

Schakelmodule Digitale Technologie - online materiaal

KLAAS BEUTE AND RUDY JONKER, AND BART KAPPÉ, AND AERNOUT VAN ROSSUM, AND JAN JA

December 21, 2024

Contents

0.1	Over NLT	1
0.1.1	Woord vooraf	1
0.1.2	Wat is NLT	1
0.1.3	Wat doet Vereniging NLT	1
0.2	Digitale technologie is tegenwoordig overal	2
0.2.1	Leerdoelen	2
0.2.2	De tandarts	2
0.2.3	Beveiliging en privacy	3
0.2.4	Landbouw	4
0.2.5	Opsporing	5
0.2.6	De huisarts	5
0.2.7	Scrum toepassen	7
0.3	Introductie op de module Digitale Technologie	8
0.3.1	Leerdoelen	8
0.3.2	Basisbegrippen	9
0.3.3	Algemene beschrijving van het schema	10
0.3.4	1. Waarnemen (hoofdstuk 3)	10
0.3.5	Vragen	11
0.3.6	2. Verwerken en beslissen (hoofdstuk 4)	11
0.3.7	Algoritme	12
0.3.8	3. Datatransport en dataopslag (hoofdstuk 5)	12
0.3.9	4. Uitvoeren (hoofdstuk 6)	13
0.3.10	5. Feedback (geen eigen hoofdstuk)	13
0.3.11	6. Verbindingen (hoofdstuk 5)	14
0.3.12	Voorbeeldschema: foto's met je mobiel.	15
0.3.13	Wat ga je doen?	15
0.3.14	Nabespreking koortsthermometer	16
0.3.15	Basis van de hardware ontdekken met Arduino, Micro:bit of Raspberry pi	16
0.3.16	Microcomputer en microcontroller	17
0.3.17	Programmeren	17
0.3.18	Waarom zijn ze handig?	17
0.3.19	Leren	18
0.3.20	Beroepen	18
0.4	Keuzeopdrachten	19
0.4.1	Titel opdracht	19
0.4.2	Leerdoelen	19
0.4.3	De omgeving waarnemen	20
0.4.4	Temperatuursensor	20
0.4.5	Analoog-digitaal omzetting van signaal uit sensor	21
0.4.6	Kalibreren en iijken	22
0.4.7	Beschrijving van diverse sensoren in digitale platforms	22
0.4.8	Betekenis van sensoren in een context: het sporthorloge	22
0.4.9	Onderzoeken en maken	24
0.5	Verwerken en beslissen	26

0.5.1	Leerdoelen	26
0.5.2	Vaardigheden	26
0.5.3	Onderdeeltjes op nanoschaal	27
0.5.4	Bouwsteentjes op de chip	30
0.5.5	Programmeren van de microcontroller	33
0.5.6	Kunstmatige Intelligentie: heel geschikt voor het leren herkennen van	33
0.5.7	Kunstmatige Intelligentie om tumoren te herkennen: het onderzoek van Daan Geijs	35
0.5.8	Stap 1: Kiezen van de hardware	36
0.5.9	De Arduino Uno	36
0.5.10	Stap 2: Opzoeken van de juiste drivers	37
0.5.11	Stap 3: Aan de slag, eventueel zónder te bouwen	37
0.5.12	Stap 4: Kalibreren, testen en verbeteren	37
0.6	Data opslag en transport	39
0.7	Leerdoelen	39
0.8	Vaardigheden	39
0.8.1	5.1 De ‘cloud’	39
0.9	Digitale signalen	39
0.9.1	0.9.1 Vraag “{exercise}	40
0.9.2	0.9.2 Verbindingen	41
0.10	Metaaldraad	41
0.11	Elektriciteit	42
0.11.1	0.11.1 Opdracht	43
0.11.2	0.11.2 Datacenters en ‘Cloud’	43
0.11.3	0.11.3 Opdracht Zoek uit wat RAID is en hoe het werkt. Welke vormen van gegevensbescherming kom je nog meer tegen?	43
0.11.4	0.11.4 Beveiliging	44
0.11.5	0.11.5 Vraag “{exercise}	44
0.12	Eindopdracht: Digitaal device ontwerpen	50
0.13	Leerdoelen	50
0.14	Projecten	50
0.14.1	0.14.1 7.1 Digitale technologie in huis: weerstation voor het binnenklimaat.	50
0.15	Opdracht:	50
0.16	Aanwijzingen	50
0.17	1. Vochtmeting	50
0.18	2. Fijnstofmeting	50
0.18.1	0.18.1 7.2 Wetropolis: sensoren en actuatoren voor waterbeweging	51
0.18.2	0.18.2 7.3 Digitale devices die een oplossing zijn bij opwarming in de stad	51
0.19	Stappenplan	52
0.19.1	0.19.1 7.4 De Q-strip: meten van transpiratievocht	52
0.19.2	0.19.2 7.4.1 Introductie	52
0.20	Meten aan zweten	52
0.21	Onderzoek doen	53
0.22	Zelf maken van een Q-strip sensor	55
0.22.1	0.22.1 7.4.2 Het Q-strip-meetsysteem bouwen	58
0.23	Benodigdheden testdevice	58
0.24	LoRaWAN gekoppeld meetdevice	58
0.25	Methode	58
0.25.1	0.25.1 7.4.3 Onderzoek en toepassing	59
0.26	E Verdiepingsopdracht	59
0.27	G Verdiepingsopdracht: Meten met de Q-strip	59
0.27.1	0.27.1 7.5 Andere thema’s voor de eindopdracht	60

0.28 8 Verder met digitale technologie	60
0.29 Fijnstof	60
0.30 Modelleren	62
0.31 Bio-informatica	62
0.32 De toekomst van de landbouw	64
0.33 Cybersecurity	64
0.34 Artificiële Intelligentie (AI)	66
0.35 Leeg analyseschema	67
0.36 Bijlagen verkrijgbaar in de moduledatabase	67
0.37 Appendix	68
0.37.1 Inleiding	69
0.37.2 Ontwerpen en ontwerpcyclus	71
0.37.3 Hardware voor de verwerking van informatie in een digitaal systeem	73
0.37.4 Hardware voor de input van een digitaal systeem	75
0.37.5 Hardware voor de output van een digitaal systeem	90
0.37.6 Hardware voor de opslag en het transport van data	100
0.37.7 Voeding voor een digitaal systeem	104
0.37.8 Testfile	109
0.37.9 Wall of Fame	110

0.1 Over NLT

0.1.1 Woord vooraf

Alsjeblieft, de schakelmodule Digitale Technologie.

Ook zonder dat je lid bent van de Vereniging NLT mag je de leerlingversie vrijuit gebruiken, maar we willen wel graag toelichten wat nIt is en wat Vereniging NLT doet.

0.1.2 Wat is NLT

NLT staat voor Natuur, Leven en Technologie en is een keuzevak voor de bovenbouw havo en vwo. Er is steeds vaker een combinatie van verschillende disciplines nodig om complexe Vraag stukken in de wereld op te lossen. Het vak NLT laat zien op welke manier de vakken aardrijkskunde, biologie, informatica, natuurkunde, scheikunde en wiskunde samenkommen om aan deze complexe opdrachten te werken. Het vak is bedoeld als voorbereiding op de keuze voor een studie op het gebied van bèta en technologie.

Het vak NLT is gebaseerd op talrijke modules over zeer verschillende onderwerpen. Het is aan de school om hier keuzes in te maken, waarbij de enige voorwaarde is dat de eindtermen uit het examenprogramma aan bod komen. De modules worden ontwikkeld door docenten in samenwerking met het hoger onderwijs. Daardoor zijn de modules optimaal afgestemd op ontwikkelingen in maatschappij, wetenschap en technologie.

0.1.3 Wat doet Vereniging NLT

Scholen die het vak NLT aanbieden zijn lid van Vereniging NLT. De Vereniging telt ruim 220 leden. Zij zorgt voor de ontwikkeling van nieuwe modules over actuele onderwerpen. Deze schakelmodule over duurzame ontwikkeling is daar een mooi voorbeeld van.

Ook zorgt de Vereniging voor het onderhoud van bestaande modules door ze elke vijf jaar aan te passen aan de nieuwste ontwikkelingen. Kijk voor het huidige moduleaanbod op de website.

Naast modules werkt de Vereniging aan het professionaliseren van docenten en TOA's door middel van conferenties en bijeenkomsten op landelijk en regionaal niveau. Ook nieuwe ontwikkelingen worden opgepakt. Zo werkt de Vereniging actief mee aan het actualiseren van het examenprogramma en het opzetten van NLT in de onderbouw.

Kortom, NLT is een dynamisch vak dat past in het onderwijs van nu en in de toekomst. Kijk voor meer informatie over NLT op www.verenigingNLT.nl. Hier vind je naast een overzicht van alle beschikbare modules, ook informatie over bijv. de activiteiten van de Vereniging en de regionale vaksteunpunten, leerlijnen, toetsing en over het invoeren van NLT.

Deze schakelmodule is ontwikkeld door een auteursteam van het vaksteunpunt NLT - Pre-U, Universiteit Twente. De module vormt de start van een leerlijn digitale technologie, ontwikkeld door Vereniging NLT. In het laatste hoofdstuk staan korte beschrijvingen van modules die in deze leerlijn passen. Vereniging NLT en het auteursteam wensen je veel leerplezier bij het gebruik van deze module.

0.2 Digitale technologie is tegenwoordig overal

0.2.1 Leerdoelen

Kennis

1. Bewust worden van de betekenis van digitale technologie in je alledaagse omgeving
2. Herkennen van digitale technologie in een context
3. Oriënteren op digitale technologie in je omgeving, waardoor je een goed beeld krijgt van beroepen en gebruik.

Vaardigheden

4. Overzicht krijgen van de vaardigheden die nodig zijn, en welke je al hebt, voor het gebruik van digitale technologie.

Digitale technologie is zó gewoon en onmisbaar geworden voor ons dagelijks leven, dat we er amper meer bij stil staan. Computers, navigatie, mobieljes, online video, streaming, appen, bellen, betalen met de pin: we kunnen niet meer zonder. Het is moeilijk om je een wereld voor te stellen zonder die technologie. Dat betekent ook: in welke richting je ook gaat studeren en werken, je ontkomt niet meer aan digitale technologie. Daarom is het goed om een idee te krijgen wat het is, hoe het werkt, blijft werken en beveiligd wordt, en welke (vaak onzichtbare) invloed deze technologie op ons leven heeft. Om je op weg te helpen hoe digitale technologie gebruikt wordt, beginnen we met een paar voorbeelden.

0.2.2 De tandarts

In de afgelopen tijd ben je vast een keer bij de tandarts geweest. Heb je toen gemerkt hoe de tandarts digitale technologie gebruikt? Waarschijnlijk niet zo bewust. De afspraak is in een digitale agenda vastgelegd. Mogelijk heb je een sms of mailtje ontvangen dat je de afspraak niet moet vergeten. De tandartsstoel wordt met knopjes keurig ingesteld en de tandarts heeft jouw gegevens bij de hand. Uiteraard op een scherm waarin precies te zien is wat er met jouw gebit gebeurd is in de afgelopen jaren, inclusief de röntgenfoto's. Nieuwe foto nodig? Jij klemt een plaatje met een draad tussen je tanden en krijgt een apparaat naast je wang. De tandarts stapt even weg, en bekijkt enkele seconden later de foto op het scherm en kan direct bepalen of er nog iets aan je gebit gebeuren moet. Opzienbarend? Misschien niet, maar wel een groot verschil met dertig jaar geleden. Een afspraak had je op een kaartje, dat je thuis moest bewaren. Met wat geluk werd je (op het vaste thuisnummer) gebeld als herinnering om te komen, maar dat was voor de uitzonderingen. De tandarts had een papieren kaart met daarop een schema van jouw gebit. Tijdens het onderzoek kreeg (net als nu) de assistent(e) gecodeerd informatie waar eventueel een gaatje zat. Dat werd op de kaart aangetekend, en bijgewerkt als het gerepareerd werd. Een foto werd gemaakt met een ingepakt lichtgevoelig negatief. Als je de volgende keer terugkwam was deze foto ontwikkeld en vastgeniet aan je kaart. In elk geval: het kostte allemaal een stuk meer tijd.



Figuur 1.1 Digitale foto bij de tandarts

[link](https://www.google.com/url?sa=i&url=https%3A%2F%2Fdepositphotos.com%2Fstock-photos%2Fdentalxray.html&psig=AOVVawOII8qXaRphhAsq_hQ|4E2r&ust=1623509026466000&source=images-CAOOjhxqFwoTCIC__tHoj_ECFQAAAAAdAAAAABBD)

Om een beugel te zetten, was het nodig dat er een gipsafdruk van je kaak gemaakt werd. De orthodontist moest met die afdruk bepalen waar blokjes en draadjes moesten komen. Ook heel tijdver gevend. De huidige orthodontist gebruikt steeds vaker digitale 3D gebitsfoto's. Met de computer wordt berekend hoe de tanden moeten verschuiven en hoe de beugel moet worden. Dat wordt uiteraard alvast in een mooi plaatje gezet, zodat je kunt zien hoe het gaat worden. De computer berekent dan precies welke onderdelen voor de beugel nodig zijn, zodat die precies op maat zijn en klaarliggen als de beugel geplaatst gaat worden. Daar komt steeds minder handwerk of berekening aan te pas.

De gegevens die de tandarts verzamelt over jou, en van al zijn/haar andere patiënten, kunnen helpen om de mondzorg te verbeteren. Er zijn duizenden tandartsen die dit doen. Al die gegevens (data) vormen een rijke bron van informatie. Als die gegevens via de computer zijn opgeslagen, kun je ze bewerken en doorsturen.

Wie die tot zijn beschikking heeft, kan bijvoorbeeld onderzoeken in welke regio's van het land veel gebitsproblemen zijn, welke behandeling tandartsen inzetten of hoe vaak patiënten de tandarts bezoeken. Je kunt zelf vast nog meer onderzoeksvragen bedenken.

0.2.3 Beveiliging en privacy

Zodra meer mensen computerdata kunnen bekijken is privacy belangrijk. Je wilt liever niet dat jouw persoonlijke gegevens (bijvoorbeeld over je gebit) zomaar door anderen bekeken kunnen worden.

Het verzamelen en gebruiken van digitale gegevens in de zorg luistert nauw. Wie mag ze inzien, waar is toestemming van de patiënt nodig?

Inloggen, veilige verbindingen, onleesbaar maken van gegevens als ze zonder sleutel bekeken worden (encryptie) en bijhouden wie er in de data kijkt: er is veel nodig om privacy te garanderen.

0.2.4 Landbouw



Figuur 1.2 Drone voor het meten aan landbouwgewassen <https://Ggriculturepost.com/7-benefit-of-remote-sensing-gis-in-agriculture/>

Op een gemengd boerenbedrijf worden graan, maïs en bieten geteeld en is een stal met 100 koeien. In het voorjaar bewerkt de boer de akker en zaait de gewassen in. Er wordt waar nodig gewasbescherming gespoten en beregend en gekeken of het gewas zich goed ontwikkelt. Allemaal met het oog. De koeien worden met de melkmachine gemolken, tweemaal per dag. In de zomerperiode zijn de koeien in de wei en worden ze daar gemolken. De melk gaat in melkbussen op de kar mee naar de boerderij. Technologie, en nu ook digitale technologie, heeft dit veranderd en die ontwikkeling gaat nog wel een poosje door.



Figuur 1.3 Een melkrobot <https://www.boerenbusiness.nl/agribusiness/artikel/10878173//lely-astronaut-a5-goedkoper-en-eenvoudiger>

De koeien worden gemolken met de melkrobot, krijgen hun rantsoen voer doordat ze herkend worden aan een zender om hun nek. De boer bepaalt welk deel van de akker bemest moet worden door satellietgegevens en dronefoto's te gebruiken. Als er geoogst wordt, houdt de oogstmachine precies bij hoeveel elk stukje van de akker opgebracht heeft. Hiermee kan bepaald worden wat er aan de grond moet gebeuren voor het volgende seizoen. De komende jaren zal de landbouw nog meer veranderen, mede door klimaatverandering en eisen rond duurzaamheid en milieu.

