanning op N ook kunnen variëren tussen 0 en 12 V. Dat kan opgelost worden met een versterker met versterkingsfactor  $A = \frac{12}{5} = 2.4$ . Elke stuurspanning wordt dan proportioneel vermenigvuldigd met een factor 2.4. Zoals je weet uit de theorie cursus Basiselektronica is het ontwerp van versterkerschakelingen een omslachtige bedoening, en een goede elektronicus besteedt die tijd liever door enkele minuten langer na te denken bij een hete kop koffie of thee, op zoek naar shortcuts die het ontwerp vereenvoudigen.

In dit geval bestaat er effectief zo’n shortcut. De versterkerschakeling is namelijk een algemeen geldig geval, waarbij elke spanning in een bereik van 0 tot 5 V proportioneel wordt versterkt naar een bereik tussen 0 en 12 V. Dat is echter niet nodig vermits het stuursignaal geen analoge waarde is maar een digitale, en dus maar 2 toestanden kent: hoog en laag. Dat betekent dat de transistor die de versterking doet niet hoeft gebiased te worden, maar meteen van saturatie naar cut-off kan gestuurd worden en omgekeerd. Alle biasweerstand vallen dus weg, er is geen emitterdegeneratie of feedback meer nodig, en door de basisweerstand klein genoeg te nemen kan verzekerd zijn dat de transistor in “harde” saturatie gestuurd wordt. Daarmee hoeft er dus niets uitgerekend te worden! Figuur 3.16 toont de resulterende schakeling.



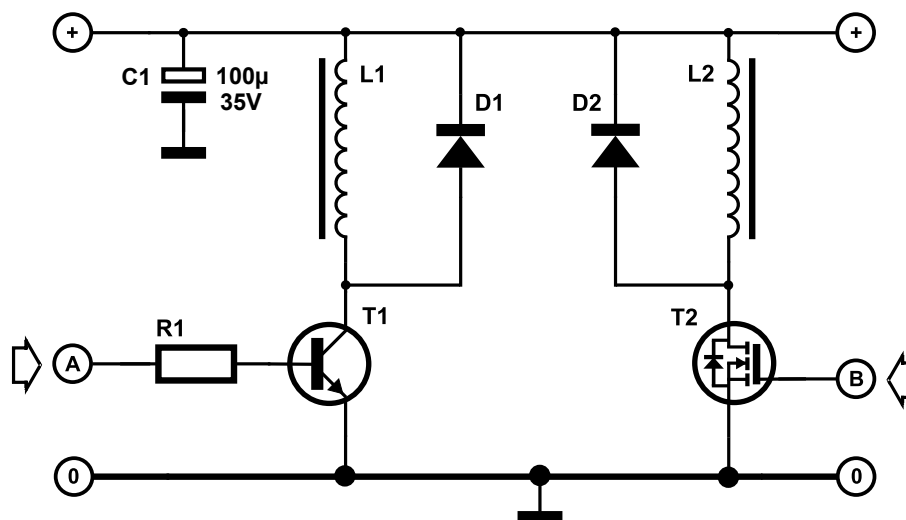
Figuur 3.16: H-brug opgebouwd uit 4 MOSFET’s: P-MOSFET’s aan de high side en N-MOSFET’s aan de low side.

Zoals je weet uit de theorielessen Basiselektronica resulteert een entrapsversterker in een fasedraaiing van  $180^\circ$ , of in dit vereenvoudigde geval gedraagt hij zich als digitale inverter. Om de *high side* P-MOSFET in geleiding te brengen moet nu een logisch “hoog” signaal aangelegd worden op sturingangen  $X_H$  en  $Y_H$ . Dit is consistent met de *low side* sturingangen.

### 3.5.2.4 Flyback

Vanuit een elektronisch perspectief zijn motors ongeacht hun type lastige componenten om aan te sturen omdat ze zelf spanningen kunnen opwekken. Die zijn terug te brengen tot twee compleet verschillende oorzaken: de *zelfinductie* enerzijds en het *generatoreffect* anderzijds.

Zelfinductie is een fundamentele eigenschap van elke spoel. Net zoals condensators de neiging hebben om spanningen uit te middelen, zo werken spoelen stroomveranderingen tegen. Een spoel werkt een toenemende stroomsterkte tegen door als grote impedantie op te treden, en werkt afnemende stroomsterkte tegen door zich als stroombron te gedragen. Dit gedrag wordt beschreven door de uitdrukking  $U_L = L \frac{di}{dt}$  waarbij  $L$  de zelfinductiecoëfficiënt van de spoel is. Als de stroom door de spoel afneemt, dan zal de spoel dit tegenwerken door een deel van de energie die opgeslagen is in de vorm van een magnetische flux om te zetten naar een stroom. Wanneer de stroom door een spoel opeens wordt afgesneden, wanneer bijvoorbeeld een MOSFET die de sturing verzorgt uit geleiding wordt gebracht, dan wordt de factor  $\frac{di}{dt}$  opeens erg groot vermits  $di$  van een positieve waarde wordt herleid tot  $\approx 0$ . Het resultaat is een kortstondig zeer hoge spanning  $U_L$  over de spoel. Halfgeleiders zoals bipolaire junctietransistors en MOSFET's zijn daar niet tegen opgewassen en zullen onherroepelijk beschadigd geraken. Een oplossing hiervoor is gebruik te maken van een vrijlooptdiode (eng. *flyback diode*) die de door de spoel zelf opgewekte spanning kortsluit van zodra ze optreedt. Figuur 3.17 toont het principe voor een inductieve last die gestuurd wordt met een MOSFET en met een bipolaire junctietransistor.



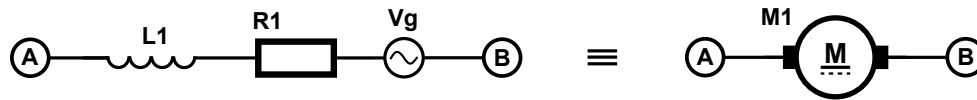
Figuur 3.17: Bescherming van halfgeleiderschakelaars met vrijlooptdiodes.

In de schakelingen in figuur 3.17 loopt de stroom steeds van  $V_{cc}$  naar grond, dus van de bovenste voedingsrail naar de onderste. In normale werkingsmodus is de spoel een verbruiker, en dus staat de spanning per conventie omgekeerd aan de stroomzin en dus naar boven gericht. De vrijlooptdiodes sperren in dat geval. Wanneer de transistor resp. MOSFET wordt uitgeschakeld dan neemt de spanning over de spoel fors toe en gedraagt de spoel zich als bron. De spanningszin wordt dus gelijk aan de stroomzin, en de vrijlooptdiodes worden nu voorwaarts gepolariseerd waardoor de spanning over de spoel wordt begrensd (eng. *clamped*) op de drempelspanning van de diode. De diodes moeten in staat zijn om een vrij behoorlijke stroompiek op te vangen. Die eigenschap wordt in diodedatasheets gedefinieerd als *repetitive peak forward current*.

Het generatoreffect heeft een gelijkaardig resultaat maar een volledig verschillende oorzaak. Waar zelfinductie een eigenschap van elke spoel ongeacht de constructie is, treedt het generatoreffect enkel op in elektrische motoren met permanente magneten. Wanneer de motor niet bekrachtigd wordt maar onder invloed van een extern koppel of inertie toch draait, dan zal in spoelen die aan een veranderende magnetische flux worden blootgesteld (door rond te draaien in een magnetisch veld bijvoorbeeld) een spanning



worden opgewerkt. Aan het vervangschema van dergelijke motors is dus een variabele spanningsbron toegevoegd die het generatoreffect vertegenwoordigt, zoals in figuur 3.18 weergegeven. Het is duidelijk dat zelfinductie in de spoel en generatoreffect 2 compleet verschillende en los van elkaar staande verschijnselen zijn. Het voornaamste probleem met het generatoreffect is niet zozeer de hoge spanningen, maar de AC-spanning die geproduceerd wordt. Bij negatieve spanning is het mogelijk dat ongewenste parasitaire juncties in halfgeleidercomponenten onbedoeld voorwaarts worden gepolariseerd waardoor er stroom door gaat lopen, en ze beschadigd geraken. Een voorbeeld hiervan is latch-up bij MOSFET's.



Figuur 3.18: Equivalent schema van een elektrische motor met permanente magneten.

Om de negatieve consequenties van beide effecten te verhinderen moeten alle schakelementen dus steeds beveiligd zijn met vrijlooptiodes. Figuur 3.19 toont een verbeterde versie van de H-brug uit figuur 3.16 waaraan o.a. 4 vrijlooptiodes zijn toegevoegd (D2, D3, D5 en D7). Deze vrijlooptiodes moeten voldoende snel zijn om te kunnen reageren op de omkerende spanningszin zonder dat de spanning te hoog oploopt. Hiervoor moeten dus snelle diodetypes zoals Schottkydiodes of lawinediodes (*eng. avalanche diode*) gekozen worden, zoals hier Schottkydiodes van type 1N5817. Bij gebruik van trage(re) diodes is een extra condensator over de inductieve last aangewezen om de spanningspiek te vertragen, C2 in figuur 3.19. Deze condensator helpt bij motors uiteraard ook om de  $\cos \phi$  (= power factor, PF) te verbeteren.



Zoals te zien is in het MOSFET-symbool bevatten MOSFET's reeds ingebouwde diodes tussen drain en source die identiek gepolariseerd zijn aan de extern toegevoegde 1N5817 vrijlooptdiodes. Vaak wordt hier de fout gemaakt om de vrijlooptdiodes weg te laten in de veronderstelling dat de interne diodes deze taak wel zullen vervullen. Dat is echter **niet** correct. De ingebouwde diodes hebben tot doel om te verhinderen dat de pn-junctie tussen een p-kanaal en een n-gate of omgekeerd voorwaarts gepolariseerd wordt en er daardoor stroom vanuit het kanaal in de gate kan lopen. Deze diodes zijn dus geoptimaliseerd op drempelspanning maar niet op snelheid, en ze zijn ook niet bedoeld om de grote piekstromen die door inductieve lasten worden gegenereerd te verwerken. Je hebt dus nog steeds externe vrijlooptdiodes nodig.

Om te verhinderen dat te hoge spanningspieken op de voedingsrail terecht komen moet dicht bij elke H-brug ook een tankcondensator voorzien worden die deze energie tijdelijk kan opslaan. In figuur 3.19 vervult C1 deze functie.