0.2.5 Opsporing



Figuur 1.4 Forensisch onderzoek aan digitale opslagmedia <https://blog.veriato.com/the-evolution-of-digital-forensics>

De politie maakt op uitgebreide schaal gebruik van digitale technologie. Vingerafdrukken of DNA-sporen van een plaats-delict komen in digitale databases. Gegevens uit telefoons of computers van slachtoffers en verdachten leveren belangrijk bewijsmateriaal. Slimme camera's detecteren hardrijders, mensen in de stad of kentekens van gezochte auto's. Een nieuw onderzoekproject gaat van start om deep-fake videobeelden te kunnen opsporen (dit zijn bedrieglijk lijkende gemonteerde beelden, waarin het lijkt dat mensen allerlei dingen doen of zeggen, die ze in werkelijkheid nooit gedaan of gezegd hebben). Het maken van dit soort nepbeelden is steeds gemakkelijker en het resultaat is steeds moeilijker van echt te onderscheiden. Speciale opsporingssoftware moet helpen deze fake te ontmaskeren (NOS journaal, 24-5-2021).

Big data

Over de hele wereld staan digitale apparaten die informatie verzamelen. Camera's, detectielussen, thermometers, wifitrackers, enzovoorts. Die informatie blijft bewaard in databastanden op computers. Via het internet zijn die gekoppeld. Samen vormen ze de 'big data'. De politie kan er misdrijven mee oplossen, bedrijven vinden zo hun klanten.

Het wordt minder plezierig als de data door criminelen gebruikt wordt om mensen op te lichten, of door de overheid om burgers te volgen of te dwarsbomen vanwege hun overtuiging. Niet voor niets zijn privacy en beveiliging extreem belangrijk.

0.2.6 De huisarts

Tijdens de Covid-19 pandemie werd de temperatuur van mensen op afstand gemeten om te ontdekken of ze Corona hebben. Als je de huisarts belt, is één van de vragen die je aan de telefoon krijgt: heb je koorts? Om dat zelf te meten, is de kans groot dat je gebruikt maakt van een digitale koortsthermometer. Verderop in de module gaan we die beter bekijken.



Figuur 1.5 Digitale koortsthermometer Svdmolen - Self-published work by Sudmolen, CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=1528736.rtf>

1.1 Scrum

Net als overal in de wereld van ICT is samenwerken belangrijk én ingewikkeld. Het ontwikkelen van digitale technologie vraagt inzet van mensen die allemaal aan een ander stukje van het project werken. Weten wat anderen doen, en vertellen wat jij doet is daarom nodig. Ook als je vast loopt, is het goed om hulp te krijgen. Daarnaast moet voortdurend in beeld blijven wat de gebruiker (klant) wil hebben.

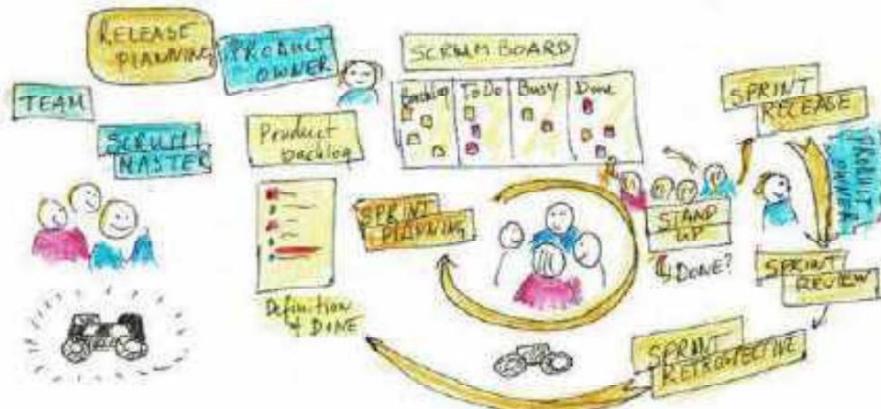
Voor de samenwerking in een project en het zo snel mogelijk opsporen van hindernissen is een methodiek ontwikkeld, die we kennen onder de naam Scrum. Het is een ‘agile’ methode, wat betekent dat het wendbaar is en zich kan aanpassen aan de situatie. De methode is ontwikkeld in de ICT-wereld voor projecten waar software ontwikkeld wordt en wordt nu bij talloze bedrijven, en ook op school, gebruikt.

Er zijn allerlei termen die gebruikt worden voor stappen in het proces, afspraken en rollen die mensen hebben. De mensen die samen aan een klus werken, het team, heeft een scrummaster, die de samenwerking binnen het team organiseert. In deze module zetten we scrum ook in. Het is belangrijk dat de teamleden van zichzelf en elkaar weten wat ze kennen en kunnen.

Een digitaal technologieproject begint meestal met een probleem, waar digitale technologie wellicht een oplossing kan zijn. We hebben in deze module ook een aantal eindopdrachten geformuleerd, die je kunt zien als probleembeschrijving. De gebruiker heeft wensen en eisen, geeft een opdracht en verwacht een oplossing. Dat is de ‘user story’.

Een goed beeld van het probleem krijgen én nadenken over mogelijke oplossingen, is de eerste fase in elk project. Je lost het probleem van een gebruiker op, want daarvoor komt hij naar jullie toe. Voor je aan het werk gaat moet je precies weten wat de bedoeling is. Als eerste beschrijf je precies wat een gebruiker wil hebben. Er is iemand die daar verantwoordelijk voor is: de product owner. Hij heeft contact met de klant. Het eindproduct is er niet in één keer. In korte periodes (sprints) worden tussenproducten gemaakt. De klant kan daar iedere keer op reageren (of de product owner reageert namens de klant).

De eerste stap naar een eindproduct is om duidelijk te krijgen wat er nodig is om het product te maken. Dat wordt de product backlog. Daarvan omschrijft het team taken en bekijkt het team hoeveel werk elke taak is. Grote taken krijgen veel punten, kleine taken weinig. De taken komen in het vak



‘backlog’ op het scrumbord.

Figuur 1.6 De Scrum-methode voor van een project

Je hoeft niet persé met een papieren scrumbord te werken. Er zijn talloze online varianten van het scrumbord om taken van een team te managen.

Taken worden niet allemaal gelijk aangepakt, maar verdeeld in groepjes die bij elkaar horen, de sprints. Een sprint duurt ongeveer twee weken. Het team staat bij het planbord (scrumbord), en plakt de taken voor een sprint bij elkaar onder ‘TO DO’. Dat zijn de taken die er voor deze sprint gedaan moeten worden. Het team verdeelt taken en noteert op de taakbriefjes wie er aan werkt. Taakbriefjes waar aan gewerkt wordt komen onder ‘BUSY’. Met één blik op het scrumbord weet je waar je team aan werkt. Vorderingen worden zichtbaar door het verplaatsen van taakbriefjes tijdens de dagelijkse ceremonie: de ‘standup’. Taken die afgerond zijn komen bij ‘DONE’. Bij taakbriefjes

die nog bij BUSY blijven hangen, vragen teamleden elkaar of de taak vordert, of er hulp nodig is. Misschien is de conclusie dat een taak in kleinere stukjes verdeeld moet worden.

Na een sprint zijn de taakbriefjes, als het allemaal soepel gaat, verplaatst naar ‘DONE’. Er is dus een stukje van het project klaar. De product owner kijkt of dit stukje van het project geslaagd is en voldoet aan de wensen van de klant. Het team kijkt terug op de sprint (sprint review), en bepaalt of er nog dingen anders moeten of vergeten zijn. Die dingen worden als taken toegevoegd aan de product backlog, en bij een volgende sprint aangepakt. Opnieuw wordt het planbord gevuld met taken voor de volgende sprint en werkt het team opnieuw een cyclus af.

Regelmatig kijken of iets dat je gemaakt hebt af is en werkt, wordt bij de ontwikkeling van computerprogramma’s veel gedaan. Werken in kleine stukjes die áf zijn is heel prettig. Daardoor blijft een project wendbaar: ‘agile’ of ‘lean and mean’. Het is prettig wanneer je hulp krijgt als je vast loopt. Als er tijdens een sprint iets anders blijkt te moeten, kan er snel bijgestuurd worden. Alles is tijdens een sprint getest. Aan het eind worden alle onderdelen samengebracht en nog een keer getest. Dan is ook duidelijk of het einddoel bereikt is. Aan het begin is duidelijk wat er gemaakt wordt, niet de exacte stappen om er te komen. Details van taken worden in een sprint bepaald en kunnen elke sprint worden bijgesteld.

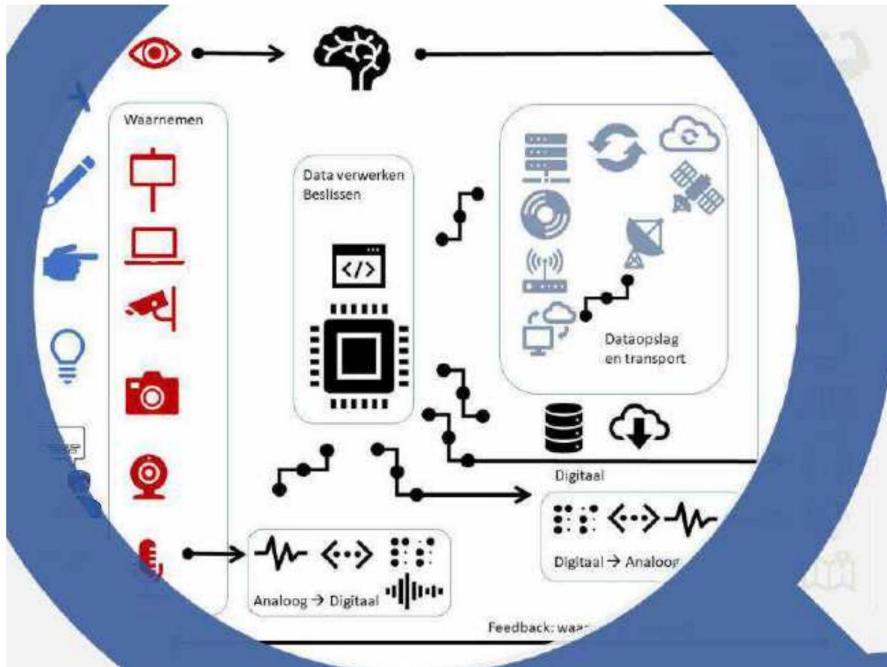
0.2.7 Scrum toepassen

In de module kun je scrum ook heel goed gebruiken, bijvoorbeeld om de eindopdracht samen aan te pakken. Precies beschrijven wat er gemaakt moet worden (design), wat daarvoor nodig is en hoe je dat als team aanpakt: dat is waar scrum voor bedoeld is. Je hebt verschillende kwaliteiten in je team nodig, zoals samenwerken, anderen helpen, schrijven, concentreren, overleggen, precies zijn, plannen, stevig doorwerken en dergelijke. De een weet hoe je code schrijft, een ander wat je wilt meten, een handleiding schrijven of praktische verpakking ontwerpen, en een derde kan bruikbare manier bedenken hoe gegevens aan een gebruiker getoond kunnen worden. Elke taak vraagt weer andere kwaliteiten in je team. Naast algemene kwaliteiten is er ook ervaring met het onderwerp van de module. Daarvoor zijn ruwweg drie mogelijkheden:

1. **Algemeen gebruiker:** Je bent iemand die digitale technologie gebruikt (bijvoorbeeld je telefoon, websites of programmatuur), maar ze zelf niet installeert of programmeert.
2. **Beheerder en ontwerper:** Je zet computers in elkaar, beheert een website of een server of denkt na over de manier waarop software zou moeten werken.
3. **Codeerder en bouwer:** Je bent bezig met het maken en aanpassen van programma’s, bouwen en testen van apparaten, solderen, drivers zoeken en installeren.

Bij de indeling van een team is het nodig om zowel op persoonlijke kwaliteiten als het inhoudelijke profiel (Algemeen gebruiker, Beheerder of Codeerder) te letten. Bij het bestuderen van de hoofdstukken komt dat onderscheid ook van pas. Je zou alles van het hoofdstuk moeten kennen en kunnen, maar het startpunt kan verschillen. Deel A is een goede start voor Algemeen gebruikers; B voor de Beheerders en C voor Codeerders. Uiteindelijk kun je elkaar uitleg geven en helpen bij de theorie en praktische vaardigheden.

0.3 Introductie op de module Digitale Technologie



0.3.1 Leerdoelen

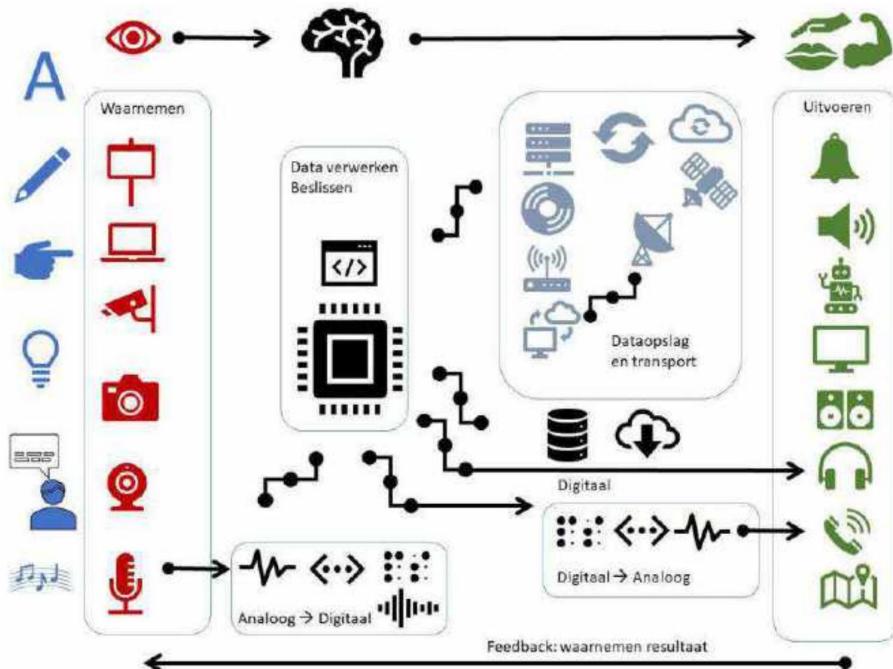
Kennis

1. Kennismaken met basisbegrippen die in de digitale wereld gebruikt worden. De volgende basisbegrippen kun je uitleggen en gebruiken: sensor, analoog-digitaal conversie, interface, data, processor, dataopslag, datatransport, datacommunicatie, actuator, cloud, ic, programma, programmeren, algoritme, digitaal-analoog conversie.
2. Opbouw van een digitaal apparaat herkennen en beschrijven aan de hand een schematische weergave (het analyseschema digitale technologie).

- Vaardigheden**
3. Je kunt de functie van een digitaal apparaat uitleggen aan de hand van de basisfuncties waarnemen, data verwerken, beslissen, dataopslag en -transport, uitvoeren en terugkoppeling.
 4. Je kunt de relatie tussen een fysieke grootheid en een numerieke (digitale) grootheid uitleggen en welke voor- en nadelen het heeft om informatie digitaal te verwerken.

In dit gedeelte maak je kennis met de basisbegrippen van digitale technologie en een manier om de onderdelen van een digitaal systeem te kunnen benoemen en herkennen: het analyseschema.

0.3.2 Basisbegrippen



Figuur 2.1 Schema digitale technologie

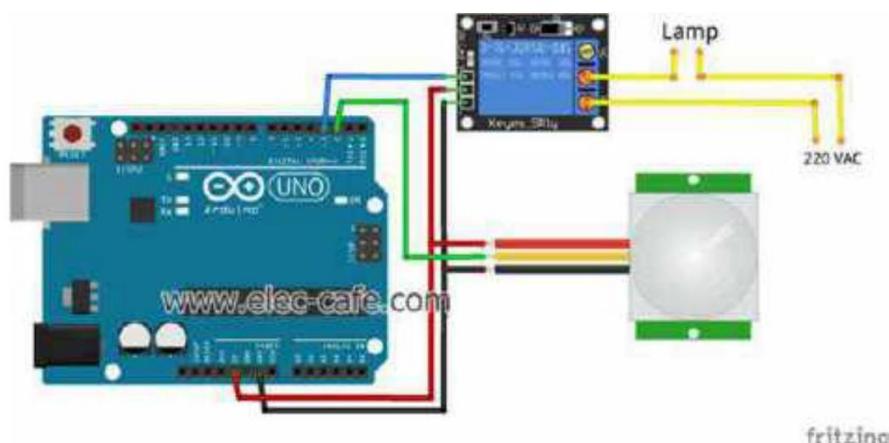
Digitale technologie maakt gebruik van ‘digitale’ apparaten. Digitaal houdt in dat het apparaat signalen gebruikt die uit maar twee verschillende waarden (bits) bestaan: nul of een. Een rijtje bits vormt samen een klein blokje informatie, bijvoorbeeld een getal of een letter.

Alle apparatuur, die bij digitale technologie wordt gebruikt, heeft overeenkomstige onderdelen en functies. Figuur 2.1 geeft hiervan een schematisch overzicht. Sommige onderdelen komen in ieder digitaal apparaat voor. Wanneer onderdelen ontbreken, betekent het vaak dat een digitaal apparaat andere apparaten of de mens nodig heeft om te kunnen functioneren. Want: digitale apparaten/digitale technologie bestaan omdat mensen daarmee werken. Altijd moet de verbinding tussen mens en machine gemaakt worden via interfaces (zoals schermen, knopjes en toetsenborden).



Figuur 2.2 Zintuig - hersenen - ledematen: waarnemen, verwerken en uitvoeren in ons lichaam.

Een digitaal systeem heeft veel overeenkomsten met onszelf. Wij hebben zintuigen om de omgeving waar te nemen, hersenen die informatie verwerken en beslissingen nemen voor ons handelen, en organen of ledematen die deze handelingen uitvoeren, zoals handen, benen of stembanden. Verandering van de omgeving nemen we weer waar met onze zintuigen, de feedback, om ons handelen weer te kunnen bijstellen.



Figuur 2.3 Sensor (voor licht), processor (Arduino) en actuator (lichtschakelaar): waarnemen, verwerken en uitvoeren in een digitaal apparaat.

Met techniek zijn zintuigen, hersenen, organen en ledematen nagemaakt. Ze heten dan: sensoren, processoren en actuatoren. In plaats van zenuwvezels lopen er in een apparaat draadjes, die de signalen doorgeven.

0.3.3 Algemene beschrijving van het schema

In het schema van een digitaal systeem (fig. 2.1) zijn vier belangrijke blokken te onderscheiden. Waarnemen - verwerken en beslissen - dataopslag en datatransport - uitvoeren. We leggen eerst uit wat er mee bedoeld wordt. In de vervolghoofdstukken gaan we elk van deze blokken verder uitwerken.

0.3.4 1. Waarnemen (hoofdstuk 3)

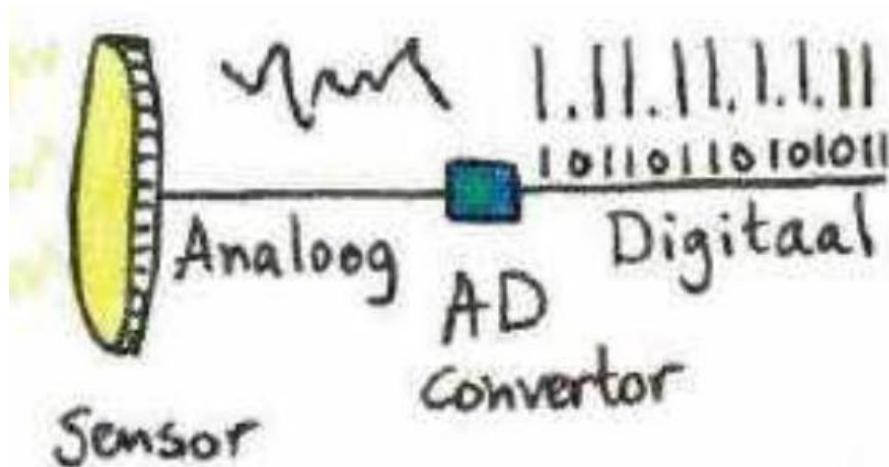


Figuur 2.4 Waarnemen

Figuur 2.5 Conversie

Als mens nemen we onze omgeving, de fysieke wereld, waar. Licht via onze ogen, aanraking met de huid, chemische stoffen in de lucht met onze neus, of bewegingen met ons evenwichtszintuig. Veel apparaten, die wij gebruiken, doen dat ook. Ze zijn uitgerust met sensoren. Een telefoon heeft bijvoorbeeld sensoren voor verplaatsing, licht (camera's), aanraking, warmte, geluid, magnetisme of radiogolven. Hiermee kunnen wij het apparaat bedienen, praten, foto's maken of onze positie bepalen. Door gegevens van sensoren op een slimme manier te combineren, kunnen veel dingen via een omweg gemeten worden, bijvoorbeeld het meten van verplaatsing door een camera te gebruiken.

Een sensor zorgt dat een fysische grootheid (bijvoorbeeld licht of temperatuur) naar een elektrisch signaal wordt omgezet. Het signaal uit de sensor is analoog (kan oneindig veel waarden aannemen) en wordt vertaald in een digitaal signaal (een reeks blokjes met een hoge óf een lage spanning). Die truc wordt uitgevoerd door een analoog-digitaal omzetter (AD converter, figuur 2.6). Het resultaat is een binair signaal (0 of 1).



Figuur 2.6 Analoog naar digitaal omzetting

Mens	Functies	Machine	Functies
Zintuig (uitwendig en inwendig)	Waarnemen geluid, temperatuur, aanraking, chemische stoffen, beweging; inwendig bijvoorbeeld bloedsuiker, hormonen, kooldioxide)	(licht, Sensor	Invoer: waarnemen van de buitenwereld (licht, temperatuur, aanraking, chemische stoffen enz.)
Zenuwcellen, hersenen	Verwerken en beslissen	Processor	Berekenen en informatie uitsturen
Geheugen	Netwerk van verbindingen tussen zenuwcellen	Dataopslag	Opslag van bits in geheugenchip of op harde schijven
Spier en klier	Uitvoeren van bewegingen (dus ook ademen, geluid maken enz.) en afgeven stoffen of hormonen (voor regeling in lichaam)	Actuator	Bewegen, licht geven, geluid
maken enz.			

2.1 Mens en digitale machine vergeleken

0.3.5 Vragen

0.3.6 2. Verwerken en beslissen (hoofdstuk 4)

Digitale signalen lopen als stroompulsjes door metaaldraadjes, flitsen als radiogolf of licht

Figuur 2.7 Verwerken en beslissen door de lucht en door glasvezel. Bij een digitaal signaal maakt het weinig uit hoe groot elk pulsje is: als er maar onderscheid te maken is tussen wel en geen puls (1 of 0). Dat voorkomt verlies van het signaal.

Het verwerken van de pulsen in een digitaal systeem gebeurt in heel kleine chips (enkele vierkante mm). Deze microprocessors zitten vol elektronische schakelingen, geïntegreerde circuits (IC), die miljoenen kleine onderling verbonden onderdelen bevatten. Dat zijn o.a. transistoren (schakelaars),

diodes (die elektronen in één richting doorlaten) en condensatoren (voor het tijdelijk opslaan van elektronen).

Een IC zet de spanning op uitgaande metaaldraadjes aan en uit op basis van binnengekomende signalen. Wat de IC precies doet, wordt bepaald door de bedrading én door de stand van de schakelaars. Door tevoren de schakelaars in een bepaalde stand te zetten met stroompulsjes, wordt een IC geprogrammeerd. Soms ligt dat programma vast: de processor is dan gemaakt om altijd hetzelfde te doen. Andere processoren zijn telkens opnieuw programmeerbaar. Alle schakelaars komen in de beginstand als de IC geen stroom meer krijgt (reset). De software kan de IC dan opnieuw programmeren. Die van je mobiel heeft telkens andere klusjes, dus vertelt de app (applicatie, het programma) wat er moet gebeuren. Wat de IC doet wordt bepaald door de regels (algoritmen, zie kader) in de software. Daardoor is een processor flexibel te gebruiken.

0.3.7 Algoritme

Het woord algoritme herinnert aan de Arabische wiskundige Al-guarijsmi, die ook de basis legde voor Al-gabbr (algebra). Hij beschreef het oplossen van een (reken)probleem in stappen, die samen een algoritme vormen, dat je kunt volgen om een probleem op te lossen.

Dat doe je in je dagelijks leven voortdurend. Als je een boterham met kaas wilt klaarmaken, volg je een aantal stappen. Welke je precies volgt is afhankelijk van de situatie.

Werk je met een groot stuk kaas of plakjes? Gebruik je een kaasschaaf of een mes? Als je de stappen precies noteert voor elke situatie, krijg je verschillende algoritmen. Verschillende algoritmen kunnen dus dezelfde uitkomst hebben.

0.3.8 3. Datatransport en dataopslag (hoofdstuk 5)

Figuur 2.8 Opslag en transport van data

Figuur 2.9 Dataopslag



Figuur 2.10 Uitvoeren

Figuur 2.11 Conversie

De enen en nullen in digitale signalen worden door de processor voortdurend verplaatst in zijn eigen werkgeheugen en als digitale data opgeslagen op harde schijven of geheugenchips. Hiermee kan

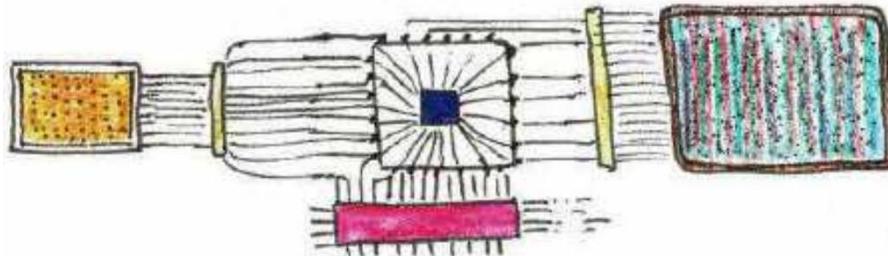
data op een ander moment verder worden bewerkt of gebruikt, maar ook verplaatst en ergens anders opgeslagen worden. Opslag in de ‘cloud’ komt er op neer dat (een kopie van) die data op de schijven van een of meerdere digitale opslagplaatsen ergens ter wereld terecht komt.

Als je een foto met je mobiel maakt en deelt met vrienden, kopiëert en verstuur je mobiel de data. Dat gaat draadloos via 4G of 5G naar een telefoonmast, of via wifi naar een netwerk van koper en glasvezel, en komt via routers terecht bij de server van de app met opslagruimte in een datacenter. Google, Microsoft, Facebook, Zoom en de bedrijven achter al de apps, gebruiken enorme datacenters én supersnelle verbindingen met hun servers om al die digitale data te kunnen ontvangen, opslaan, bewerken én weer door te sturen. Heel dynamisch dus. De foto die je maakt, en die vlak daarna te zien is op de mobiel van je vriendin die naast je staat, heeft al binnen een seconde de wereld rondgereisd.

Het beschermen van computersystemen, cybersecurity, is een belangrijk thema bij transport, opslag en gebruik van digitale data. Het is belangrijk dat niet de hele wereld jouw foto’s kan bekijken als je dat niet wilt. Ook verwijderen van data blijkt lastig, omdat in een paar seconden een aantal kopieën op allerlei servers staan, waar je geen controle meer over hebt.

0.3.9 4. Uitvoeren (hoofdstuk 6)

Aan pulsjes, de bits, in een digitaal systeem heb je alleen maar iets, als je er iets mee kunt doen. Digitale data van een foto wil je graag op een schermpje kunnen bekijken. Een rijtje bits kan gebruikt worden om één pixel van een schermpje licht te laten geven. Er zijn miljoenen enen en nullen nodig om een afbeelding of tekst op een beeldscherm te laten zien. Maar je kunt ook geluid maken via een stroompje door de spoel in een luidspreker. Op die manier vormen een scherm, luidspreker of motortje de uitvoer van een digitaal systeem en worden daarom actuatoren genoemd. Een digitaal signaal wordt vaak direct gebruikt om een beeldpuntje aan of uit te schakelen. Om een luidspreker te laten werken, moet het digitale signaal eerst omgezet worden in een analoog signaal door een digitaal-analoog omzetter (DA-converter). Dit analoge stroompje kan de luidspreker laten werken om je muziek af te spelen.



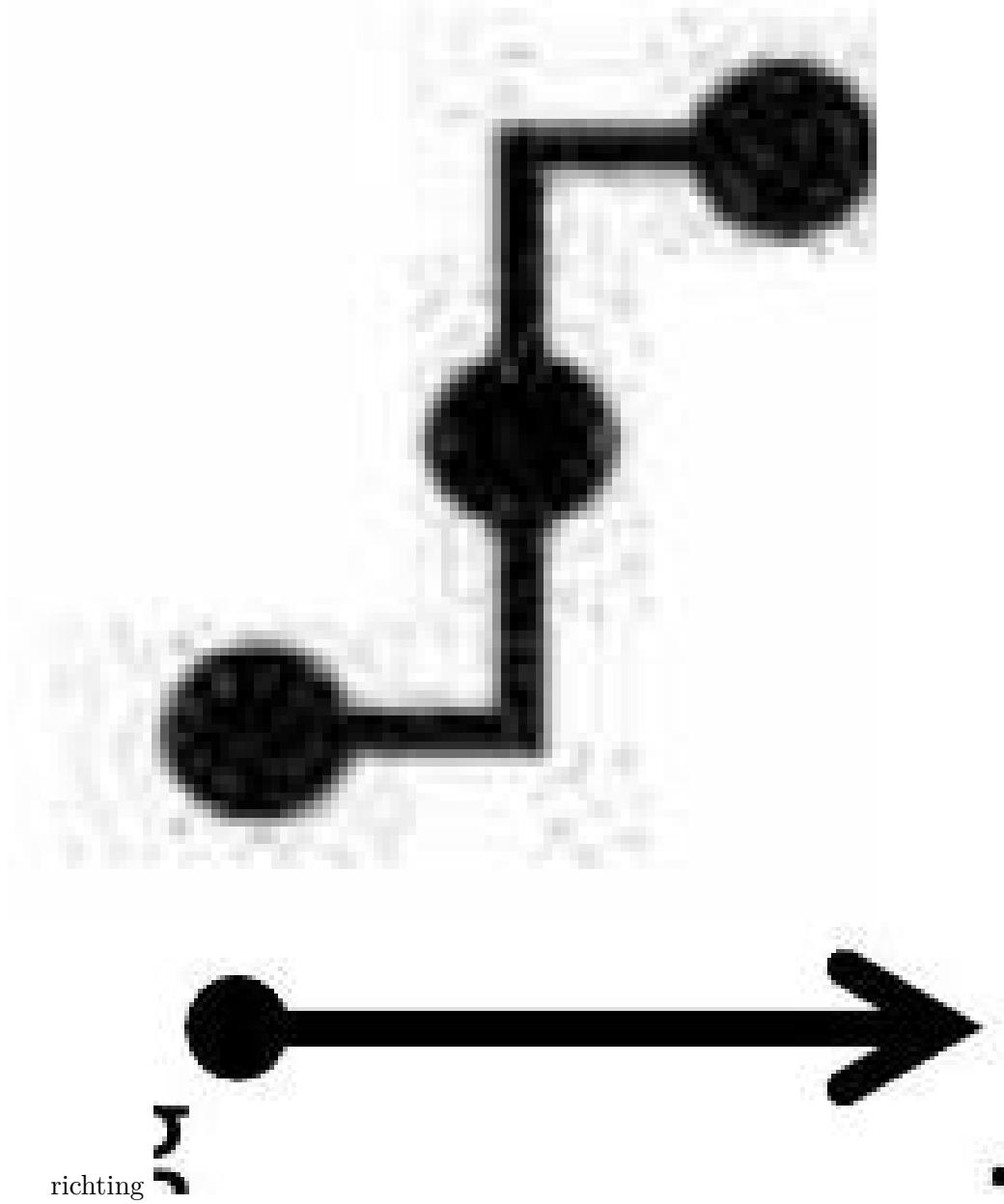
Figuur 2.12 Digitale camera: sensor - processor - opslag - actuator (beeldscherm)

0.3.10 5. Feedback (geen eigen hoofdstuk)

In het analyseschema staat het begrip feedback (terugkoppeling): je neemt met sensoren waar hoe een fysische grootheid in de omgeving verandert als een actuator actief is. Bij het nauwkeurig en correct uitvoeren van opdrachten, is het enorm belangrijk dat er controle is op de effecten. Als jij water in je glas giet, blijf je kijken wat je doet, om te voorkomen dat je er naast giet of het glas overstroomt. Net als jij je ogen gebruikt om waar te nemen welk effect je handelingen hebben (terugkoppelen), moet ook een digitaal systeem zijn sensoren gebruiken om te ontdekken wat het effect is van de opdrachten die de actuatoren uitvoeren. Die sensoren ontbreken vaak, omdat digitale systemen met en voor de mens werken. Daardoor doet de gebruiker zelf vaak de terugkoppeling, bijvoorbeeld het regelen van het geluidvolume als je muziek beluistert. Hoe automatischer en zelfstandiger een apparaat werkt, hoe belangrijker de rol van terugkoppeling is. Als heel nauwkeurig te voorspellen is wat een actuator doet (bijvoorbeeld: een beeldscherm), is het minder belangrijk (maar wel prettig) om allerlei sensoren te hebben. Je mobiel gebruikt een lichtsensor om te bepalen hoeveel omgevingslicht er is en welke kleur dat heeft. Het scherm is op die manier in fel daglicht en in het donker goed te gebruiken.