### 3.5.2.5 Clamping

Dat vrijlooptdiodes bij inductieve lasten noodzakelijk zijn voor de begrenzing (clamping) van de drain-sourcespanning is algemeen geweten. Nochtans ligt de maximale drain-sourcespanning ( $V_{DS(max)}$ ) bij moderne MOSFET's vrij hoog. Bij een IRF540 N-MOSFET is dit bijvoorbeeld 100 V en bij een IRF630 N-MOSFET zelfs 200 V. De spanning tussen gate en source, die bepalend is voor de geleiding van de MOSFET, heeft echter heel wat lagere maximumspanningen. Bij een IRF630 bedraagt de absolute maximumspanning tussen gate en source,  $V_{GS(max)} = \pm 20$  V. Bekijk het schema in figuur 3.16 met deze maximumwaarde in gedachten, in de veronderstelling dat de H-brug gevoed wordt met een DC-spanning van 24 V. Dat is een standaardspanning in CNC-applicaties zoals 3D-printers. Voor de IRF540's aan *low side* is er geen enkel gevaar, want zij worden rechtstreeks gestuurd met een digitaal signaal tussen 0 en 5 V. In dat geval is  $V_{GS} \leq 5$  V wat ruim onder de maximaal toegelaten spanning is.

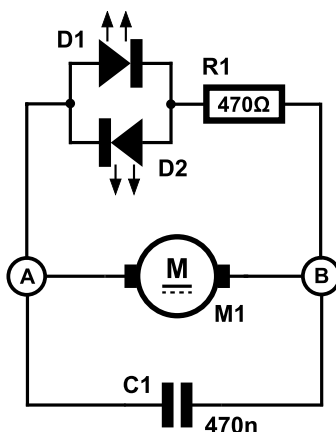
Als de *high side* is uitgeschakeld, en transistors T1 en T6 niet in geleiding zijn, dan worden de gates van T2 en T4 opgetrokken door resp. R2 en R4 die als pullupweerstand fungeren naar de 24 V voedingslijn. In dat geval is  $V_{GS}$  voor beide P-MOSFET's 0 en is er dus evenmin een probleem. Vanaf het ogenblik dat T1 of T6 in geleiding komen loopt het echter fout. De gates worden in dat geval naar beneden getrokken tot een spanningsniveau dat dicht bij de grond ligt omdat  $V_{CE(sat)} \approx 1.2...2$  V. In dat geval is voor de P-MOSFET's  $V_{GS} \approx -22$  V wat de absolute maximumwaarde overschrijdt en dus de MOSFET's beschadigt of in ieder geval hun levensduur sterk vermindert.

Om de H-brug te kunnen gebruiken in configuraties met voedingsspanningen boven 20 V is het dus noodzakelijk om voor de *high side* MOSFET's  $V_{GS}$  te begrenzen tot maximaal 20 V. Een gekende manier om dit soort clamping te implementeren is door gebruik te maken van een zenerdiode met een zenerspanning die overeenkomt met de spanning waarop begrensd moet worden. Als de spanning over de zenerdiode deze waarde overschrijdt dan zal de diode gaan geleiden waardoor de spanning niet verder kan toenemen. Bij gebruik van een zenerdiode van 15 V tussen gate en source bijvoorbeeld zou  $V_{GS}$  nooit onder -15 V kunnen zakken waardoor er niet langer gevaar dreigt.

Het oplossen van het ene probleem brengt echter een nieuw probleem met zich mee. Wanneer de zenerdiode in geleiding komt dan daalt de impedantie sterk waardoor de stroom erdoor fors zal toenemen als er geen andere componenten zijn die deze stroom beperken. Deze stroom is ook de collectorstroom van resp. T1 of T6. De impedantie van deze geleidende transistors is ook laag vermits ze gesatureerd zijn, en de stroom door de diode en transistor zal bijgevolg sterk oplopen totdat ofwel de zenerdiode ofwel de transistor uitbrandt. Dit is uiteraard geen wenselijke situatie. Een oplossing hiervoor is de stroom in het diode-transistor netwerk te limiteren door een extra weerstand toe te voegen. Figuur 3.19 toont een mogelijk oplossing. Bij een voedingsspanning van bv. 24 V zal D1 geleiden als  $X_H$  logisch hoog is. Transistor T1 wordt dan in saturatie gestuurd. Volgens de datasheet is  $V_{CE(sat)} = 250$  mV. Met R2 als stroombegrenzing wordt de totale stroom in de tak nu  $i_{D1} = i_{CE} = \frac{24\text{ V} - 12\text{ V} - 250\text{ mV}}{10\text{ k}\Omega} \approx 1\text{ mA}$  wat ruim acceptabel is.

### 3.5.2.6 Monitoring

H-bridgen worden in de regel digitaal gestuurd, met mogelijk uitzondering van PWM-gestuurde H-bridgen. LED's zijn dus uitstekend geschikt om de werking van de H-brug visueel te monitoren. Praktisch kan dit door 2 LED's antiparallel met de motor te schakelen zodat over de LED's dezelfde spanning valt als over de Motor. Afhankelijk van de polariteit (= draaizin bij DC-motors) zal één van beide LED's oplichten. Als geen van beide LED's oplicht en staat er geen of onvoldoende spanning over de motor, bijvoorbeeld als de brug uitgeschakeld is. Figuur 3.20 toont de configuratie.



Figuur 3.20: LED-configuratie voor monitoring van een H-brug.

Uiteraard bereken je de serieweerstand voor de LED's op basis van de spanning over de motor zodanig dat de stroom door de LED's ca. 5 mA - 10 mA bedraagt.

### 3.5.2.7 Logische sturing

Hoewel het ontwerp in figuur 3.19 elektronisch gezien perfect bruikbaar is, is er een belangrijk nadeel aan verbonden. De 4 MOSFET's moeten afzonderlijk gestuurd worden, wat erop neerkomt dat er per H-brug 4 digitale pinnen op de Arduino worden ingenomen. Voor een stappenmotor die 2 H-bridgen vereist, zijn dat zelfs 8 digitale pinnen. Afhankelijk van de topologie die je voor ogen hebt kan dit zowel een probleem als een opportuniteit zijn.

De afzonderlijke sturing van zowel *high side* als *low side* onafhankelijk van elkaar betekent ook dat op elk willekeurig moment *high side* en *low side* tegelijk kunnen ingeschakeld worden. Dit leidt tot kortsluiting van de voedingsrails, en het doorbranden van de zekering. Als je geen zekering voorzien hebt, dan brandt de zwakste schakel in de elektronische ketting door, wat meestal de draden op je breadboard zijn. Er is dus ook nood aan een systeem om te verhinderen dat *high side* en *low side* per ongeluk gelijktijdig kunnen geactiveerd worden.

Voor deze 2 problemen kan er zowel een software als een hardware-oplossing bedacht worden. De eenvoudigste manier om dit probleem in software aan te pakken is een microcontroller te voorzien die bij in de H-bridgschakeling wordt geïntegreerd en enkel als taak heeft om de 4 MOSFET's te sturen zonder dat ze kortsluiting maken. De microcontroller zelf kan dan digitaal gestuurd worden. De mogelijkheden in tabel 3.1 zijn op te delen in 4 situaties:

Om 4 combinaties te coderen zijn er  $\log_2 4 = 2$  bits nodig, in tabel 3.2 respectievelijk D0 en D1. In software is het weinig moeite om D0 en D1 te mappen op digitale ingangspinnen van een microcontroller en op basis daarvan de 4 uitgangscombinaties voor de 4 MOSFET's op te zoeken in een tabel. Een microcontroller voor dit doel inzetten lijkt misschien *overkill* maar is dat geenszins, in tegendeel: het bevordert het modulaire karakter van je robotontwerp, maakt het minder gevoelig voor fouten, en decentraliseert taken zodat je centrale microcontroller minder belast wordt. Een microcontroller biedt

D0	D1	status
0	0	motor uitgeschakeld
0	1	motor draait in wijzerzin
1	0	motor draait in tegenwijzerzin
1	1	motor wordt actief geremd

Tabel 3.2: Mogelijke ingangscombinaties voor een intelligent gestuurde H-brug.

nog andere mogelijkheden, je zou bijvoorbeeld meteen de volledige stuursequentie van een stappenmotor kunnen coderen en tegelijk 2 H-bruggen sturen. Je hebt dan effectief je eigen stepper driver gebouwd.