0.3.11 6. Verbindingen (hoofdstuk 5)

Tussen de vakken in het analyseschema zijn verbindingen getekend. Hiermee wordt aangegeven dat informatie van een vak naar een ander vak wordt overgedragen. Van een onderdeel, zoals een sensor, loopt elektriciteit, via stekkertjes en draadjes of via pinnetjes op de printplaat, naar de processor. De informatiestroom kan beide kanten op



. Tussen computers worden netwerkkabels of glasvezel gebruikt of draadloze overdacht via radiogolven. Een deel van die verbindingen hoort bij het vakje datatransport en -opslag.



Photo by Mika Baumeister on Unsplash <https://www.vecteezy.com/free-photos> Pixelpatroon, Wikipedia: Bayer filter

0.3.12 Voorbeeldschema: foto's met je mobiel.

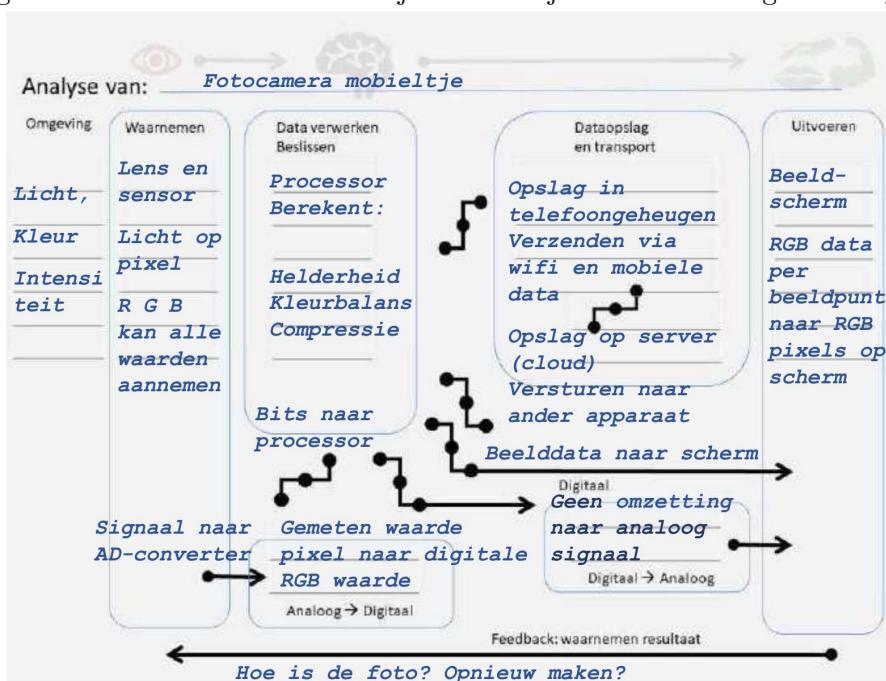
Een mobiele telefoon heeft een camera waarmee je foto's en filmpjes maakt. De camera heeft daarvoor een sensor die via een lens licht opvangt. De sensor is verdeeld in kleine blokjes (pixels), die binnenvkomend licht vertalen in een elektrisch stroompje. Naast elkaar zitten een pixel voor R (rood), G (groen) en B (blauw) licht, die samen één RGB beeldpunt vormen. De waarde voor R, G en B is analoog, dat wil zeggen dat het alle mogelijke waarden kan aannemen.

In de camerasensor wordt het analoge signaal vertaald naar een reeks enen en nullen: een digitale waarde. Daarvoor gebruikt de sensor een analog-digitaal omzetter. Het digitale signaal uit de sensor is een enorme tabel van miljoenen getallen, die voor ieder beeldpunt de RGB waarde weergeeft in (een rijtje van 8) bits. Een goede camera heeft al snel 5 tot 10 miljoen beeldpuntjes (megapixel). Voor één RGB beeld moeten minstens 3 (RGB) x 5 miljoen (pixels) x 8 bits verstuurd worden. Ruwe beelden uit de camera gaan digitaal naar de processor in de telefoon. Die processor verwerkt de informatie en berekent bijvoorbeeld hoe het beeld aangepast moet worden. Correctie voor tegenlicht (achtergrond veel te licht) of witbalans (gele foto in kunstlicht) doet de processor. Het fotobestand wordt kleiner gemaakt; als er honderd enen achter elkaar staan, kun je dat verkleinen tot "1, 100 keer". Dat heet datacompressie.

Als dat allemaal gebeurd is, gaat de afbeelding naar het schermpje op je mobiel. Een schermpje is een actuator: de digitale informatie van de foto gaat naar de beeldpuntjes op het scherm (ook R, G en B). Bekijk het scherm van je mobiel maar eens met een vergrootglas, dan zie je dat.

Tegelijk slaat de mobiel het op bij de foto's (op het camerageheugen). Automatische online opslag (de app foto's) of versturen naar een app (Instagram, Whatsapp of iets anders) betekent dat je mobiel de digitale informatie stuurt via een draadloze verbinding (wifi, 4 of 5G) naar een server die het dataverkeer voor de app regelt. Je foto staat zo in enkele seconden op meerdere plaatsen in de wereld. Als je de foto via een app met een vriend(in) deelt, stuurt de server deze onmiddellijk door, en geeft de mobiel of computer een melding. Voor je het weet is het fotobestand bewerkt, opgeslagen, gedeeld en op meerdere schermen getoond.

Met deze informatie is het analyseschema (figuur 2.6) als voorbeeld ingevuld. Zo ontdek je hoe je andere digitale systemen kunt beschrijven.



Figuur 2.13 Ingevuld analyseschema voor camera van mobiele telefoon

0.3.13 Wat ga je doen?

- Haal een digitale koortsthermometer uit elkaar (of gebruik er een die al uit elkaar gehaald is) en bekijk welke onderdelen aanwezig zijn.
- Verdeel de onderdelen over de vier basiscategorieën aan

de hand van het analyseschema (leeg schema achterin de lesmodule of verkrijgbaar als los werkblad): waarnemen, verwerken en beslissen, datacommunicatie en -opslag, en uitvoer. Is er omzetting van analoog naar digitaal signaal en omgekeerd? Zo ja, waar? c. Wat voor omgevingssignaal of -signalen registreert de koortsthermometer? d. Welke sensor(en) heeft de koortsthermometer? e. Welke actuator(en) zijn er aanwezig?

0.3.14 Nabespreking koortsthermometer

Een digitale koortsthermometer heeft in het algemeen geen datacommunicatie met de buitenwereld. Een mens bedient het apparaatje en die gebruikt de getoonde gegevens.

De temperatuursensor geeft altijd een waarde aan. Als het meetsysteem aangezet wordt (schakelaar, ook een sensor) wordt de weerstand gelezen en door de AD converter omgezet in een digitaal signaal. De processor bepaalt of de waarde van de weerstand verandert (vergelijk metingen). Als de waarde gedurende 30 seconden niet meer verandert is dat de waarde die op het schermpje (uitvoer) moet komen. De processor geeft de opdracht om een piepje (uitvoer via luidsprekertje) te geven. De waarde die op het schermpje getoond wordt blijft in een geheugen achter. Indrukken van de schakelaar zorgt dat het apparaat uit gaat. Maar: dit moet vertaald worden in een opdracht aan de processor om te stoppen en het display uit te schakelen. Bij aanschakelen moet de processor alles weer aanschakelen en de laatst gemeten hoogste waarde op het scherm laten zien en na enkele seconden opnieuw beginnen met meten. Voor een koortsthermometer kan ook een minimale waarde ingesteld zijn om een meting te laten starten, bijvoorbeeld 30° Celcius.

Samenwerken met andere apparaten is niet nodig. Het apparaat is ook niet afhankelijk van andere apparaten om zijn werk te doen, het is een zelfstandig, standalone, apparaat.

In het vervolg van de module kom je meerdere systemen tegen, die we aan de hand van dit schema gaan beschrijven.

Jaarlijks belanden miljoenen digitale apparaten op de afvalberg (e-waste). Ook op school of bij jou thuis zijn vast oude apparaten te vinden, die binnenkort worden afgevoerd. Ook zo'n apparaat kun je uit elkaar halen en daarmee de opdracht uitvoeren. Op dezelfde manier kun je andere systemen analyseren. Kies zo'n apparaatje uit en probeer het schema er voor in te vullen. Let op: wanneer er veel vakjes leeg blijven, bedenk dan welke (andere) apparaten er nodig zijn om de functie van ieder vakje te vullen.

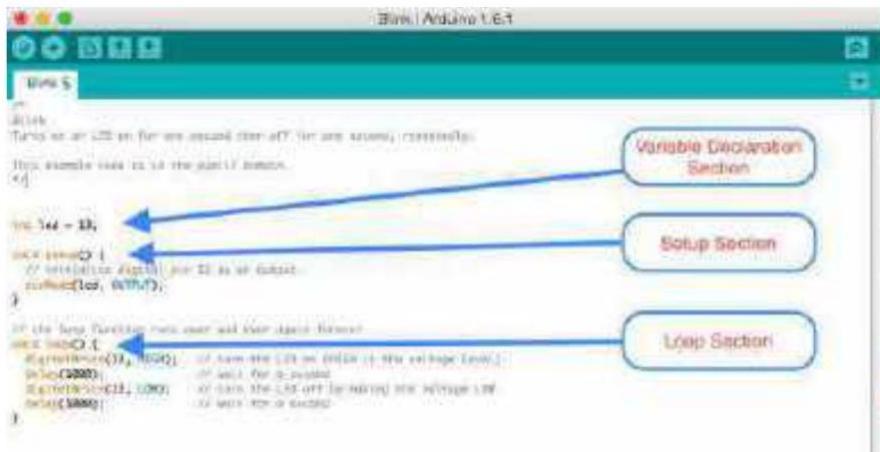
Als het om e-waste gaat: wat gooï je allemaal weg en hoe kun je dat hergebruiken?

Digitale technologie is overal om ons heen. Er zijn diverse manieren ontwikkeld om ermee te experimenteren, zelf te bouwen en programmeren. In dit gedeelte maak je kennis met de meest gebruikte platforms, zodat je daar verderop in de module je eigen projecten mee kunt uitvoeren.

0.3.15 Basis van de hardware ontdekken met Arduino, Micro:bit of Raspberry pi

Als je een goed beeld wilt krijgen hoe digitale systemen in elkaar zitten, helpt het om ze zelf in elkaar te zetten en te programmeren. Daar zijn inmiddels veel mooie materialen voor beschikbaar, zoals 'Single Board computers' als Arduino, Micro:bit en Raspberry Pi. Misschien heb je er al wat ervaring mee, want deze kleine apparaatjes worden steeds meer gebruikt op basis- en middelbare scholen.

0.3.16 Microcomputer en microcontroller



Figuur 2.14 Arduino code

De Raspberry Pi is een microcomputer, een kleine computer waar je een scherm en een toetsenbord op kunt aansluiten, die een verbinding heeft met het internet, en die je kunt programmeren. Eigenlijk een gewone computer, maar dan veel kleiner en met een aantal aansluitingen waarmee je iets uit kunt lezen (een druktoets) of aan kunt sturen (een LED). De Arduino en de Micro:bit zijn microcontrollers: je kunt er een programma opzetten, waarmee je een sensor uitleest en bijvoorbeeld een LCD display aanstuurt. Je schrijft dat programma op een computer en ‘upload’ het op de microcontroller. Je kunt maar één programma tegelijk uitvoeren op deze microcontrollers, maar je kunt ‘m wel steeds opnieuw programmeren.

Alle drie gebruiken ze Open Source software. Dat betekent dat de computercode voor iedereen toegankelijk is. De programmeurs delen hun code met anderen en ze schrijven samen aan de software, vaak in hun vrije tijd. Die Open Source gedachte is overgenomen door de gebruikers. Als je iets gemaakt hebt worden de resultaten vaak gedeeld met anderen. Overal op het internet zijn voorbeelden te vinden van projecten, die je na kunt bouwen. Zo leer je van elkaars projecten.

Alleen Arduino gaat nog een stapje verder in de Open Source gedachte, omdat het Open Source Hardware is. Het bouwplan van de Arduino staat online en iedereen mag er een (na)maken. Als je wilt, kun je zelfs je eigen Arduino bouwen (<https://www.arduino.cc/en/main/standalone> of <https://www.instructables.com/How-to-make-your-own-Arduino-board/>).

Zo komt het dat je in China voor heel weinig geld een

0.3.17 Programmeren

Een computer werkt met instructies (een programma, software) geschreven in een programmeertaal, zoals Arduino taal, C, Python, machinetaal, Visual Basic enzovoorts. Een programmeur schrijft zulke software.

Elke taal definieert , heeft **commando’s**, <>en een duidelijke markering van {begin} en {eind} van een opdracht. In elk goed programma (code) staat uitleg (//comments//). De programmeur beschrijft wat het blokje code doet.

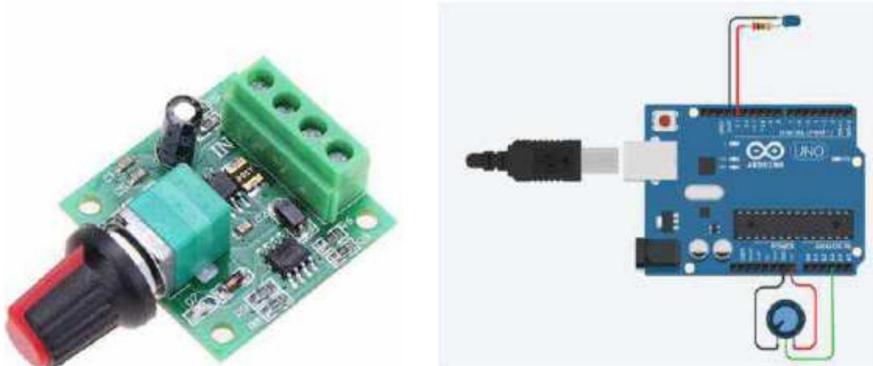
Een werkend programma is fijn. Maar de uitleg is nog fijner, omdat iemand anders jaren later het programma kan begrijpen en aanpassen.

Arduino kloon kunt kopen, terwijl de Micro:bit en Raspberry Pi alleen gemaakt worden door fabrikanten in de EU. Die laatste zijn vaak wat duurder, maar ook wat steviger en betrouwbaarder.

0.3.18 Waarom zijn ze handig?

Waarom zijn microcontrollers en microcomputers zo populair? Voor het antwoord op die vraag moeten we terug in de tijd. Vroeger, voor de microcontrollers, maakten elektronische apparaten gebruik van printplaten: plaatjes waarop de weerstanden, condensatoren en transistoren gesoldeerd zijn, onderling verbonden met een netwerk van geleidende koperbanen. Voor elke kleine verandering moet zo’n printje opnieuw gemaakt worden.

Kijk maar eens naar deze dimmer (fig. 2.16), met een printplaatje met elektronica, een potmeter, transistoren en condensatoren. Klein maar fijn, maar niet flexibel. Want stel dat ik mijn LED niet wil dimmen met een potmeter, maar met een drukschakelaar, die door lang en kort drukken de LED aan en uitzet én dimt? Dan heb je een heel ander printje nodig.



Figuur 2.16 Dimmen voor LEDs: een printje met variabele draaiweerstand (links) en digitaal met Arduino (rechts)

Een microcontroller werkt veel flexibeler, omdat je 'm kunt programmeren. Je sluit er een LED en een potmeter op aan en schrijft een computerprogramma om de potmeter uit te lezen. Die waarde gebruik je, na wat rekenwerk, om de helderheid van de LED aan te sturen. Maar vervang de potmeter door een drukknop en verander wat aan het computerprogramma, dan kun je de helderheid ook regelen met de tijd die de schakelaar is ingedrukt en hem met een korte klik aan en weer uit zetten.

0.3.19 Leren

Om te leren werken met Arduino, Micro:bit en Raspberry Pi heb je eigenlijk geen lagere of middelbare school meer nodig. Op internet wemelt het van de uitleg, lessen en filmpjes. Op de websites van deze single board computers zijn ook lessen, uitleg en achtergrondinformatie beschikbaar. Als je vastloopt en het niet meer weet, zijn er internetforums waar andere gebruikers je kunnen helpen als je vragen hebt.

0.3.20 Beroepen

In de wereld van de digitale technologie is veel te doen. De kans is groot dat je zelf, via een onverwachte route, in deze wereld terecht komt.

FutureNL heeft hier de serie DigiDoeners over gemaakt. De labels van schoolvakken laten zien dat het heel breed is. <https://www.lessonup.com/nl/channel/futurenl/series/7b1fc5805d7f73078018673d> <https://www.lessonup.com/nl/channel/futurenl/series/b8914c0f6056a12da919c22a>

Het leuke is dat je al snel begint met het maken van projecten. Je maakt een kleine schakeling, schrijft wat code, en voila, daar heb je een digitale thermometer, een stoplicht of een robot. Er zijn heel erg veel websites waar je leuke projecten kunt vinden, die je kunt maken. Probeer er eens een paar te vinden en maak zo'n project.

Let wel op. Techniek is soms taai. Dingen werken niet, je snapt niet waarom en je zoekt je suf. Dan blijkt dat je ergens iets niet gezien hebt, dat er iets stuk was of dat je een fout(je!) hebt gemaakt met grote gevolgen. That's the life of an electronics nerd, sorry about that. Hou vol, blijf nadenken en redeneren, Vraag hulp (maar denk vooral eerst zelf na) aan een klasgenoot, een TOA of docent of op Internet. Uiteindelijk komt het bijna altijd goed.

Mocht je nou afvragen waarom je deze tekst hebt gelezen, omdat alles toch online te vinden is?! We hebben hier een globaal overzicht gegeven en proberen uit te leggen, waarom dingen zijn zoals ze zijn. Wat hier staat vind je dan weer niet online, omdat dat vaak gedetailleerd is of op een niveau, dat voor beginners te hoog gegrepen is. En natuurlijk, omdat dit het vak nlt is.

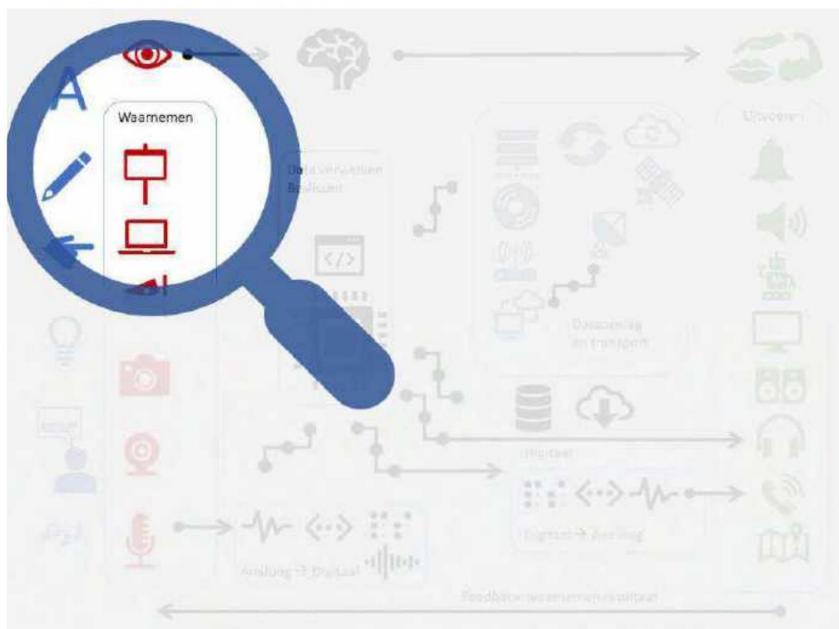
Je leert veel door dingen zelf te maken. We geven in het laatste deel van elk hoofdstuk maakopdrachten waar je (in overleg met docent en TOA) een keuze uit maakt.

0.4 Keuzeopdrachten

Op de leerlingenwebsite bij deze module zijn praktische uitleg en maakopdrachten te vinden. Als je ergens mee aan de slag wilt, overleg met je docent of TOA wat er op school mogelijk is. De instructie heeft een vaste structuur:

0.4.1 Titel opdracht

1. Korte beschrijving: wat is het en waar is het voor bedoeld? Hoe ingewikkeld?
2. Wat heb je nodig? (materiaallijst)
3. Bouwinstructie
4. Software / code die gebruikt wordt en waar te vinden
5. Wat hebben we geleerd over de componenten, code en principes (verwijzen naar uitleg in de bijlagen).
6. Waar kan (de kennis uit) dit project voor worden gebruikt in de praktijk?



7. Waarnemen en sensoren

0.4.2 Leerdoelen

Kennis

1. Je kunt uitleggen dat digitale systemen verschillende soorten informatie kunnen waarnemen
2. Je kunt uitleggen dat voor elk soort informatie verschillende sensoren beschikbaar zijn
3. Je hebt kennismemaakt met enkele digitale sensoren (zoals de TMP36)
4. Je kunt uitleggen dat sensoren vrijwel altijd analoge informatie omzetten in digitale informatie
5. Je kunt uitleggen dat in het dagelijks leven sensoren een belangrijke rol spelen en kunt aangeven waar je verschillende sensoren tegenkomt

Vaardigheden

6. Je hebt ervaren dat het noodzakelijk is om sensoren te kalibreren om ze goed te kunnen gebruiken en je weet hoe je dit uitvoert

7. Je hebt ervaren dat de digitale informatie van sensoren in een digitaal systeem in verschillende typen variabelen opgeslagen kan worden
8. Je kunt uitleggen hoe informatie in het werkgeheugen van een digitaal systeem is opgeslagen en dat dit in verschillende programmeertalen verrassend veel overeenkomsten vertoont.

0.4.3 De omgeving waarnemen

Een digitaal apparaat moet de omgeving kunnen waarnemen. Dat is nodig wanneer een mens het apparaat wil gebruiken. Bij een computer zijn daar het toetsenbord, de camera en muis voor bedoeld. Een druktoets is een sensor, die aanraking waarneemt en die deze informatie doorgeeft aan de processor. Toch kan bij een apparaat zowel invoer- als uitvoerinterface met de mens ontbreken. In dat geval vormt het apparaat een stukje van het geheel, dat met andere apparaten samenwerkt. Je kunt een draadloos toetsenbord en een draadloze muis zien als aparte, maar niet los van elkaar werkende, digitale apparaten. Immers: de muis is bedoeld om bewegingen en keuzes van de gebruiker aan de computer door te geven. In een computermuis zitten meerdere sensoren. Samen zorgen ze dat veranderingen in de fysieke omgeving (vingerbeweging, schuifbeweging) worden omgezet in een digitaal signaal dat via een draadje of via radiogolven aan een ontvanger (ook een sensor) in de computer wordt doorgegeven.



Figuur 3.1 Interface: Invoer in het systeem

Elke sensor in de elektronica of digitale technologie zorgt uiteindelijk dat een verandering van een fysische grootheid (bijvoorbeeld licht, vochtigheid, temperatuur, beweging of wat dan ook) een meetbaar spanningsverschil oplevert.

0.4.4 Temperatuursensor

Temperatuur heeft invloed op bijvoorbeeld de dichtheid van stoffen. Het vloeibare metaal kwik zet uit als het warm wordt, net als alcohol. Doe de vloeistof in een buisje, zet er streepjes naast en je kunt de krimp en uitzetting gebruiken als maat voor de temperatuur: de thermometer. Elektrische stroom, die door metaal of ander geleidend materiaal gaat, ondervindt weerstand die varieert met de

temperatuur. Dat principe wordt gebruikt om een temperatuursensor voor een elektronische thermometer te maken. De weerstand is te bepalen door een kleine spanning (bijvoorbeeld 1 V) op de sensor te zetten. Als de temperatuur verandert, zal het spanningsverschil over de weerstand ook veranderen. Voor elektronische metingen wordt tegenwoordig vaak een NTC schakeling als sensor



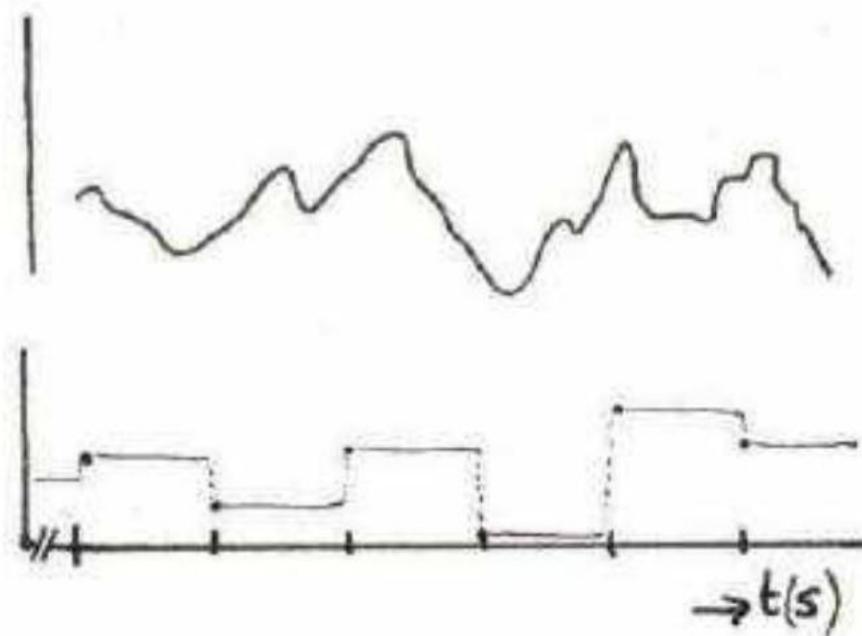
gebruikt.

Figuur 3.2 Interface: uitvoer van het systeem

De eerste stap voor elektronisch meten is: een analoge fysische grootheid (temperatuur) door de (temperatuur)sensor laten vertalen in een analoge spanning (V). Analoog betekent: kan alle waarden aannemen. Bij het aflezen van de sensor noteren we de waarde als een getal (discreet). Een gemeten spanning tussen 0,2 en 0,27 V komt bijvoorbeeld overeen met een temperatuur van 30 graden Celcius. Deze waarde moet op het schermpje komen. Elke spanning tussen 0,20 en 0,27 V levert dezelfde waarde 30 op. Elke spanning tussen 0,17-0,20 V wordt 31, enzovoorts. Preciezer uitlezen betekent ook dat de temperatuur in kleinere stapjes kan worden weergegeven, bijvoorbeeld 30,7 graden C. We houden rekening met afleesfouten. De afgelezen 30 graden is een benadering, waarbij de werkelijke waarde tussen 29,5 en 30,5 graden kan liggen. Bij 30,7 ligt de werkelijke waarde tussen 30,65 en 30,75. Hoe groter de nauwkeurigheid van de thermometer, hoe dichter deze foutgrenzen bij elkaar liggen.

0.4.5 Analoog-digitaal omzetting van signaal uit sensor

Een tweede stap is nodig om van een discreet signaal een digitaal signaal te maken. Een spanning in de sensor (bijvoorbeeld 0,25 V) wordt door een schakeling, de analoog-digitaal omzetter (ADconverter), vertaald naar een reeks bits (bijvoorbeeld 11011110). Deze schakeling kan bij de sensor ingebouwd zijn, waardoor een sensor direct op een digitale ingang aangesloten kan worden. Veel sensoren geven alleen een variabele spanning (V) af. Dan moet een AD-convertor op de processor dit omzetten in een reeks bits (zie figuur 2.6). De chip van de Arduino heeft hier een analoge ingang voor en doet zelf de AD-conversie. Het is dus belangrijk om te weten wat voor soort signaal de sensor stuurt om de juiste aansluiting te maken.



Hoeveel data?

Figuur 3.4 Sampling: aantal metingen/s moet passen bij de verandering van het signaal

Bij het aflezen van de sensor is naast de gewenste nauwkeurigheid ook van belang hoe vaak je dat doet en waar de meetresultaten voor nodig zijn. Nauwkeurigheid wordt ook bepaald door de snelheid waarmee de fysische grootheid verandert. Een langzaam veranderende grootheid, zoals lichaamstemperatuur, meet je af en toe (alleen als je ziek bent) en dan ook maar een paar keer per dag. Ook de buitentemperatuur zal in één seconde niet sterk veranderen. Je kunt ermee volstaan om bijvoorbeeld eens per uur af te lezen. We noemen dat bemonsteren of een sample nemen. Hoe sneller er verandering is, hoe groter de ‘samplefrequentie’ moet zijn. De vuistregel is: als een grootheid verandert, moet je twee keer zo vaak een sample nemen als de snelste verandering. Als dat 100x per seconde (100 Hz) is, moet je 200x per seconde (200 Hz) samplen. Vaker samplen betekent ook dat er meer data ontstaan.

0.4.6 Kalibreren en ijkken

Als je met een metalen draadje of een schakeling van transistors en diodes een temperatuursensor gemaakt hebt, wil je weten welke temperatuur er bij welke spanning hoort. Het bepalen van het verband tussen de temperatuur en de spanning (en omzetting daarvan naar een digitale waarde), noemen we het kalibreren van een temperatuursensor. Dat gebeurt bijvoorbeeld met smeltend ijs (dat is 0 graden C) en kokend water (geeft de waarde 100 graden C) of door te vergelijken met een geklakte standaardthermometer. Als je dit bij een serie temperaturen in een grafiek zet, ontstaat een ijklijn, waarin de relatie tussen de gemeten grootheid en de waarde van het signaal staat. Voor andere grootheden als licht of vochtigheid moet een vergelijkbare methode gebruikt worden.

0.4.7 Beschrijving van diverse sensoren in digitale platforms

Op de leerlingensite van de module (https://maken.wikiwijs.nl/171772/Schakelmodule_Digitale_Technologie_online_materiaal) is een overzicht opgenomen van sensoren en de manier waarop ze gebruikt kunnen worden. Raadpleeg die informatie bij de maakopdrachten in deel C.

0.4.8 Betekenis van sensoren in een context: het sporthorloge

Sensoren voor hartslag, snelheid, positie

Je gebruikt steeds meer apparaten die met internet verbonden zijn: muziek luisteren via Spotify, de verwarming aanzetten met een app, video's kijken op je telefoon via Youtube, de lamp aan en uit

zetten via wifi of Bluetooth, of betalen via je telefoon. We noemen dit ook wel het Internet of Things (*IoT*). In huis gebeurt dat vaak al via wifi. Dit is een van de redenen dat 5G wordt aangelegd, zodat alle apparaten heel snel met elkaar kunnen communiceren via een mobiele internet verbinding.

In deze opdracht ga je voor een situatie uitzoeken hoe het werkt: een sporthorloge en een app die de metingen kan vastleggen. Je denkt na over vragen als: via welke stappen wordt de informatie verzonden? Waar is beveiliging van gegevens van belang? Waar wordt de informatie opgeslagen? Welke gegevens zijn nodig om de taak uit te voeren? Etc.

Strava app

We gaan kijken naar een app waarmee je sportprestaties kunt vastleggen. Strava is wereldwijd een van de grootste sportapps, maar er zijn diverse andere die je ook kunt gebruiken. Jullie gaan aan de slag met een sporthorloge. Misschien heb jij zelf of iemand anders in je omgeving een sporthorloge. Een sporthorloge geeft een compleet beeld van digitale technologie, omdat het meet, communiceert, data verwerkt en inzichtelijk maakt. Verder speelt beveiliging van je gegevens een rol.

Sporthorloges zijn in alle soorten en maten verkrijgbaar, met prijzen variërend van 35 tot 900 euro.