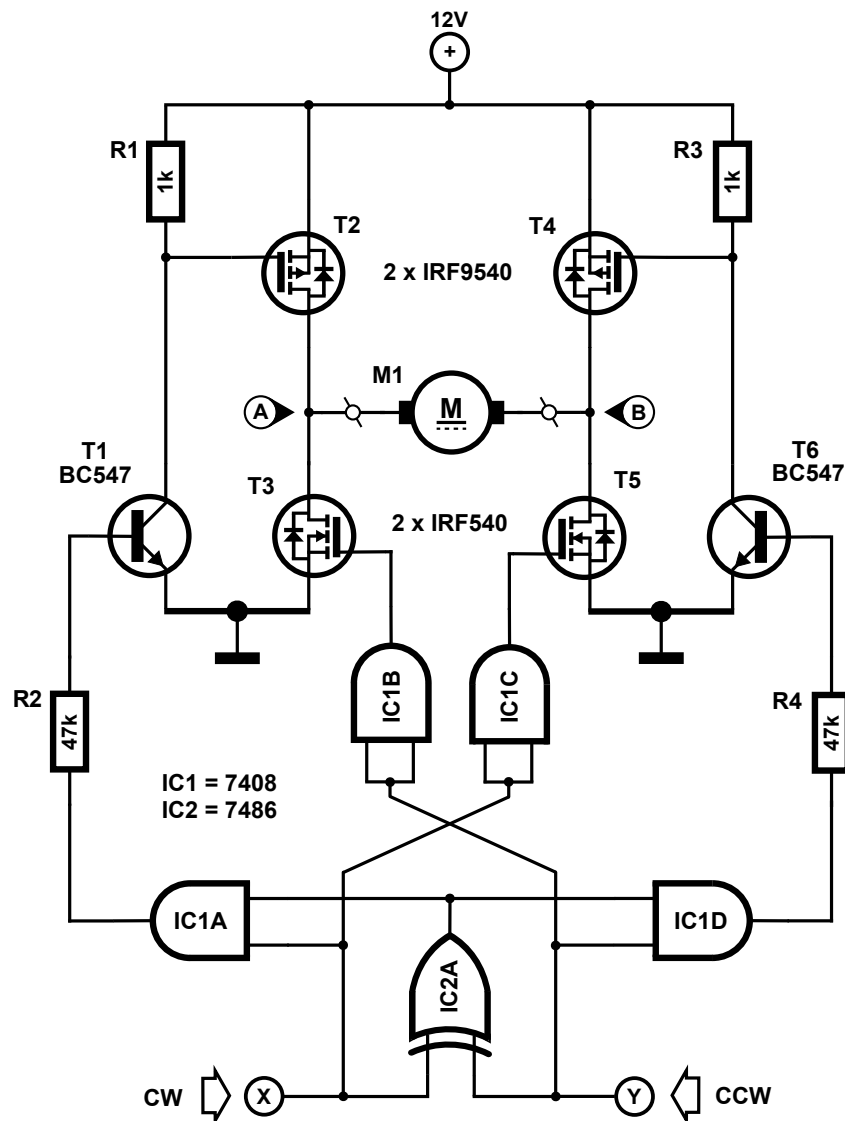
Een alternatieve methode is gebruik te maken van hardware. Het verband tussen de ingangscombinaties in tabel 3.2 is zodanig simpel dat de corresponderende combinaties voor de 4 MOSFET's direct kunnen afgeleid worden. Dit kan via een Karnaughmap, maar nog gemakkelijker door logisch te redeneren. Bij combinatie "00" moeten alle MOSFET's uitgeschakeld zijn, wat overeenkomt met de 4 stuursignalen (alle 0). Bij "01" of "10" moet respectievelijk  $X_H$  en  $Y_L$  of  $X_L$  en  $Y_H$  geactiveerd worden. Ook deze combinaties zijn triviaal omdat de 2 stuurbits overeenkomen met de stuursignalen voor overstaande MOSFET's. De enige problematische combinatie is "11", waarbij de motor moet geremd worden. De combinatie "11" kan gezien worden als een logische OR van "01" en "10" met als gevolg dat de 4 MOSFET's tegelijk worden ingeschakeld en er dus dubbel kortsluiting wordt gemaakt. Deze ingangscombinatie moet dus gedetecteerd en afgehandeld worden. Figuur 3.21 detecteert de "11" ingangscombinatie via een XOR-poort. Bij ingangscombinaties "01" en "10" is het resultaat van een XOR operatie steeds logisch '1' waardoor de AND-poorten IC1A en IC1D stuursignalen doorlaten en de *high side* MOSFET's kunnen ingeschakeld worden als respectievelijk ingang X of Y logisch '1' is. Als beide ingangen logisch '1' zijn dan resulteert de XOR-operatie in een logische '0' en worden geen van beide *high side* MOSFET's ingeschakeld. Beide *low side* MOSFET's worden echter wel gelijktijdig ingeschakeld, waardoor alsnog het remmingseffect optreedt.

Je vraagt je ongetwijfeld af wat de functie is van IC1B en IC1C. Bij beide AND-poorten zijn de ingangen met elkaar verbonden, en de werking is dus identiek aan een buffer. Aangezien een standaard 74LS08 IC 4 AND-poorten bezit, en de schakeling er strikt genomen maar 2 nodig heeft (IC1A en IC1D) blijven er 2 onbenut. Die "gratis" poorten op overschot kunnen dus nuttig gebruikt worden als gate drivers voor de *low side* MOSFET's waarmee de pulldownweerstand worden uitgespaard en de inschakel- en uitschakeltijd wordt bevorderd. Bovendien doen ze dienst als buffer zodanig dat de logische ingangen D0 en D1 van de schakeling hoogimpedant worden, en de Arduino dus niet langer de gestroom van de MOSFET's moet leveren.

### 3.5.2.8 Inschakelvertraging

De werking van MOSFET's steunt op een kunstmatig verarmingsgebied in het drain-sourcekanaal dat via een elektrisch veld veroorzaakt door een spanning op de gate wordt opgebouwd of afgebroken. Dit proces neemt een eindige tijd in beslag. Wanneer de polariteit van de H-brug wordt omgekeerd, ontstaat er dus een kortstondige periode waarin *high side* nog niet volledig spert en *low side* nog niet volledig in geleiding is, of omgekeerd. De impedantie is te hoog om van een echte kortsluiting te spreken, en de tijd te kort om een zekering te activeren. Dergelijke herhaaldelijke piekstroom is wel nefast voor de levensduur van de MOSFET's. Om het H-brugontwerp nog verder te verbeteren moet er dus een oplossing gezocht worden om de geleidende MOSFET de tijd te geven volledig te gaan sperren alvorens de andere in geleiding te brengen.

Als de sturing van de 4 MOSFET's via software wordt gedaan dan is de oplossing triviaal en hoeft er enkel een korte delay te worden toegevoegd. Het probleem kan ook in hardware opgelost worden met een discreet opgebouwde RC-timer zoals weergegeven in figuur 3.22. Het gaat hier meer specifiek over een inschakelvertraging voor de *low side* N-MOSFET van een H-brug. De rest van de H-brug is verbonden met punt A maar weggelaten voor de duidelijkheid. De schakeling maakt weer gebruik van 2 AND-poorten als buffers, omdat de meeste "echte" buffer IC's uit de 74-reeks een open collector



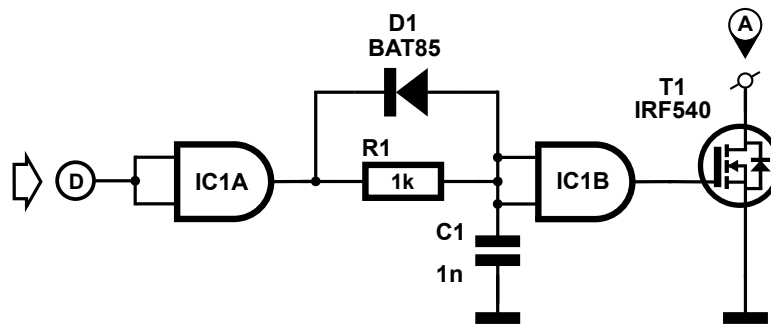
Figuur 3.21: Logische sturing van de H-brug via discrete logica.

uitgang hebben en dus niet bruikbaar zijn.

Bij overgang van een logische “0” naar een logische “1” op de dataingang D moet de MOSFET T1 met vertraging ingeschakeld worden. In de veronderstelling dat C1 leeg is, en de spanning erover dus 0 is, zal de uitgang van IC1B logisch laag zijn en de MOSFET dus sperren. Van zodra de D logisch hoog wordt dan verschijnt op de uitgang van IC1A een spanning van 5 V, en via R1 wordt C1 opgeladen. R1 en C1 vormen dus een RC-netwerkje met een tijdsconstante die het product is van hun waarden. In dit voorbeeld is de tijdsconstante  $\tau = 10^3 \cdot 10^{-9} = 10^{-6}$  s. T1 blijft sperren tot de spanning over C1 de treshold van IC1B bereikt, die rond de 2 V ligt, waarna de MOSFET wordt ingeschakeld. Op basis van de tijdsconstante  $\tau$  kan de inschakelvertraging dus ingesteld worden. De formules vind je in de labnota’s Basiselektronica uit 1<sup>e</sup> Bachelor.

Wanneer ingang D van logische “1” naar “0” gaat moet de MOSFET uitgeschakeld worden zonder vertraging. Daarvoor werd diode D1 toegevoegd. Als C1 is opgeladen tot ca. 5 V en de uitgang van IC1A laag wordt, dan hoeft C1 zich niet meer via R1 te ontladen, met vertraging to gevolg, maar kan zich direct door D1 ontladen. De MOSFET wordt dus quasi direct uitgeschakeld.

Het principe van de ene MOSFET uit te schakelen vooraleer de andere in te schakelen heet *break*

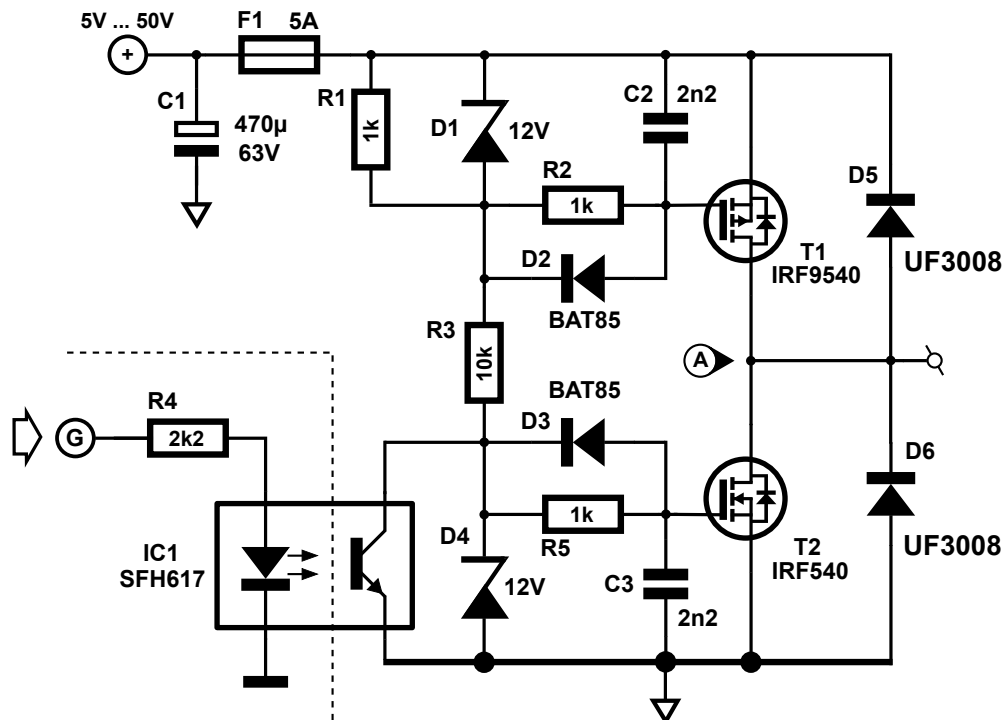


Figuur 3.22: Logische sturing van de H-brug via discrete logica.

before make logic.