Als je geen sporthorloge hebt dan is het installeren van Strava op je telefoon ook genoeg. Naast Strava zijn er ook andere apps. Als je al een andere sportapp gebruikt dan kun je die ook gebruiken voor de opdrachten. Vermeld dat dan wel.

Je onderzoekt een sporthorloge in combinatie met het gebruik van Strava. Hoewel er allerlei verschillende platforms zijn waarop



Figuur 3.6 Een sporthorloge Bron: Coolblue



Figuur 3.7 Screenshot van Endomondo van een fietsroute van Erik Dekker (tijdens de lock-down, in plaats van Italië) en overzichten van de data die de app verzamelt en presenteert.

Activiteiten die je met het sporthorloge of je telefoon registreert, levert gegevens op. Dat zijn waarden die uit een reeks sensoren komen. Die gegevens uit je sporthorloge / de sportapp op je telefoon worden via je telefoon naar Strava gestuurd en daar worden de gegevens in jouw account geplaatst. De positiegegevens uit de gps van je device worden op een kaart getekend en de gegevens van andere sporters worden daarmee vergeleken. Vervolgens wordt die informatie weer naar je telefoon of computer gestuurd om het daar te kunnen bekijken en naar de mensen die je volgt gaat een berichtje van wat je hebt gedaan. Vervolgens reageert iemand en die opmerking gaat via internet naar Strava en weer naar jouw telefoon.

Als je een sportactiviteit doet met je telefoon of sporthorloge, kun je de gegevens van die activiteit via een app in kaarten en grafieken te zien krijgen. De volgende gegevens zijn o.a. te zien: de gemiddelde snelheid, hartslag, aantal stappen per minuut, hoogte, afstand en tijd van de activiteit.



Figuur 3.8 Screenshots van de gegevens van een sportactiviteit in Strava: een rondje hardlopen.

De verwerking van dergelijke meetgegevens is vaak een black box. Je weet ongeveer wat er ingaat en wat er uitkomt, maar wat er in de black box gebeurt is onduidelijk. Het onderstaande analyseschema is bedoeld om inzicht te krijgen wat er met de informatie gebeurt en welke processen een rol spelen.

We gaan kijken naar wat de invoer is, wat er op de plaats van de streepjes gebeurt en welke processen spelen in de black box.

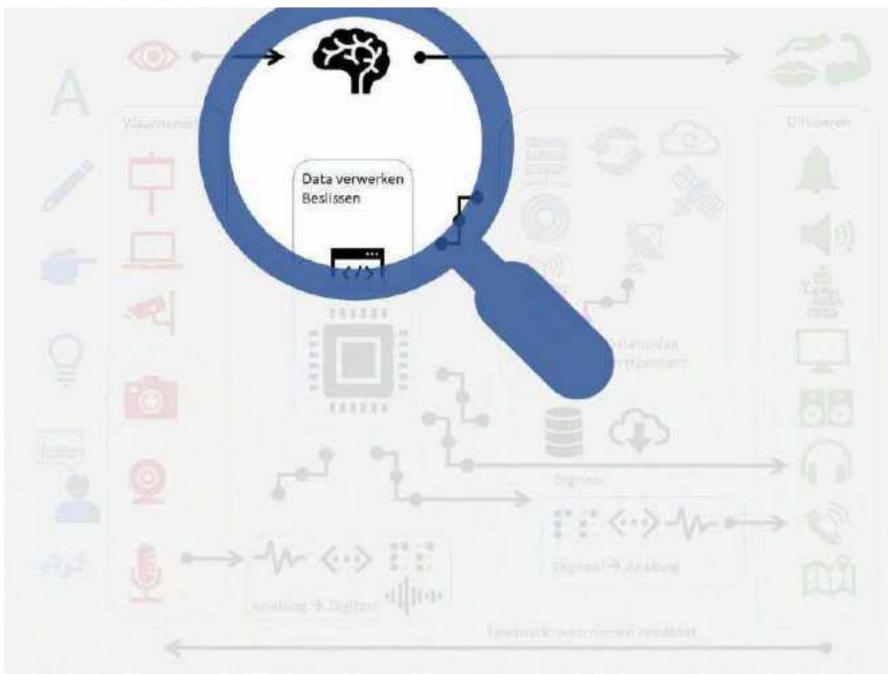
0.4.9 Onderzoeken en maken

Met sensoren kun je allerlei onderzoek doen, zelf bouwen en gebruiken. Onderstaand een aantal suggesties. Kies er één uit:

- Een windmeter maken is interessant en goed te doen, bijvoorbeeld zoals op <https://www.microsoft.com/en-us/education/education-workshop/anemometer.aspx>
- Onderzoek de eigenschappen van één of meerdere sensoren uit het sporthorloge (wat kan de sensor meten, welke gevoeligheid en met welke nauwkeurigheid?)

3. Haal een digitale koortsthermometer uit elkaar en onderzoek de eigenschappen van de temperatuursensor. Andere kant: kun je de koortsthermometer laten werken met een andere, zelf gemaakte, sensor?
4. Meten van vochtigheid: geleidbaarheid. Gebruik hiervoor de standaard beschikbare vochtigheidsensoren bij Arduino. Kalibreer de sensor voor toepassing in een relevante situatie. Bijvoorbeeld: een tuinder laten weten op welk moment zijn planten water nodig hebben of een waarschuwing dat een ruimte geventileerd moet worden omdat de luchtvochtigheid te hoog is.
5. Windsnelheid: gebruik een rotatiesensor
6. Meten van waterstroming (rotatie, verwarming-temperatuurverschil).
7. Indirecte meting (proxy) van grootheden die niet rechtstreeks meetbaar zijn, bijvoorbeeld met camera's, weerkaatsing van geluid, versnellingssensoren.
8. Maak zelf een vochtigheidssensor met koperdraad, gips en een houtblokje. Meet de weerstand in het koperdraad. Breng water aan op het gips en maak een ijkreeks. Wat betekent de waarde die de sensor oplevert? Hoe kan de gemeten waarde gebruikt worden in de praktijk?
9. De nlt-module Sportprestaties beschrijft het maken van een draaihoek-sensor. Hiervoor wordt een potmeter (draaiweerstand) gebruikt, die aangesloten wordt op de computer via Coach. Op welke manier is deze sensor te gebruiken in combinatie met Arduino of een ander microcontrollerplatform?
10. Een eenvoudige versie van de papieren vochtsensor van de Q-strip kan zelf gemaakt en getest worden. Het eerste gedeelte van de opdracht *Q-strip* (hoofdstuk 7) beschrijft deze werkwijze. Onderzoek de relatie tussen hoeveelheid vocht in de sensor en de waarde die met de Arduino gemeten wordt. Maak daarvan een ijkreeks.
11. Apple heeft een beker ontworpen die bijhoudt hoeveel je drinkt. Handig voor mensen die zó in beslag genomen worden door de computer dat ze alles om zich heen vergeten. Welke sensoren zouden er in zo'n beker aanwezig zijn en hoe werken ze? Pas je kennis toe in het ontwerp van een klein blokje dat je op een willekeurig drinkflesje kunt plakken en registreert wanneer en hoeveel er uit gedronken wordt.

0.5 Verwerken en beslissen



0.5.1 Leerdoelen

Kennis 4.1 Je kunt in grote lijnen beschrijven hoe een computerchip gemaakt wordt en welke materialen daarvoor gebruikt worden 4.2 Je kunt uitleggen wat ‘programmeren’ van een computerchip inhoudt en welke rol transistoren daarin hebben. 4.3 Je kunt uitleggen dat computers een computertaal hebben die uit enen en nullen bestaat en dat programmeertalen hierin omgezet moet worden door een compiler

0.5.2 Vaardigheden

4.4 Je kunt een programma en bijbehorende drivers voor de Arduino (of vergelijkbare microcontroller) downloaden, bewerken, uploaden en testen. 4.5. Je kunt de belangrijkste commando’s voor de Arduino die je nodig hebt voor deze module gebruiken en uitleggen waar ze voor bedoeld zijn. 4.6 Je kunt uitleggen dat programmeertalen (zoals Matlab, C++, Python) ontworpen zijn voor specifieke toepassingen en dat ze ieder sterke en zwakke punten hebben. 4.7 Je bent in staat bronnen te vinden waarmee je verschillende programmeertalen kunt leren.

Digitale technologie bestaat niet zonder microchips. Na de uitvinding van de transistor is de electronica in een snel tempo kleiner en krachtiger geworden. We geven je in dit gedeelte een idee hoe die chips gemaakt en geprogrammeerd worden. Als je meer wilt weten kun je terecht in bijlage 3 in de moduledatabase.

Digitale apparaten verwerken informatie en beslissen wat er met die informatie moet gebeuren. Dat gebeurt allemaal in chips, waarvan een processor er ook een is. Dat is de hardware. Informatie kan opgeslagen worden, doorgestuurd of naar actuatoren (scherm, printer, luidspreker etc.). De chip kan een vaste bedrading hebben, waardoor telkens dezelfde bewerking wordt gedaan. Een programmeerbare chip kan telkens andere taken of bewerkingen uitvoeren. De instructies hiervoor zijn gecodeerd in de software. We kijken hoe die chips gemaakt worden.

4.1 Fabricage van electronica

Computerchips worden met tientallen tot honderden tegelijk gemaakt op ronde schijven zuivere silicium van 1 mm dik (wafers). Dat gebeurt in superschone ruimten (cleanrooms). Op de siliciumwafer worden tientallen superdunne laagjes aangebracht: metaal, siliciumoxide en mengsels van silicium en andere elementen. Dit is de Metaal, Oxide, Silicium (MOS) techniek, waarmee het overgrote deel van de computerchips en sensoren gemaakt worden. De siliciumwafer is aan het begin 1 mm dik. Ieder

laagje is zo'n 30 nm dik. De wafer is na al die laagjes maar 2000 nm dikker, dus 1,002 mm. Een computerchip is meestal niet groter dan 1×1 cm, dus er kunnen tientallen tot honderden chips op een enkele wafer gemaakt worden. De wafer wordt daarna in kleine stukjes ('chips') gezaagd. Ze worden dan bevestigd in houders met metalen pinnetjes om ze op een printplaat te kunnen aansluiten.

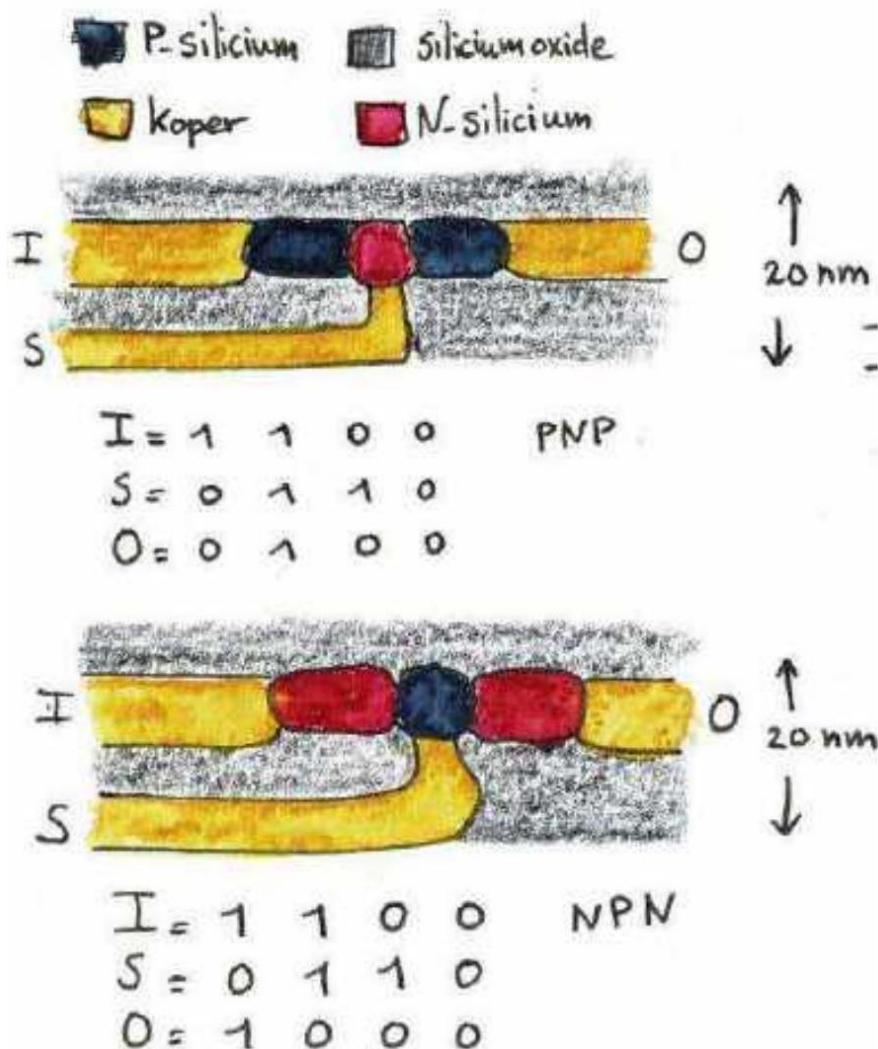
0.5.3 Onderdeeltjes op nanoschaal



Figuur 4.1 Een productieruimte voor computerchips (clean room)

De patronen in laagjes van de chip vormen transistoren, verbindingstraadjes en diodes. De lijntjes en blokjes van metaal, oxide en silicium hebben een breedte en dikte van enkele tot tientallen nanometers (nm , een miljoenste millimeter). Stel dat je met verf heel precies een lijn wilt schilderen, dan doe je afplakband (maskeertape) langs de randen. Op een wafer is het allemaal zo klein, dat daarvoor een speciale techniek gebruikt wordt: lithografie. De wafer krijgt een dun laagje UVgevoelige lak. Wanneer deze lak droog is, kan die weer zacht gemaakt worden met UV-licht. Om het UV-licht precies op de goede plek te krijgen gebruikt men een 'masker'. Een masker is een doorschijnende zwart-wit afbeelding met de verbindingen en onderdeeltjes voor de chip. De afbeelding van het masker wordt met UV op een stukje van 1×1 cm op de lak geprojecteerd. Dat gebeurt voor alle chips op een wafer achter elkaar in een chipmachine. Elk laagje op de chip heeft een eigen masker, dus de machine moet een hele serie maskers na elkaar op dezelfde wafer projecteren.

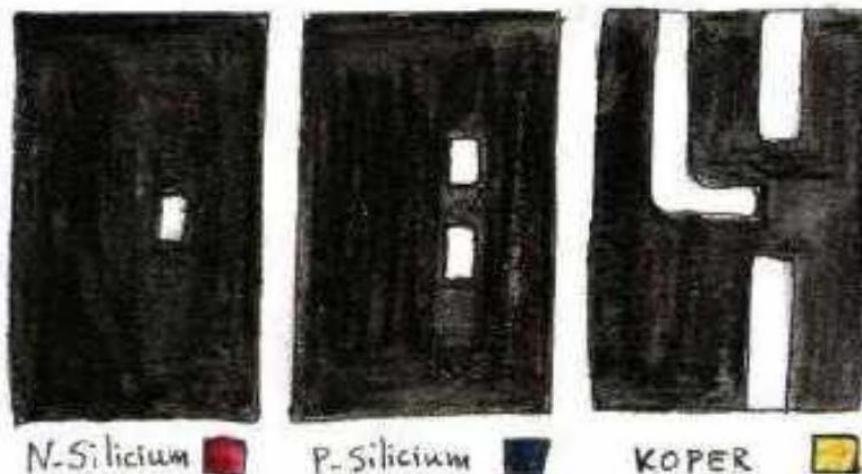
Transistor op chip



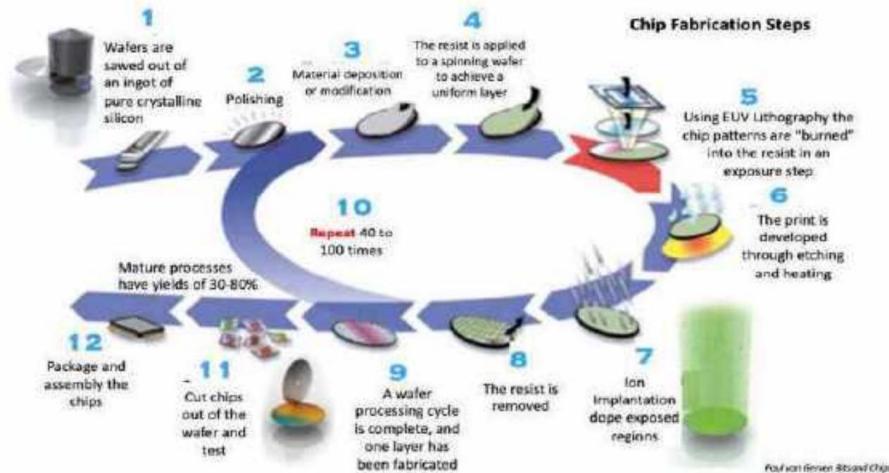
Figuur 4.2 Laagjes metaal, silicium (met n of p toevoegingen) en de verschillende maskers die voor elke laag nodig zijn om een transistor te maken. De structuren zijn enkele nanometers breed.

Realiseer je goed: er zitten tientallen chips op een wafer en elke wafer krijgt tientallen laagjes die tot op de nanometer precies op elkaar moeten zitten. Een wafer moet dan heen en weer geschoven worden tijdens het belichten, naar andere machines voor spoelen, opdampen van metaal en lakken en dan weer terug voor de volgende belichting en je wilt graag heel veel chips achter elkaar kunnen maken.

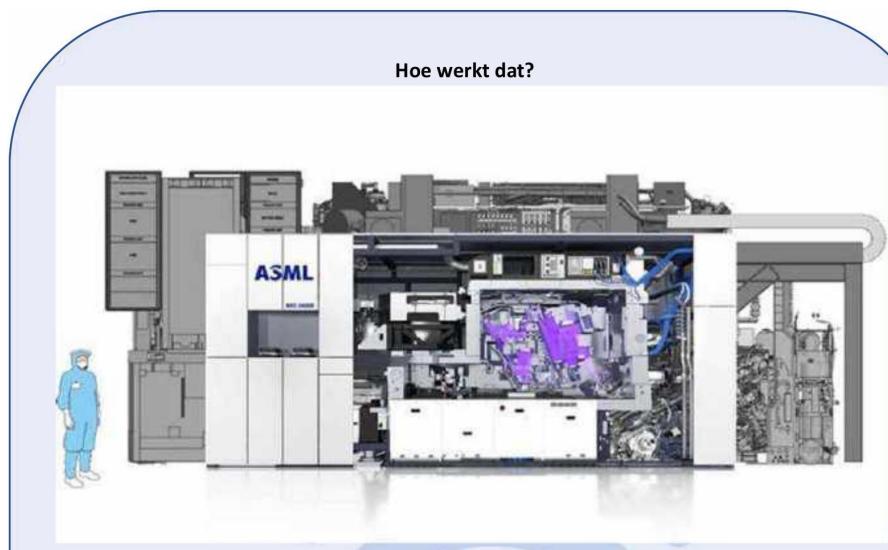
De machines die dat snel en goed kunnen zijn extreem ingewikkeld en kostbaar. Het Nederlandse bedrijf ASML uit Veldhoven loopt voorop als het gaat



Figuur 4.3 Schema van klein stukje belichtingsmaskers in chipmachine voor onderdelen van één transistor



Figuur 4.4 Productiestappen voor een wafer met microchips om het maken van deze chipmachines. Ongeveer driekwart van alle chips wordt gemaakt met de machines van ASML (zie kader). Ze staan vooral in fabrieken in Taiwan, Zuid-Korea en de Verenigde Staten. Daar wordt het overgrote deel van de microchips gemaakt. Inbouwen van deze microchips in printplaten gebeurt op veel meer plekken



in de wereld.

Voor het belichten van chips zijn uiterst precieze Lithografiemachines nodig. Dat zijn zeer dure en grote apparaten, die door 80% van alle chipfabrikanten gekocht worden bij het Nederlandse bedrijf ASML. Een masker met een afmeting van zo'n 30×30 cm wordt verkleind afgebeeld op de lak van de wafer. Voor het belichten wordt UV licht gebruikt. Hoe korter de golflengte van het licht, hoe kleiner de afbeeldingen kunnen worden. De nieuwste machines die ASML maakt gebruiken EUV

(Extended Ultraviolet, met een golflengte van 13,5 nm). De ‘lamp’ bestaat uit een krachtige laser die minuscule druppeltjes tin verhit die zo EUV gaan uitstralen. Daarmee kunnen lijntjes geprojecteerd worden die enkele nm uit elkaar liggen. Zo kunnen superkleine onderdelen op de chip worden gebouwd. Hoe kleiner de onderdelen, hoe sneller de chips kunnen schakelen. Elektronen hoeven minder ver te reizen, zodat energieverlies en warmteproductie minder zijn. Er kunnen méér bouwsteenjes op de chip gezet worden, zodat die meer bewerkingen kan uitvoeren en ook meer informatie kan opslaan. Volgens de wet van Moore verdubbelt elke 10 jaar de hoeveelheid transistoren op een chip. Met de nieuwste productietechnieken lukt het om die lijn vast te houden. Maar er is een grens aan: de afmeting van een atoom kun je niet kleiner maken. Kijk hier hoe een chipmachine van ASML werkt (link, url).

Buurten bij ASML - Wat is een chip? Hoe werkt een chipmachine?
(<https://youtu.be/5YJHwgoE3pE>)

How it's made Microchips ASML (<https://youtu.be/C4gYf-eaZTE>) EUV: Lasers, plasma, and the sci-fi tech that will make chips faster | Upscaled (<https://youtu.be/oliqVrKDtlc>)

De chipmachine moet afbeeldingen kunnen maken waarbij lijntjes enkele nanometers uit elkaar liggen. Op die manier lukt het om de schakelingen op de chip op een fractie van een vierkante millimeter bij elkaar te persen. Met de nieuwste E-UV chipmachines kunnen draadjes en transistoren van enkele nanometers dikte gemaakt worden, waardoor er miljoenen bouwsteenjes op een vierkante millimeter passen. Klein is vooral belangrijk voor de snelheid: de elektronen in de chip leggen dan korte afstandjes af. Méér onderdelen betekent dat de chip meer bewerkingen tegelijk kan doen en meer geheugen heeft.

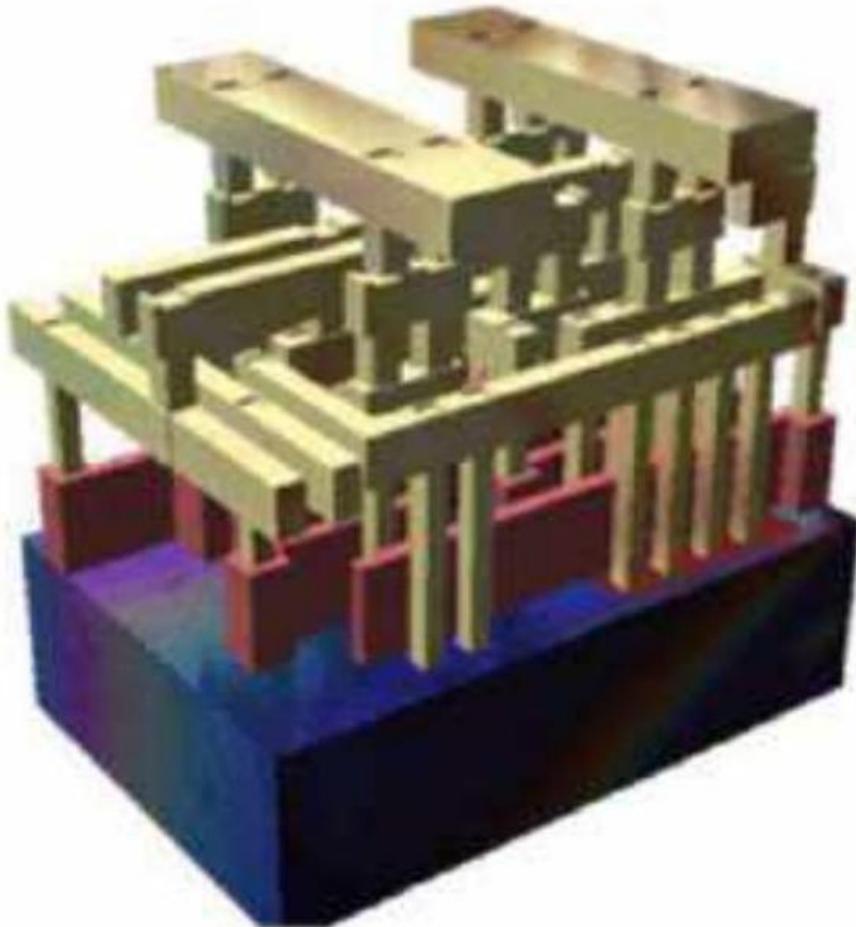
In de bijlagen in de moduledatabase is veel meer te lezen over technieken die bij de productie van chips gebruikt worden en de techniek achter transistoren en logische schakelingen.

0.5.4 Bouwsteenjes op de chip

De ontwikkeling van de miniatuurversie van de transistor (elektronen-schakelaar, uitgevonden rond 1950) zoals die in je telefoon of computer zit laat zien dat het steeds sneller en kleiner kan. In een moderne microprocessors zitten miljoenen transistoren. MOS-techniek zorgt dat die op grote schaal en goedkoop te maken zijn. Met deze techniek worden ook diodes gemaakt, die zorgen dat elektronen maar in één richting kunnen lopen. Een bekende diode is de LED (Light Emitting Diode). Dat de elektronen hier maar in één richting doorheen kunnen kun je uitproberen. De LED geeft alleen licht als hij in een stroomkring in de juiste richting in aangesloten: met het lange pootje op de pluspool. Ook de condensator, die elektronen kan opslaan en afgeven, is een belangrijke bouwsteen op de chip.

4.2 Programmeren van een chip

Chip-ontwerpers streven er naar om v  el componenten bij elkaar te



Figuur 4.5 3D-structuur van de laagjes op een chip. De geleidende pilaartjes van metaal met alle componenten van de chip worden laag voor laag opgebouwd, tot een dikte van enkele micrometers. De blauwe onderkant is de silicium wafer. pakken op   en chip. Via contactpuntjes aan de buitenkant van de chip staat die in verbinding met de pinnetjes van de behuizing. Zo staat de chip in verbinding met de andere componenten van het apparaat en krijgt de chip stroom. Hoe ingewikkelder de chip, hoe meer contactpinnetjes er zijn. Door de stroom op bepaalde pinnetjes aan en andere uit te zetten, kunnen de transistoren in een bepaalde stand gezet worden. Chips kunnen dat schakelen alleen doen als er spanning op de voedingspin staat.

Er bestaan twee belangrijke typen chips. Logische chips kunnen informatie verwerken en geprogrammeerd worden, zoals de centrale processor (CPU) van een computer. Bepaalde software is vast in het geheugen van de chip opgeslagen (firmware, embedded software) zodat die snel beschikbaar is. Geheugenchips houden informatie vast. Het snelle DRAM geheugen doet dat alleen als er spanning op de chip staat. Het trage NAND geheugen, zoals een flashkaartje, houdt informatie ook vast als er geen spanning op de chip staat.

Een chip verwerkt opdrachten in een vaste regelmaat: de kloksnelheid. Op iedere tik van de klok kan de computer   en actie uitvoeren. In een computerchip zit een kristal dat met een bepaalde frequentie trilt (in de orde van MHz of GHz) en daarmee bepaalt de chip de tijdsduur en dus zijn kloksnelheid. Die kan bijvoorbeeld verlaagd worden om stroom te besparen. Hoe hoger de frequentie, hoe meer verwerkingsstappen een chip per seconde aan kan.

Hoe moet je het programmeren van een chip nu voorstellen? Programmeren is vergelijkbaar met omzetten van schakelaars. Dat gebeurt bijvoorbeeld in de cockpit van een vliegtuig. De piloten zetten handmatig schakelaars in een bepaalde stand voor vertrek, tijdens de vlucht en na de landing. In het vliegtuig gaan dan allerlei systemen aan of uit. Hoe de schakelaars moeten staan, hebben ze in een

checklist staan. Je kunt dat vergelijken met het programma, de software.

Ook het programmeren van een chip is weinig anders dan het omzetten van schakelaars. De software programmeert de chip en zet ook de schakelaars om. Dat zijn in een chip de transistoren. Een pinnetje ‘aan’ schakelen doe je door er spanning op te zetten (bijvoorbeeld 3 V). ‘Uit’ is dan 0 V. Dat is in



digitale waarden: 1 en 0.

Figuur 4.6 Het schakelbord in een cockpit vergeleken met de pinnen van een ATMEGA 32P microcontroller. Een programma zorgt er voor dat een chip taken kan uitvoeren. De afzonderlijke stappen in een programma worden algoritmen genoemd. Als het beginpunt en eindpunt gelijk is, hoeven de tussenliggende stappen niet hetzelfde te zijn. Ofwel, in veel gevallen zijn er meerdere algoritmen te bedenken met dezelfde uitkomst. Een slimme volgorde van handelingen kan een sneller resultaat opleveren. Dat betekent dat het uit maakt hoe je het algoritme schrijft en welke algoritmen je voor het programma gebruikt. Dat is de kunst van het elegant programmeren. Terug naar de cockpit: het is wel handig om de knoppen niet kriskras door elkaar te bedienen, dat kost nodeloos extra tijd. Daarom is bij een goede checklist de volgorde zo efficiënt mogelijk.

Abstract Op het niveau van de chip moet het algoritme vertellen wat de chip moet doen met de spanning op ieder van de pinnetjes. Maar programmeren door per pinnetje op te schrijven wanneer die 1 of 0 moet zijn, is voor een mens niet te doen. De reeksen 1 en 0 die de chip kan verwerken is machinetaal (machine code). Dat is voor mensen niet te volgen. Vandaar dat er programmeertalen zijn verzonnen die makkelijker te begrijpen zijn en eigenlijk een hele serie handelingen of algoritmen met één commando kunnen geven (assembler code). De computer rekent dit zelf om naar de algoritmen in machinetaal. Dat gebeurt door de compiler. De hogere orde programmeertalen zijn voor mensen beter begrijpelijk, maar staan verder af van de machinetaal.

We kunnen niet van buitenaf zien hoe de schakelaartjes in de computerchip staan. Daardoor kunnen we ons ook moeilijk voorstellen wat een programma met een chip doet en zijn we afhankelijk van informatie die we uit de chip terugkrijgen en vertalen naar iets begrijpelijs.

Er zijn tientallen programmeertalen ontwikkeld, die voor andere toepassingen bedoeld zijn. De Arduino-taal, Python of C++ zijn voorbeelden. Bepaalde talen zijn geschikt om apparaten aan te sturen, andere om plaatjes te bewerken of lange rijen getallen te bewerken.

Als een computerprogrammeur een programma maakt dat werkt, maar dat een ander niet kan begrijpen, is dat niet handig. Slordigheid kost heel veel rekentijd, fouten opsporen en verlies van kennis. Daarom is het plezierig om gebruik te kunnen maken van stukken code die goed werken, maar die je niet zelf hoeft te verzinnen. Er zijn allerlei databases met computercode beschikbaar. Programmeurs halen daar blokken code uit waarmee ze snel een werkend programma kunnen maken. Je hoeft van de code uit een blok niet precies te weten hoe die in elkaar zit. Als die code ook nog eens als een (gekleurd) blokje wordt weergegeven, zie je echt ‘blokken-taal’ zoals je die bij veel populaire systemen tegenkomt.

Abstract

0.5.5 Programmeren van de microcontroller

Als je met een microcontroller gaat werken, zoals Micro:bit of Arduino, dan programmeer je in Arduino-taal. Jouw Arduino-programma wordt door de software in je computer (de compiler) omgezet naar machinetaal voor de Arduino en naar het bordje gestuurd. Daarmee gaat het bordje de taak uitvoeren. Zolang er geen nieuw programma komt blijft het bordje dezelfde taak uitvoeren. Als je het geheugen wist (dat kan met de RESET knop), is het programma verdwenen en doet het bordje niets meer. Opnieuw moet je vanuit je computer het programma in machinetaal kopiëren om het bordje te laten werken.

Over computerchips en programma's is veel meer te ontdekken. Bijvoorbeeld: de architectuur van processorchips met 4, 8, 16, tot 64 bit of hoger, parallele of seriële processen, de opbouw van programmeertalen, hoe kunstmatige intelligentie via neurale netwerken geprogrammeerd wordt, opbouw van schakelingen, het bewerken van beelden of de manier waarop een sporthorloge zijn verzamelde data verwerkt en doorstuurt. Hiervoor kun je terecht in verdiepende (vervolg)modules.