### 3.5.2.9 Galvanische isolatie

Als de motors draaien op spanningen die veel hoger zijn dan de logicaspanning (5 V in het geval van Arduino) dan is het een riskante keuze om de H-brug rechtstreeks vanuit de Arduino aan te sturen. Van zodra er iets misloopt bestaat immers de kans dat die hoge spanning op de digitale uitgangspinnen van de Arduino terecht komt, en die overleeft dat waarschijnlijk niet. In dergelijke gevallen wordt gekozen voor een zogenaamde *galvanische scheiding* tussen beide onderdelen van de schakeling. Dat betekent dat er geen rechtstreekse elektrische verbindingen zijn tussen beide onderdelen. Een typische methode hiervoor is om informatie van de ene naar de andere deelschakeling over te brengen via licht, bijvoorbeeld door middel van een *optocoupler*. Een optocoupler bestaat uit een LED en fototransistor in één enkele behuizing. Van zodra de LED wordt aangestuurd dan wordt de fototransistor belicht en zal zijn impedantie aanzienlijk dalen. In onbelichte toestand is de impedantie vrij hoog. Figuur 3.23 toont een mogelijk implementatie van een fototransistor om een halve H-brug (een *half bridge* genoemd) te sturen.



Figuur 3.23: Galvanische isolatie van de stuurlogica en de vermogenselektronica van een half bridge.

Figuur 3.23 toont een *half bridge* die verschillende elementen uit vorige ontwerpen combineert. In dit ontwerp wordt symmetrische sturing toegepast door zowel de *high side* als de *low side* samen aan te sturen. De netwerkjjes rond R2/D2/C2 aan *high side* en D3/R5/C3 aan *low side* zijn de discrete inschakelvertragingen die in de vorige paragraaf uitvoerig besproken werden. D1 en D4 zorgen voor een begrenzing van de gate-sourcespanningen tot max. 12 V, en R3 begrenst de stroom als één van de zenerdiodes in geleiding komt. R1 en de fototransistor zijn complementaire componenten: R1 fungeert als pullup en trekt beide gates passief hoog terwijl de fototransistor dienst doet als actieve pulldown en de beide gates laag trekt. De fototransistor is met een LED geïntegreerd in een optocoupler van type SFH617. R4 begrenst de stroom door de LED van de optocoupler als ingang G met een digitaal signaal wordt gestuurd, bijvoorbeeld 5 V vanuit de Arduino. De stippenlijn geeft aan waar de galvanische scheidingslijn loopt.

C1, F1 en D5/D6 beveiligen de schakeling. C1 is een tankcondensator die stroompieken opvangt en verhindert dat ruis “ontsnapt” uit de H-brugschakeling. F1 is een zekering die dienst doet als failsafe voor de H-brug: als er iets misloopt en de stroom stijgt boven 5 A dan wordt de zekering getriggerd en de stroom afgesneden. D5 en D6 ten slotte zijn de vrijloopdiodes die bij T1 resp. T2 horen.

### 3.5.2.10 Stroombegrenzing

Zolang motors draaien wordt de stroom tegengewerkt door de BEMF die door de roterende magnetische velden zelf wordt gegenereerd. Als de BEMF wegvalt doordat de rotatie stopt, bijvoorbeeld als een DC-motor blokkeert of een stappenmotor wordt stilgezet, dan wordt de stroom enkel nog begrensd door de resistieve component van de totale impedantie. Vermits spoelen slechts uit goed geleidend koperdraad bestaan, kan de stroom dus erg hoog oplopen. Als niet de gepaste maatregelen genomen worden dan zal het ontwikkelde vermogen zodanig hoog oplopen dat zowel de MOSFET's als de motors in kwestie ongezond heet zullen worden. Bij een goed ontwerp zal ook de zekering van de H-brug getriggerd worden. Nochtans zijn er tal van situaties denkbaar waarin DC-motors geblokkeerd geraken, bijvoorbeeld een akkuboormachine waarvan de boor vast zit of een stappenmotor die een constante positie moet behouden om een rechte lijn te kunnen 3D-printen. Dit issue moet dus opgelost worden.

De eenvoudigste manier om de stroom te begrenzen is een constante weerstand in serie te zetten met de motor of de H-brug. Daarmee wordt de impedantie constant verhoogd, waardoor de stroom bij stilstaande motor kan verminderd worden tot elke gewenste waarde. Het nadeel is dat er ook bij rotatie een spanning over die weerstand zal vallen waardoor het uitgangsvermogen van de motor daalt. Het vermogen dat in de serieweerstand verstoekt wordt is ook onnuttig gebruikt en dus verlies.

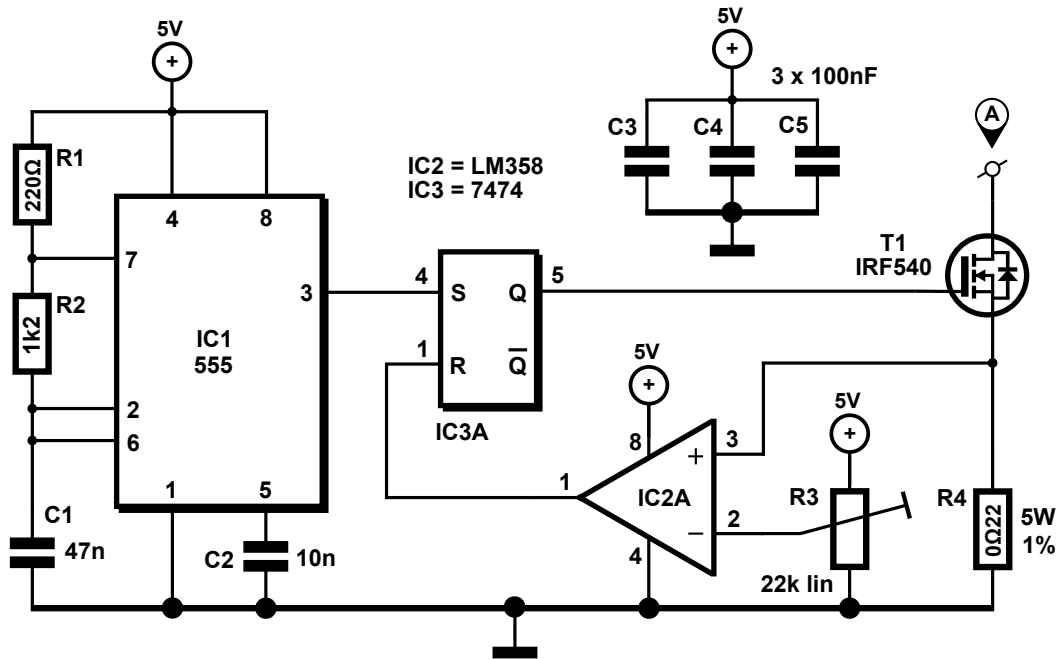
Een elegantere oplossing is te verhinderen dat de stroom te hoog oploopt door gebruik te maken van de zelfinductie van de spoelen in de motor. Vermits een spoel stroomveranderingen tegenwerkt, duurt het een eindige tijd vooraleer de stroom door een spoel na een trapspanning aan te leggen zijn maximale sterkte bereikt. Dat effect kan nuttig gebruikt worden in combinatie met een PWM-generator. Via een PWM-generator kan de spanning snel genoeg aan- en uitgeschakeld worden zodanig dat de stroom door de spoel nooit de kans krijgt om te hoog op te lopen. Cruciaal hierbij is de PWM-frequentie, die in functie van de zelfinductie van de motor moet gekozen worden. De duty cycle van het PWM-signaal bepaalt het gemiddelde vermogen dat de motor bereikt.

Net zoals in de vorige paragrafen kan het probleem zowel opgelost worden in software als in hardware. Via de ingebouwde PWM-generator van de Arduino kan een PWM-signaal gegenereerd worden op de compatibele digitale IO-pinnen via de `analogWrite` functie. Met een duty cycle van 100 % wordt de H-brug gestuurd op vol vermogen, met een lagere duty cycle schiet de PWM in actie. De duty cycle voor een geblokkeerde stepper kan dan bepaald worden aan de hand van de gewenste *holding torque*.

Alternatief kan het PWM-signaal ook met discrete hardware gegenereerd worden. Een 555 IC, waarmee je vorig jaar hebt geëxperimenteerd in de practica Basiselektronica, is daarvoor perfect geschikt als hij in een astabiele multivibratorconfiguratie wordt toegepast. Nog interessanter is om de duty cycle aan



te passen aan de werkelijke belasting, daarvoor is dan weliswaar terugkoppeling (*eng. feedback*) van informatie over de stroom vereist. Figuur 3.24 toont een mogelijke implementatie.



Figuur 3.24: Terugkoppeling van stroominformatie van een shuntweerstand voor de bijregeling van de PWM-duty cycle.

IC1 is een standaard 555 timer-IC geconfigureerd als astabiele multivibrator met constante frequentie, ingestelbaar via C1. Deze blokgolf wordt aan de SET-ingang van een SR-latch aangelegd, die bij elke stijgende klokflank van de 555-oscillator logisch “hoog” wordt gezet. Uitgang Q wordt meteen ook hoog waardoor de MOSFET T1 wordt ingeschakeld en er dus een stroom gaat lopen door de inductieve last. Deze stroom loopt ook door shuntweerstand R4, met een waarde die voldoende klein gekozen wordt zodanig dat er niet teveel verliezen over optreden. De spanning over deze weerstand, die via de Wet van Ohm evenredig is met de stroom erdoor, wordt aangelegd op de niet-inverterende ingang van comparator IC2A. Op de inverterende ingang wordt een referentiespanning aangelegd, instelbaar via trimpot R3. Van zodra de spanning over de weerstand de ingestelde spanning overschrijdt dan zal de uitgang van comparator IC2A “hoog” worden en de SR-latch resetten waardoor de MOSFET weer wordt uitgeschakeld. De stroom door de last neemt dan af via een vrijloopdiode (niet getekend) en blijft uitgeschakeld tot de volgende stijgende klokflank van de 555 waarna de cyclus herbegint. Afhankelijk van de zelfinductie zal de referentiespanning van R3 vroeger of later overschreden worden, en de duty cycle van het signaal dus langer of korter zijn.

Een SR-latch in 74-serie is niet zo gemakkelijk te vinden, maar flipflops zoals de 7474 wel. Die kunnen gebruikt worden als SR-latch door de *data* en *clock* inputs aan grond te leggen, en de *preset* input te gebruiken als SET voor de latch en de *clear* input als RESET, zoals in figuur 3.24 is gedemonstreerd.

Zoals gebruikelijk heeft elk IC zijn eigen ontkoppelcondensator van 100 nF om ruis ten gevolge van digitale transities weg te werken.

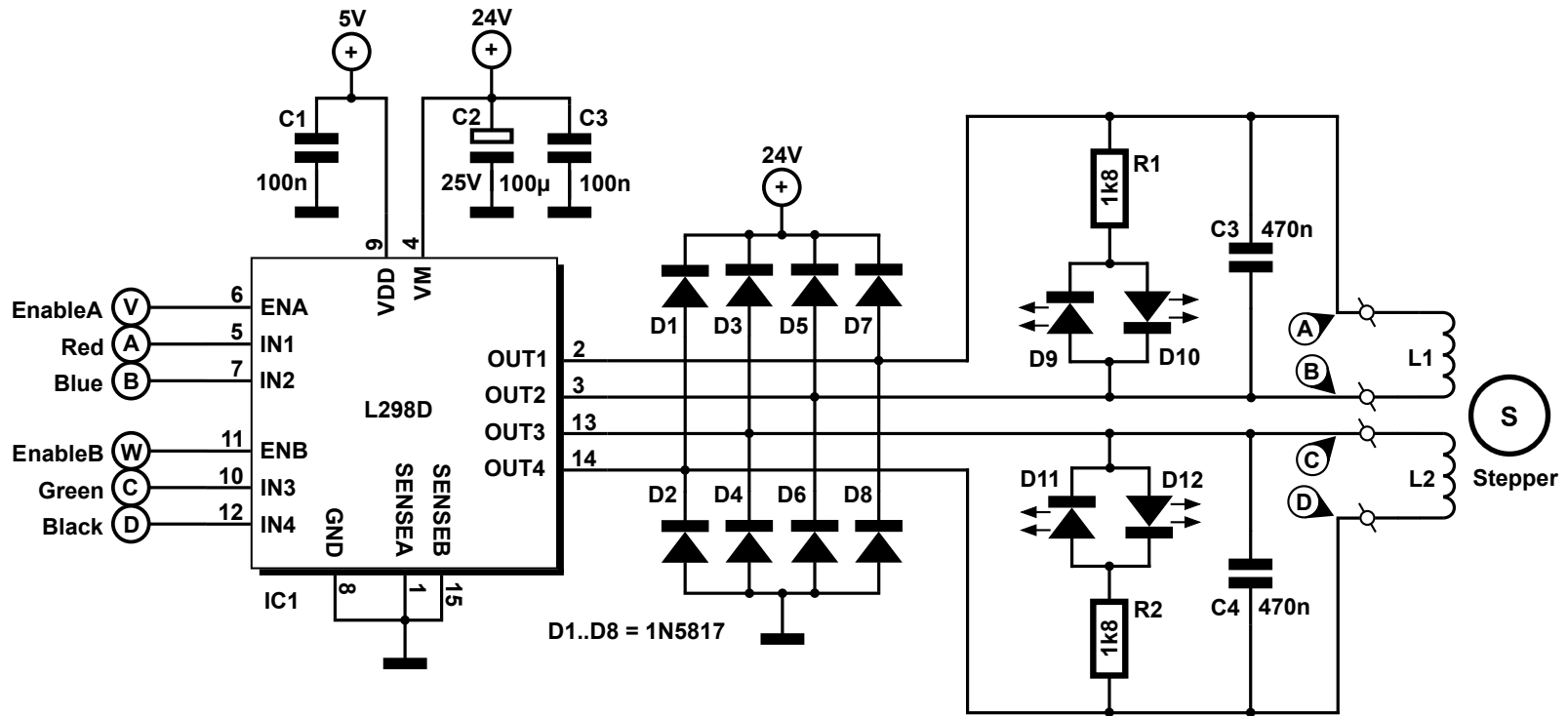
Naast een discrete uitvoering is deze schakeling ook verkrijgbaar als geïntegreerde IC's die als stroomgestuurde PWM-controllers gebruikt worden. Een veel toegepast IC hiervoor is de L6506 van ST Microelectronics dat zowel de 555-oscillator, de SR-latch en comparator bevat.

### 3.5.2.11 Geïntegreerde H-bruggen

Omdat H-bruggen voor zoveel toepassingen gebruikt worden, zijn ze ook goed verkrijgbaar als volledig geïntegreerde IC's waarin soms zelfs stuurlogica en inschakelvertraging verwerkt zit. Dergelijke IC's kunnen met de Arduino digitaal aangestuurd worden, en langs de uitgangszijde kan er rechtstreeks een motor aangesloten worden.

Geïntegreerde H-bruggen hebben ook beperkingen: de maximale stroom is zeer beperkt (500 mA of minder) waardoor ze niet inzetbaar zijn voor de krachtige stappenmotors die je voor het Ontwerpproject gebruikt. Ze kunnen wel een oplossing bieden voor DC-motors, of in andere toepassingen waar zeer kleine stappenmotors moeten gestuurd worden. Een ander probleem is dat veel H-brug-IC's op basis van bipolaire junctietransistors zijn opgebouwd, waardoor ze zeer heet worden en de koeling een omslachtige opgave wordt. Vrijlooptdiodes kunnen al dan niet ingebouwd zitten.

Populaire H-brug-IC's zijn o.a. de L293 en L298 die respectievelijk 250 mA en 750 mA continue belasting aankunnen. Figuur 3.25 toont een ontwerp van een dubbele H-brug die rond een L298 is opgebouwd. Dit IC bevat 2 afzonderlijke identieke H-bruggen zodat met één IC een kleine stappenmotor of 2 DC-motors kunnen aangestuurd worden. IC's die grotere stromen aankunnen zijn ook commercieel beschikbaar maar enkel in surface mount (SMD) package, dat niet te prototypen is op breadboard en speciale PCB-design technieken vereist.



Figuur 3.25: Sturing van een kleine stappenmotor met een geïntegreerd H-brug-IC van type L298. Let op de externe vrijlooptdiodes en buffer- en ont-koppelcondensators op de voeding.

# 4

## Prototyping

“Perfection is not attainable, but if we chase perfection we can catch excellence.”

---

Vince Lombardi

### 4.1 Inleiding

Prototyping is zonder twijfel één van de belangrijkste vaardigheden (misschien wel de allerbelangrijkste) die je als ingenieur moet aanleren. Wie de kunst van het prototypen onder de knie heeft – elektronisch, mechanisch of een combinatie van beiden – kan snel kwaliteitsvolle prototypes creëren. Voor wie deze vaardigheid niet bezit daarentegen, worden de meeste praktische aspecten van engineering een ellendig lange, frustrerende lijdensweg.

Nadat je de datasheets van de componenten waarmee je aan de slag wilt hebt uitgepluisd, kan het echte werk beginnen en start je met het uittekenen van je ontwerp. In deze ontwerpfase (*eng. design phase*) vertaal je ideeën naar elektronica en breng je componenten samen tot schakelingen. **Het ontwerp begint op een blad papier!**

De kans is klein dat je de eerste bent die een bepaalde transistor, MOSFET of IC gebruikt in zijn ontwerp. Eens rond kijken wat er al bestaat kan dan interessant zijn om inspiratie op te doen en te leren uit fouten van anderen. De zoekopdracht `L298 example circuit` via *Google Images*<sup>1</sup> levert een groot aantal voorbeeldschakelingen op. Je merkt bijvoorbeeld dat er aan de uitgangen van de L298 diodes nodig zijn om het IC te beveiligen, mocht je dit gemist hebben in de datasheet. Je ziet ook tekeningen in verschillende EDA-pakketten, voorbeelden van PCB's enz. Een schat aan informatie die je zeker niet mag missen! Let er wel op steeds informatie van meerdere bronnen te vergelijken, want lang niet alle schakelingen en schema's die je op het Internet vindt zijn correct! Bovenal je gezond verstand gebruiken is dus de boodschap. Naast Google Images zijn er nog tal van andere online en offline bronnen die nuttig kunnen zijn. Gespecialiseerde EDA-fora zoals <http://www.edaboard.com> kunnen je helpen met gerichte problemen. Denk er ook aan dat de vakgroep INDI geabonneerd is op het vakblad *Elektor* waar je regelmatig interessante schakelingen mét ondersteunende beschrijving in het Nederlands in kan terugvinden. Je kan *Elektor* inkijken in onze bibliotheek.

De belangrijkste vereiste om een schakeling te kunnen prototypen is **te begrijpen wat je aan het doen bent**, en daarvoor moet je **begrijpen hoe je schakeling precies werkt**, tot in detail. “Ongeveer”

---

<sup>1</sup><https://encrypted.google.com/imghp>

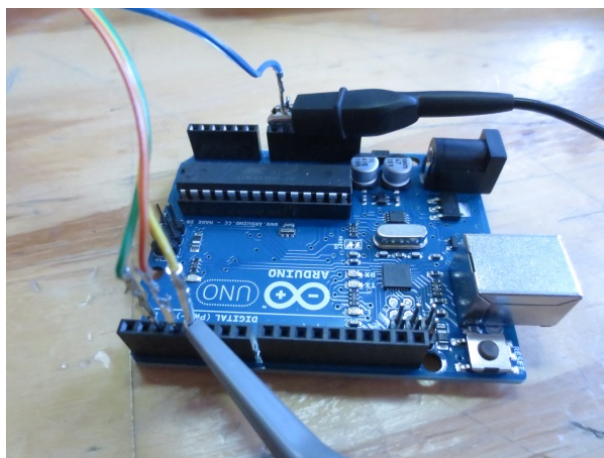
weten hoe het werkt is niet voldoende als de schakeling niet blijkt te werken en je moet metingen op je multimeter interpreteren om de fout op te sporen.

## 4.2 De breadboardimplementatie

Ben je ervan overtuigd dat je een schema hebt dat zou kunnen werken, dan kan je beginnen aan de bouw ervan. Dit doe je altijd eerst op een breadboardje, en **nooit** direct op een PCB!

Breadboards bestaan uit metaalstrips met geïntegreerde klemmen, die per kolom zijn doorverbonden. De meeste breadboards bestaan uit 3 of 4 blokken: 2 signaalblokken die van elkaar gescheiden worden met een diepe geul, en 1 of 2 voedingsrails die gemarkeerd zijn met een blauwe en rode streep voor resp. grond en voeding. Bij sommige breadboards zijn de voedingsrails halverwege onderbroken om je toe te laten met verschillende voedingsspanningen te werken in je schakeling. Dit check je dust best alvorens je je schakeling begint te bouwen! Een breadboard is **enkel** geschikt voor het prototypen van elektronica, m.n. spanningen tot ca. 60 VDC en stromen tot max. 1 A. De behuizing bestaat bovendien uit een thermoplastische kunststof, die zal smelten als de componenten op je breadboard te heet worden! Nog een extra reden dus om de spanning en de stroombeperking van je voeding te controleren alvorens die in te schakelen.

De sockets van een breadboard zijn ongeveer 5 mm diep, afhankelijk van het type, en kunnen pinnen tot 0.9 mm accommoderen. Aan componenten met dikkere leads zul je dus eerst draden moeten solderen of verbinden met een krokodillenklem (*eng. alligator clip*). Als draden te kort zijn, dan zitten ze niet goed vast en dit geeft aanleiding tot valse contacten. Fijnaderige kabels, zoals gebruikt in netsnoer, flexkabel of de kabels van je stappenmotors bijvoorbeeld, kan je niet direct in een breadboard pluggen. Je moet hier eerst stijve pinnen op knijpen met behulp van de krimptang.



*Figuur 4.1: Vieraderige flexkabel aangesloten op een Arduino met behulp van krimppinnen. Dit voorbeeld toont hoe het **niet** moet: 3 blote crimp plugs op een rij die bij de minste schok kortsluiting maken, en de Arduinoheaders die gebruikt worden als wire-to-board connectors in plaats van de board-to-board connectors waarvoor ze bedoeld zijn.*

Een andere situatie waarin breadboards soms moeilijkheden kunnen opleveren, is bij gebruik van componenten met een pitch (= afstand tussen de leads) die niet gelijk is aan (een veelvoud van) 2.54 mm, of exact 100 mil.

Een vereiste om te kunnen prototypen op breadboard is uiteraard dat de component in kwestie leads (draden, 'pootjes') heeft die kunnen ingeplugd worden. Met surface mount componenten (SMD) kan je niet direct prototypen. Je zal er eerst een *breakout board* voor moeten ontwerpen.

Indien mogelijk bouw je je schakeling op het breadboard geografisch gelijklopend aan je schema.

Componenten die bij elkaar horen, zet je dan ook zo dicht mogelijk bij elkaar. Doordat elke kolom 5 terminals heeft, kan je dus zonder bijkomende draden netten met 5 verbindingen creëren. Waar nodig zal je draadbruggen moeten gebruiken om kolommen te verbinden met andere kolommen of met voedingsrails. Maak hiervoor bij voorkeur gebruik van speciale *breadboardkabels* die je kan herkennen aan hun gestandaardiseerde afmetingen, stijve uiteinden en zwarte schoentjes. Je kan ook stukjes montagedraad afknippen, maar die moet je dan manueel strippen. Bovendien is montagedraad minder soepel waardoor je de kans loopt dat de uiteinden afbreken en in de breadboardsockets blijven steken.

Een schakeling bouwen op breadboard vergt **concentratie**. Begin met de voedingsrails aan te brengen, en werk vervolgens net per net af totdat je de volledige schakeling hebt opgebouwd. Bij voorkeur zoek je alvorens te starten met de constructie eerst alle nodige componenten samen op basis van de BOM (*eng. Bill Of Materials*) die je uit Altium Designer of CircuitMaker kan afdrukken. Zo weet je zeker dat je geen componenten vergeten bent, als er na afloop geen componenten meer op je tafel liggen.

Om de schakeling te testen stel je uiteraard eerst de voeding in op de juiste spanning, en de stroombegrenzing op de geschatte stroom van de schakeling. Let er zeker op om de voeding niet per ongeluk om te polen bij het aansluiten op je schakeling! Sluit de stappenmotor nog niet aan, maar maak gebruik van LED's om de belasting te visualiseren (zie *Tips & Tricks* in het volgende hoofdstuk).

Schrijf de nodige code om de H-brug aan te sturen via de digitale pinnen, of maak gebruik van de gepaste Arduino-library om de logica voor het sturen van een stappenmotor te implementeren op je Arduino. Sluit je Arduino aan op de schakeling en schakel de spanning in. Als je de sequentie traag genoeg doorloopt, dan zouden de 4 LED's steeds in dezelfde sequentie moeten oplichten. Als dit correct werkt, dan kan je de stappenmotor aansluiten en het daarmee proberen.

Werkt alles naar behoren? Laat de motor dan een tiental minuten lopen en hou de temperatuur en andere statusparameters van de schakeling nauwlettend in het oog. Als er fouten zitten in de dimensionering van componenten, dan zullen die problemen snel zichtbaar worden.

Na 10 minuten nog alles OK? Dan kan je nog een laatste *stresstest* uitvoeren. Blokkeer de motor om te controleren of je schakeling ook met deze situatie overweg kan.



# 5

## Debuggen

“Believe you can and you’re halfway there.”

---

Theodore Roosevelt

### 5.1 Inleiding

De werking van een elektrische of elektronische schakeling wordt snel complex naarmate componenten worden toegevoegd, omdat elke component een invloed kan hebben op één of meerdere andere componenten in de schakeling. Wanneer de schakeling uiteindelijk niet doet wat je ervan verwachtte, dan kan is de oorzaak vaak niet eenduidig te bepalen. Het probleem zou bijvoorbeeld kunnen liggen aan

- een defecte component;
- een fout in het ontwerp van de schakeling;
- een kortsluiting op de printplaat;
- een slechte soldeerverbinding;
- een haperend elektrisch contact in de bekabeling;
- inputparameters die buiten de berekende grenzen liggen;
- ...

Je merkt dat er nood is aan een gestructureerde manier om fouten op te sporen, zodat het het protoypen van elektronische schakelingen niet uitmondt in een eindeloze lijdensweg.

### 5.2 Basistechnieken

In wat volgt leggen we enkele basistechnieken voor het debuggen van elektronica uit aan de hand van voorbeelden.

Om efficiënt te kunnen debuggen moet je over **het schema van de schakeling** beschikken, en indien van toepassing ook **de layout van de PCB**. Het vergt heel wat ervaring om een complexe schakeling te debuggen zonder schema bij de hand. Bij voorkeur druk je dit schema (of schema's) af op papier zodat je meteen ook aantekeningen kunt maken, spanningen en signaalvormen kunt bijschrijven enz.



Om een bestaande schakeling – meestal op PCB – te kunnen debuggen, moet je deze eerst onderwerpen aan een *reverse engineeringproces* om de werking ervan te achterhalen en de functie van de verschillende componenten te identificeren. Reverse engineering is een ingewikkeld en tijdrovend proces dat veel ervaring vereist, en valt dus ook buiten de scope van deze nota's.

### 5.2.1 Modulair debuggen

Vaak kan een schakeling opgesplitst worden in kleinere subsystemen die onafhankelijk van elkaar één of meerdere functies vervullen. Een mooi voorbeeld is de Arduino zelf: die bevat een voedingsmodule die zorgt voor de energievoorziening op het bordje, er is een microcontrollermodule met bijhorend LED'je en oscillator, en tot slot de programmeermodule met USB-interface die de verbinding met de computer verzorgt. Hoewel alle modules samenwerken om de functionaliteit van de Arduino te verzorgen, kunnen ze afzonderlijk van elkaar gestest worden.

Om een complex systeem te debuggen tracht je dus eerst om de verschillende modules te identificeren en logisch van elkaar te ontkoppelen. Meet de uitgang(en) van de betreffende modules en zoek de module die defect is. Vervolgens gebruik je *tracing* om fouten binnen een module op te sporen.

### 5.2.2 Tracing

Eens je hebt kunnen indentificeren welke module defect is, kan je aan de slag met het zoeken naar het probleem binnen die module. Hiervoor bepaal je enkele testpunten in de schakeling, die indicatief zijn voor de werking van de schakeling. De input(s) en output(s) van de module, maar ook de voedingsrail zijn hoogstwaarschijnlijk interessante testpunten.

Begin met de markering van de testpunten en bijhorende spanningen op je schema('s). In het geval van analoge systemen zoek je de DC-instelpunten. Condensators worden in dat geval open keten, inductors kortsluitingen. Als je de spanning op elk testpunt kent, dan kan je ze gaan nameten in je schakeling.

Controleer eerst de voedingsspanning(en), want zonder voeding kan geen enkele schakeling werken. Maak indien mogelijk gebruik van een labovoeding om de module te testen zodat voedingsproblemen kunnen uitgesloten worden. Vervolgens start je bij de input(s) van de module en verifieer je systematisch de spanning op elk testpunt tot aan de output(s). Meet je een afwijking, dan is de kans groot dat de fout tussen de 2 laatst gemeten testpunten ligt. Nu kan je gaan verfijnen, en het defect identificeren.

Als een groot aantal testpunten een foutieve spanning draagt, dan is de kans groot dat het probleem zich gecentraliseerd voordoet. Zoek in dat geval naar componenten die geografisch centraal gelegen zijn tussen de foutieve testpunten, en met veel andere punten verbonden zijn zoals bijvoorbeeld IC's.

Het is mogelijk dat alle (DC) spanningen op de testpunten kloppen, maar de module toch niet doet wat je ervan verwacht. In dat geval zit er niks anders op dan op AC te gaan tracen. Leg een referentiesignaal aan op de input(s) binnen de verwachte grenzen. Een gemakkelijk startsignaal is bijvoorbeeld een sinusoidaal signaal met een spanning van  $1 V_{pp}$ , een offset van 0 V en een frequentie van 1 kHz. Een signaalgenerator kan je helpen bij het genereren van dit referentiesignaal.

### 5.2.3 Defecten identificeren

Tracing geeft je slechts een indicatie waar het probleem zich situeert, maar duidt zelden aan welke component defect is. Eens het probleem geïsoleerd is tot een subsectie van een module, dan start de laatste stap: de zoektocht naar de defecte component(en).

Alvorens dieper in te gaan op de zoektocht naar defecte componenten, eerst een woordje uitleg over *failure modes* van elektronica. Dit geeft een antwoord op de vraag *waarom gaat elektronica überhaupt vroeg of laat kapot?* Er zijn verschillende factoren die een rol spelen:

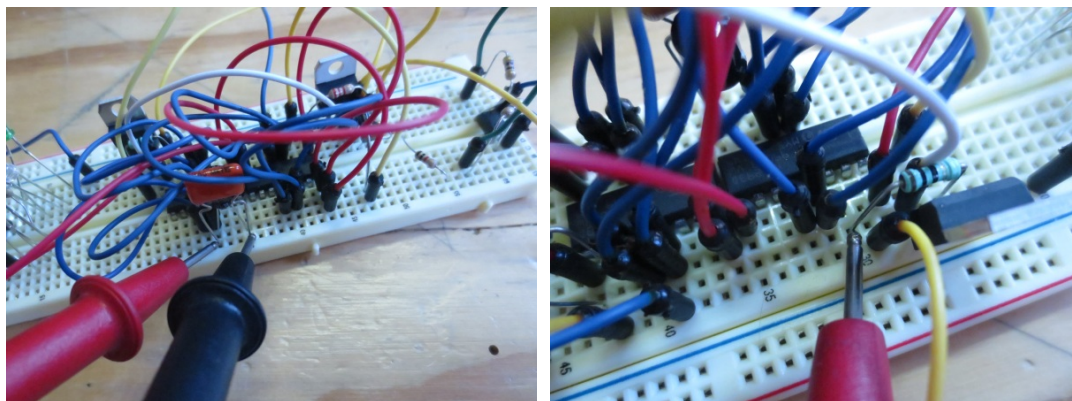
- componenten worden overbelast: ze werken te dicht bij hun *maximum ratings* in hun normale werkmodus;
- componenten worden te heet: het vermogen werd foutief gedimensioneerd, of de koeling werkt niet naar behoren;
- statische ontladingen: de behuizing werd niet geaard, geen afscherming tegen elektromagnetische interferentie (EMI);
- componenten bereiken hun normale levensduur: mechanische onderdelen zoals schakelaars slijten;
- fouten in het ontwerp: de *design guidelines* van de fabrikant werden niet gevolgd;
- ...

Hierbij moet je goed beseffen dat één defecte component de module, en dus misschien ook het volledige schakeling, buiten werking stelt. De andere componenten in de schakeling zijn dus hoogstwaarschijnlijk nog in orde in geval van kleine defecten. Dat is ook de reden waarom we in het fablab en labo Elektronica bakken afgedankte PCB's houden: hoewel de meesten uit defecte toestellen afkomstig zijn, zijn het overgrote deel van de componenten op de PCB's nog in uitstekende staat!

Soms heb je geluk, en is het defect visueel waar te nemen. Bijvoorbeeld:

- componenten zijn verkleurd (tot zwartgeblakerd) door oververhitting, zie volgende paragraaf;
- delen van een component ontbreken na een explosie;
- een elektrolytische condensator staat bol, is opengebarsten of heeft elektrolyt gelekt;
- een glaszekering is doorgebrand;
- een component wordt abnormaal heet en begint te smeulen na korte tijd onder spanning te staan;
- ...

In deze gevallen is de boosdoener snel gevonden.



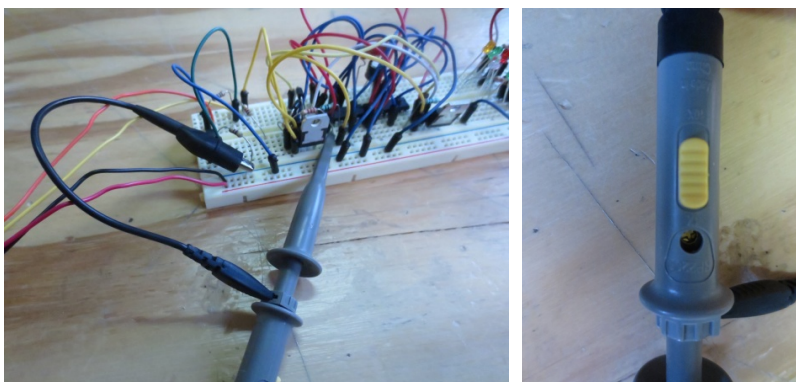
*Figuur 5.1: 2 technieken om defecten snel te identificeren: spanningen over componenten meten (links) en de spanning op een punt met als referentie de grond meten (rechts).*

Zijn alle componenten even vatbaar voor defecten? Gelukkig niet, en dat maakt het identificeren van defecten een stuk eenvoudiger. In de regel zijn actieve componenten gevoeliger voor defecten dan passieve componenten, en zijn halgeleiders gevoeliger dan al de rest. Kijk componenten dus na van gevoelig naar ongevoelig, te starten met de meest gevoelige component (IC's).

Componenten nameten kan je enkel als ze verwijderd zijn uit een schakeling. Bij breadboard-schakelingen is dit gemakkelijk, bij PCB's een pak ingewikkelder (zeker als de component meer dan 2 pootjes heeft!). Dit is één van de voornaamste redenen waarom je **altijd** je schakeling eerst op breadboard moet testen alvorens ze op een PCB te implementeren! Componenten kan je opmeten als volgt:

- weerstand: ohmmeter
- condensator: RLC-meter
- inductor/spoel/transformator: ohmmeter en RLC-meter
- diode: diodemeter
- transistor: diodemeter
- MOSFET: referentieschakeling
- schakelaar: ohmmeter
- IC: referentieschakeling

Sommige componenten, zoals IC's en MOSFET's, zijn erg moeilijk na te meten via conventionele technieken. De eenvoudigste manier om hun werking te verifiëren is hen in een referentieschakeling te steken – dit is een schakeling waarvan je het gedrag precies kent, en die wel werkt met een andere exemplaar van de component die je wil nameten.

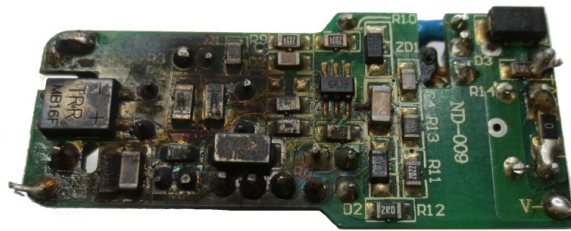


*Figuur 5.2: Meten van AC signalen in een schakeling met een oscilloscoop. De referentie van de probe wordt verbonden met de grond (zwarte klem) en het meetpunt met de schakeling (grijze klem). Beiden worden bij voorkeur zo dicht mogelijk bij elkaar aangehecht (foto links). Probes kunnen met een interne spanningsdeler gecofigureerd worden ('10x') dus zorg dat de configuratie op '1x' staat om de amplitude correcte te registeren! (foto rechts)*

Als je een defecte component geïdentificeerd hebt, vervang hem dan in je schakeling door een ander exemplaar van *exact* hetzelfde type en test je schakeling opnieuw. Werkt deze nog steeds niet, controleer dan eerst of de component die je net vervangen hebt niet weer defect is! Is dit het geval, dan is de kans groot dat er een ontwerpfout in je schakeling zit. Is de gewisselde component wel OK, doorloop dan opnieuw de stappen om het volgende defect op te sporen.

#### 5.2.4 Catastrophic failures

In schakelingen die onderhevig zijn aan een grote stroom, spanning of flux zullen defecten ook een navenant effect hebben. Dit resulteert vaak in overtuigende audiovisuele signalen zoals rook, sissende componenten, explosies enz. Men spreekt in dat geval over *catastrophic failures*. Bovendien duurt het meestal niet lang na inschakelen van de spanning vooraleer deze effecten optreden.



Figuur 5.3: AC/DC converter op 230 VAC met zwartgeblakerde componenten in het 230 V gedeelte van de schakeling, getuige van een catastrophic failure.

In deze gevallen blijft het defect zelden beperkt tot één enkele component en is het wellicht efficiënter om het (vermogens)gedeelte van de schakeling volledig vanaf 0 te herbouwen. De schakeling of PCB is in dat geval FUBAR<sup>1</sup> (eng. *Fucked Up Beyond All Repair*).

### 5.3 PCB's debuggen

Als een schakeling werkt op een breadboard dan is de kans groot dat ze ook zal werken op een PCB. Is dit niet het geval, dan ligt het probleem bij de PCB zelf: onderbroken tracks, kortsluitingen tussen tracks, grond die niet doorverbonden is, slechte soldeerverbindingen enz.

De meeste PCB-gerelateerde problemen kan je vermijden door je pas gefreesde of geëtste PCB op de *lichttafel* te leggen en deze nauwgezet te inspecteren. Een geoefend elektronicus kan dit met het blote oog, beginners kunnen misschien baat hebben bij een vergrootglas.

Merk je kortsluitingen op, dan kan je deze corrigeren met een scalpel (eng. *lancet*) of een scherp mes door de koper te verwijderen. Dit lijkt een amateuristische methode, maar ze wordt ook toegepast bij commerciële (industriële) PCB's! Als de correctie succesvol is, is deze techniek in ieder geval een stuk sneller dan een volledig nieuwe PCB te maken. Verifieer na de correctie met een ohmmeter dat de foutieve verbinding werd verbroken. Foutief onderbroken tracks zijn een stuk moeilijker te corrigeren, afhankelijk van de spanwijdte van het defect. Een druppel soldeertin al dan niet in combinatie met een fijn stukje blank montagedraad kan hier uitkomst bieden. Ook hier controleer je de effectiviteit van je correctie met een ohmmeter.

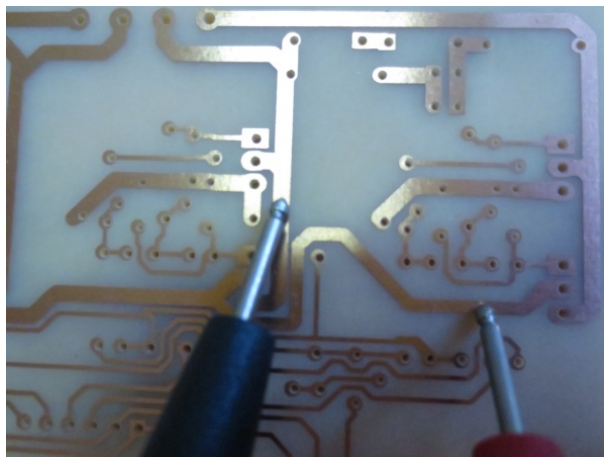
Als alle tracks OK zijn, dan kan je beginnen met het bestukken en solderen van je PCB. **Soldeer je PCB module per module** en meet elke module individueel na met behulp van een voeding en/of signaalgenerator. Uiteraard begin je bij voorkeur met de voedingsmodule(s), indien van toepassing.

Treden er problemen op, dan moet de fout noodgedwongen bij de soldeerverbindingen liggen vermits je schakeling op breadboard werkt en de tracks op de PCB eveneens in orde zijn. Controleer met een ohmmeter de impedantie van de soldeerverbindingen door tussen de leads van de componenten te meten. Een *netlist* of plattegrond van de PCB is hierbij een handig hulpmiddel. Slechte verbindingen kan je corrigeren door de verbinding terug te smelten, een klein beetje soldeertin toe te voegen, en terug te laten stollen. Overmatig soldeertin kan je verwijderen met de desoldeer pomp (eng. *desoldering pump*), of met behulp van *litze* (eng. *desoldering braid*).

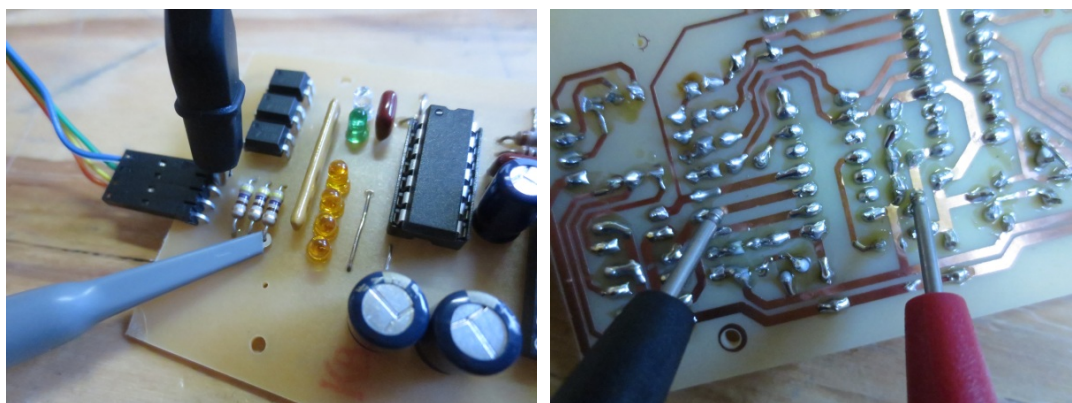
### 5.4 Tips & Tricks

Elektronische schakelingen zijn onderhevig aan het onzekerheidsprincipe: telkens je probeert om een spanning of stroom op te meten, loop je het risico per ongeluk kortsluiting te maken, verbindingen fout

<sup>1</sup>“fubar, adj.”. Oxford English Dictionary (third ed.). Oxford: Oxford University Press. 2005



*Figuur 5.4: Controle van 2 naburige tracks op kortsluiting met behulp van een Ohmmeter. Let erop dat je steeds eerst de fotoresist verwijdert van geëtste PCB's, vermits je anders geen elektrisch contact kan maken met de tracks!*



*Figuur 5.5: Uitmeten van een PCB met behulp van een oscilloscoop (links) en controle van soldeerverbindingen aan de onderzijde van de PCB met een Ohmmeter (rechts).*

terug te steken of draden los te trekken. Dergelijke situaties zijn uiteraard zenuwtergend. In wat volgt geven we enkele tips die je zeker van pas zullen komen.

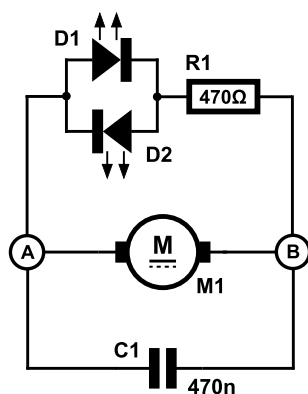
#### 5.4.1 LED's

LED's zijn goedkoop en redelijk ongevoelig voor storingen. Ze zijn dus uitermate geschikt om de status van (delen van) je schakeling te monitoren! Met een LED op de voedingsrails van je schakeling zie je bijvoorbeeld in één oogopslag of er al dan niet spanning aanwezig is op je breadboard. Je kan LED's in principe overal toepassen waar je een digitaal signaal (spanning of geen spanning) wilt visualiseren, op voorwaarde dat je enkele mA op overschot hebt voor de LED. In het begin kan het dus de moeite zijn om LED's aan te sluiten op de digitale uitgangen van je schakeling, van je Arduino enzovoort.

LED's kunnen ook dienst doen als substitutie voor andere types belastingen zoals relais, buzzers of motors. We raden je ten zeerste aan om je H-bruggen bijvoorbeeld eerste te testen met LED's en pas daarna de respectievelijke DC of stappenmotor aan te koppelen. Motor en LED's kunnen ook perfect in parallel blijven zitten, zelfs op de uiteindelijke PCB!

#### 5.4.2 Testpunten

Hoewel LED's erg handig kunnen zijn, hebben ze ook enkele nadelen. Om te beginnen nemen ze behoorlijk wat plaats in op je PCB omdat ze nog een voorschakelweerstand nodig hebben om de stroom



Figuur 5.6: Gebruik van LED's om de draairichting van een DC-motor te visualiseren.

te begrenzen, en als je er teveel hebt dan gaat je PCB lijken op een lunapark. Bovendien verandert de lichtintensiteit van LED's zeer weinig bij veranderende stroom, en kan je ze dus alleen gebruiken om digitale signalen te monitoren.

Om analoge signalen, zoals geluid of video te meten, zal je je oscilloscoop moeten aansluiten op de schakeling. Hiervoor voorzie je dus best al bij het ontwerp van de PCB aansluitpunten. Deze punten heten testpunten (*eng. test points*) en bestaan uit een cirkelvormig koperen pad van 2 tot 4 mm diameter waartegen je een oscilloscoopprobe kan drukken. Indien gewenst kan je ook een gat voorzien in het midden om de probe gedurende langere tijd achter te haken.

De Altium Designer library `std-th.PcbLib` bevat reeds een component `MECH-TEST` die je rechtstreeks op de PCB kan plaatsen en aan een net naar keuze kan hechten.

### 5.4.3 Rubber Duck Debugging

Soms verzeil je in een situatie waarin je alles al hebt uitgetoet, gecontroleerd en doorgemeten, en de schakeling wilt nog steeds niet werken. In dat geval bestaat de kans dat je ergens iets over het hoofd ziet doordat je al (te) lang aan dezelfde schakeling aan het werken bent.

*Rubber duck debugging* kan dan de oplossing bieden. Deze debuggingtechniek bestaat erin je ontwerp of schakeling uit te leggen aan iemand anders vanaf het begin. Deze persoon kan iemand van je groep zijn, die misschien sneller de fout zal zien, maar dit hoeft niet. Rubber duck debugging werkt ook met willekeurige fablabpassanten, en zelfs objecten zoals een rubberen badeendje. Vandaar ook de naam.

### 5.4.4 Thermisch debuggen

Of het nu over transistors, condensators of andere types componenten gaat maakt niet zoveel uit bij het zoeken naar fouten, want ze hebben allemaal één eigenschap gemeen: als er wat fout loopt, dan resulteert dat meestal in een abnormale warmteontwikkeling. Door te checken welke componenten (te) warm worden, kan je dus een snelle indicatie krijgen van de locatie van de fout.

De gemakkelijkste manier om dit te doen is gewoon je vinger op de verdachte componenten te leggen. Je moet dan wel alle componenten sequentieel aflopen. Je kan ook gebruik maken van een infraroodcamera om de temperatuur van componenten te bekijken. Omgekeerd kan je ook je volledige schakeling snel bevriezen met de cryogene spray, en dan kijken welke componenten het snelst ontdooien.

Let op: in sommige situaties is het normaal dat componenten warm worden als deel van de normale werking van de schakeling. De L298's die je als H-brug gebruikt in het Ontwerpproject kunnen bijvoor-

beeld behoorlijk warm worden en dit is volstrekt normaal. Anderzijds, als de ATmega328 op je Arduino even warm wordt, dan loert er wellicht een probleem om de hoek.

#### 5.4.5 Weerstanden en jumpers

Het is bijzonder moeilijk om diodes, transistors, LED's enz. uit te meten zolang ze verbonden zijn met de rest van de schakeling doordat serie- en parallelschakelingen de eigenschappen van deze componenten beïnvloeden. Op een PCB betekent dit dat je in geval van problemen componenten zal moeten lossolderen (met alle nadelige effecten ten gevolge van oververhitting van dien) of dat je tracks tijdelijk zal moeten doorkrassen.

Je kan het jezelf ook veel gemakkelijker maken als je al bij het ontwerp van je schakeling en van de PCB anticipeert op debugging. Dit doe je door op strategische punten een header met 2 pinnen in een verbinding te plaatsen. Zet je over deze pinnen een *jumper* dan worden ze verbonden, zo niet dan is het een open keten. Jumpers nemen weinig plaats in, zijn goedkoper dan schakelaars, en laten je dus toe om je PCB gemakkelijk op te delen in modulaire stukken. Op die manier kan je bijvoorbeeld bepaalde stukken van je schakeling gemakkelijk loskoppelen van een voedingsrail, of van andere delen van de schakeling.

In het geval van digitale communicatielijnen, tussen microcontrollers, bussen of andere digitale componenten, kan je ook weerstanden benutten. Als er dan problemen optreden (bijvoorbeeld het net wordt door 2 componenten gelijktijdig naar '1' en naar '0' getrokken) dan zal de spanning over de weerstand vallen en verhinderen dat de stroom zo hoog wordt dat er schade optreedt. Deze techniek wordt veel toegepast in commerciële bussystemen, in de vorm van pullup en pulldown weerstanden. Weerstanden hebben bovendien nog het extra voordeel dat je er op de PCB tacks kan onderdoor trekken, wat het routeren vergemakkelijkt.