4.3 Artificial Intelligence (AI)

Bij kunstmatige intelligentie (Artificial Intelligence, AI) maakt de software zelf allerlei instellingen die we niet kunnen zien. Dat gebeurt in programma's die kunstmatige neurale netwerken (Artificial Neural Networks, ANN) genoemd worden. Hierdoor kan de software leren door voortdurend patronen te vergelijken. De software kan zo bijvoorbeeld beelden herkennen, zoals bij de gezichtsherkenning van een fotocamera of het zoeken naar bepaalde vormen in afbeeldingen. Doordat de computer zelf de algoritmen aanpast, is het ook niet meer te achterhalen hoe die precies in elkaar zitten. Daarom verzinnen wetenschappers methoden om de algoritmen na te bouwen en te vergelijken met wat AI heeft gedaan. Hoe AI werkt, kun je in een vervolgmodule ontdekken (zie H8).

0.5.6 Kunstmatige Intelligentie: heel geschikt voor het leren herkennen van beelden.

Zoek Wally! Een bekende tekenaar van plaatjesboeken voor kinderen tekende in iedere afbeelding van een boek het figuurtje Wally. Voor de lezers was het de uitdaging om die te ontdekken. We zijn als mensen visueel ingesteld, en heel behendig in het analyseren van beelden, maar het herkennen van wat we zien vraagt oefening. Zeker als het ingewikkelde en niet-alledaagse beelden zijn. In een foto herkennen we snel dat er menselijke gezichten op staan. Maar het is een stuk lastiger om elke afgebeelde persoon te herkennen, daarvoor kennen we te weinig mensen. Computers die gezichten kunnen herkennen hebben de mogelijkheid om in grote databases te zoeken en razendsnel vergelijkingen te maken tussen het afgebeelde gezicht en de afbeeldingen in de database. Door de verbeterde rekenkracht



Figuur 4.7 Deze lijkt op Wally gezichtsherkenning steeds meer toegepast. Beveiligingscamera's, toegangspoortjes, toegang tot je telefoon: Al met gezichtsherkenning is wijdverspreid.

Wat met gezichten kan, is ook mogelijk met alle andere patronen. Herkennen van planten, vogels, verkeersborden, de weg, obstakels, kwaadaardige cellen in een orgaan, pakketjes in een sorteercentrum of de boodschappen in een mandje: voor de computer is het allemaal even ingewikkeld. Voor ons is dat heel nuttig: in de zelfrijdende auto, een app die je vertelt welke vogel je ziet, robots die automatisch je boodschappen of pakketjes bezorgen. In medische centra wordt veel onderzoek gedaan naar het herkennen van afwijkingen in medische beelden. Is dit plekje een onschuldig bultje of een kwaadaardig gezwel? Pathologen moeten die Vraag elke dag tientallen keren beantwoorden als ze stukjes weefsel moeten onderzoeken tijdens operaties. Kan de computer artsen helpen om antwoord te geven op deze Vraag? Zal hen dat helpen om betere diagnoses te stellen?

0.5.7 Kunstmatige Intelligentie om tumoren te herkennen: het onderzoek van Daan Geijs

Een jaar of tien geleden zat Daan op de middelbare school. Hij had een natuurprofiel, maar volgde niet de juiste vakken om de studie te doen die hij graag wilde: biomedische technologie. Daarom is hij wiskunde B en natuurkunde gaan bijspijkeren om toch in Twente te gaan studeren. Inmiddels afgestudeerd als biomedisch ingenieur én met een lerarenopleiding natuurkunde, is hij aan de slag gegaan als promovendus bij de medische faculteit van de Radboud Universiteit. Hij werkt aan een promotieonderzoek om kunstmatige intelligentie (patroonherkenning door computers) in te zetten bij het opsporen van tumoren. Preciezer gezegd: om een systeem te ontwikkelen dat een patholoog ondersteunt bij het bekijken van weefselplakjes om te ontdekken of er kankercellen aanwezig zijn. Bedenk: van ieder verdacht plekje bij elke operatie moet een patholoog een stukje weefsel in plakjes snijden en beoordelen op de aanwezigheid van kankercellen. Dat is veel werk én het Vraagt de deskundigheid van een hoog opgeleide specialist. Kan je een computer leren om in foto's van die plakjes weefsel de kankercellen er uit te pikken? Kan een computer ingezet worden om de patholoog het vermoeiende routinewerk uit



handen te nemen?

Figuur 4.8 Daan Geijs vertelt over de toepassing van digitale technologie in het ziekenhuis. Daan heeft een aantal filmpjes opgenomen waarin hij vertelt over zijn opleiding, wat hij nu onderzoekt, wat kunstmatige intelligentie is en waar hij na zijn promotie zou kunnen gaan werken. Deze zijn terug te vinden op de leerlingsite (https://maken.wikiwijs.nl/171772/Schakelmodule_Digitale_Technologie_online_materiaal)

4.4 Maken en programmeren: digitale thermometer met Arduino

In deze opdracht ga je zelf ervaring opdoen met bouwen en programmeren van een Arduino (of vergelijkbaar device): een digitale (koorts)thermometer. Deze meet de temperatuur en toont de temperatuurwaarde op een display. Misschien denk je: "wat heeft dat voor zin? Ik kan voor weinig geld ook zo'n ding kopen". Klopt, maar een thermometer is maar een voorbeeld van wat je allemaal met een Arduino kunt maken en doen. Misschien krijg je zelf een heel goed idee en wil je dat gaan maken!

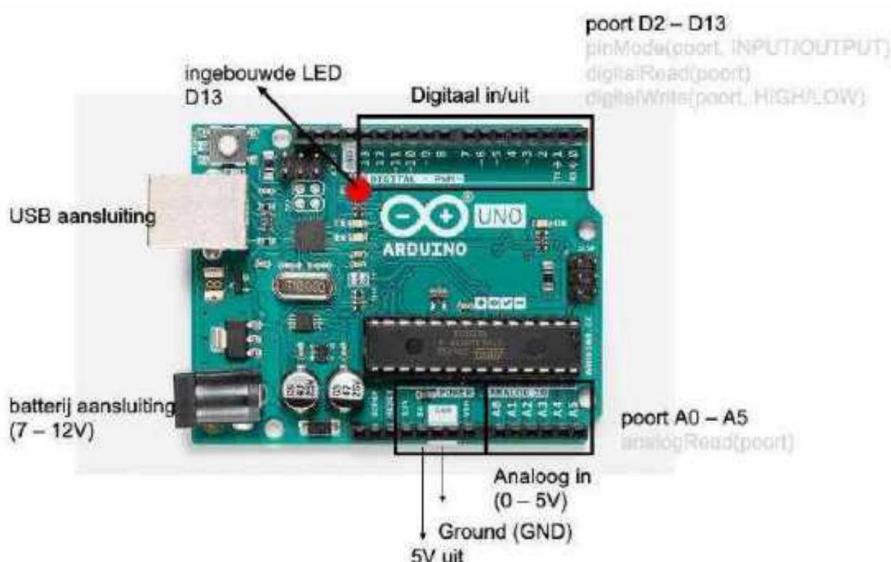
Het is dan handig om te weten hoe je dat aanpakt. Daarom hieronder eerst een paar stappen die je in ieder project zet. De volledige stap-voor-stap opdracht is in een bijlage beschreven die verkrijgbaar is in de moduledatabase.

0.5.8 Stap 1: Kiezen van de hardware

Voor Arduino zijn veel verschillende onderdelen beschikbaar. Als je ze wilt laten doen wat je wilt, moet je de juiste onderdelen kiezen voor je project. Voor de digitale koortsthermometer wil je temperatuur meten, verwerken en laten zien. Je hebt in elk geval een temperatuursensor, schakelaar, Arduinobordje met voeding, schermpje (display) en LED nodig.

Wanneer je met andere spullen aan de slag wilt (zoals de Raspberry Pi, Micro:Bit, M-block of nog andere) moet je ook hier de juiste selectie maken voor je project. Vaak zijn sensoren en andere onderdelen door elkaar te gebruiken, mits de aansluitingen passen. Losse steekpennetjes zijn vaak vervangen door connectoren om onderdelen foutloos aan te sluiten.

0.5.9 De Arduino Uno



Figuur 4.9 Het bordje van de Arduino Uno

In figuur 4.9 zie je een afbeelding van een Arduino Uno. Er zijn meerdere soorten Arduino's, maar dit is de belangrijkste. Misschien heb je een kloon die er iets anders uitziet, maar dat maakt niet uit, de aansluitingen zijn hetzelfde. Je moet de Arduino zien als een klein computertje. Op het printplaatje zit een processor (net zoals in je telefoon) voor alle taken. Aan die Arduino sluit je sensoren, display, lampjes etc. aan. Dat doe je op de zwarte insteekpennetjes.

De aansluitingen:

1. USB aansluiting: Hiermee koppel je de Arduino aan de computer
2. Batterij aansluiting: De Arduino kan niet werken zonder spanning. Als de USB is aangesloten, dan krijgt de Arduino daarvan de spanning. Wil je zonder USB werken, dan kun je een batterij/accu/voedingskast aansluiten tussen 7 - 12V. Vaak hebben de sensoren ook een spanning nodig. De 5V aansluiting levert die spanning (5V dus). De GND (= Ground) is de min-kant. Het maakt niet uit of de batterij of de voeding uit USB gebruikt wordt.
3. (rechtsboven) digitale in- en uitgangen: Dit zijn in- en uitgangen die een 0 of een 1 kunnen “lezen” (read) of “schrijven” (write). Dat kun je zelf bepalen. Je kunt hiermee een stand van een schakelaar (aan/uit) lezen (read) of een lampje (LED) aan/uit zetten (write). Hoe dat werkt wordt hieronder verder uitgewerkt. In totaal zijn er 11 van dit soort aansluitingen, genummerd D2 t/m D13 (Digitale poorten).

4. (rechtsonder) analoge ingangen, genummerd A0t/m A5 (poorten): Dit zijn 6 bijzondere poorten die een spanning (tussen 0 en 5V) kunnen omzetten in een getal. Deze heten ADomzetters: Analoog (spanning) naar Digitaal (getal) omzetter. Deze moeten we gaan gebruiken om de sensor te gaan lezen en kun je dus ook alleen gebruiken om te “lezen”.
5. De Uno heeft een ingebouwde LED, gekoppeld aan D13.

0.5.10 Stap 2: Opzoeken van de juiste drivers

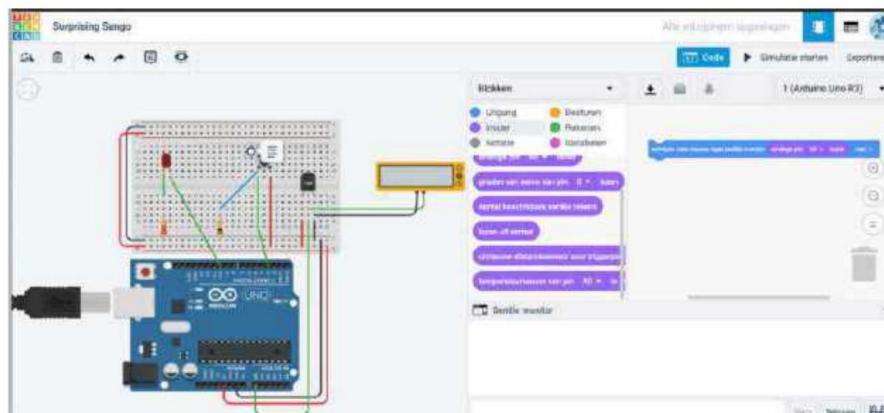
Als je een onderdeel, zoals de temperatuursensor, op een ingang van de Arduino aansluit moet je de Arduino informatie geven. Op welke pinnen is deze aangesloten? Komt er een digitaal of een analoog signaal uit? Wat betekent een waarde die het onderdeel doorgeeft of nodig heeft? Al deze informatie zit verpakt in stukjes code, de drivers. Op internet staan bibliotheken (libraries) met drivers voor al die sensoren en andere bouwstenen. Om die op te zoeken, heb je de naam van een onderdeel nodig, zoals de TMP36 (temperatuursensor). Wanneer je zelf een sensor of actuator bouwt, moet je zelf dit soort informatie bij elkaar zoeken en in een infobestandje zetten voor de Arduino.

0.5.11 Stap 3: Aan de slag, eventueel zónder te bouwen

Om de Arduino te vertellen wat die allemaal moet doen met die in- en uitgangen moet je deze programmeren. Je moet dus eigenlijk een app maken voor de Arduino. Dat doe je op de computer. Als het programma klaar is, stuur je het via de USB naar de Arduino en die gaat vervolgens doen wat je geprogrammeerd hebt. Het programmeren doe je in de Arduino IDE.

Het bouwen en programmeren van een apparaat kan een stuk eenvoudiger als je dit vooraf kunt samenstellen en testen met je computer. Voor Arduino kan dat heel mooi in een online omgeving: TinkerCad (ga naar [tinkercad.com](https://www.tinkercad.com) en maak een account aan).

Je kunt hier met plaatjes je apparaat samenstellen en de code (sketch) schrijven (in blokkentaal, of in de programmeertaal C). Je kunt het programma opslaan, downloaden en in de echte Arduino laden (*.ino bestand). Je kunt ook de Arduino-IDE (Integrated Development Environment) gebruiken om je sketch te maken of een bestaande sketch aan te passen. Als je het fysieke apparaat bouwt en de sketch laadt (via de Arduino IDE), doet deze wat je al uitgeprobeerd hebt. Tenminste: dat is wat je kunt verwachten. Uiteraard moet je testen of dat echt zo is. Daarna kun je de echte metingen gaan



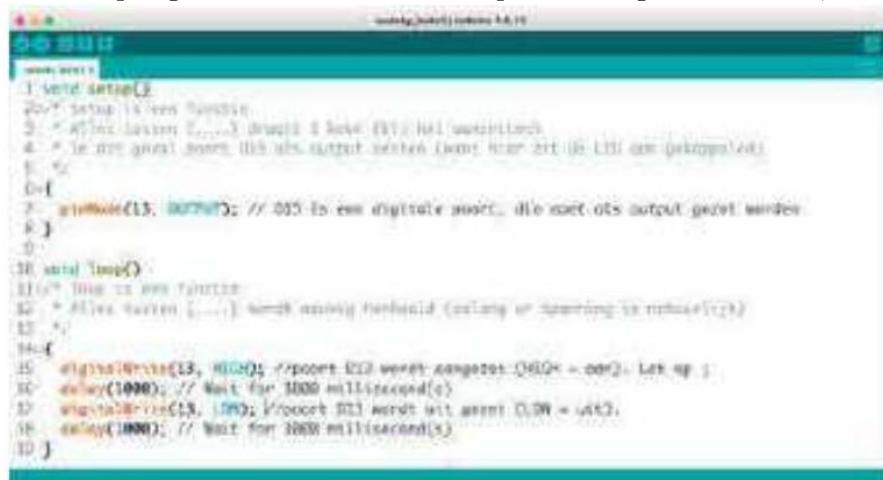
doen.

Figuur 4.10 In TinkerCad kun je virtueel een device bouwen (links de hardware en aansluitingen) en het programma in blokken weergeven (midden). De coderegels kunnen ook via een venster getoond worden. Met een knop bovenaan het scherm kan de simulatie gestart worden.

0.5.12 Stap 4: Kalibreren, testen en verbeteren

Als je een apparaat hebt gemaakt dat metingen doet, is kalibreren een belangrijk onderdeel. Een waarde uit de sensor (tussen 0 en 1024) moet omgerekend worden naar een correcte waarde op het schermpje. In TinkerCad kun je ook het kalibreren van je sensor uitvoeren, zodat je de juiste code kunt schrijven. Uiteraard controleer je die metingen met de fysieke Arduino nog wel.

Wanneer je metingen opslaat of stuurt, is het ook nodig om te weten wat er gemeten is en op welk tijdstip. Daarvoor moet je ook een timer op de Arduino aansluiten. De gegevens kunnen op een SD-kaartje gezet worden of via een USB-kabel of draadloos via Bluetooth, WiFi of LoraWan naar een ander apparaat gestuurd. Opslag van data komt in deze opdracht nog niet aan bod, maar wel in die

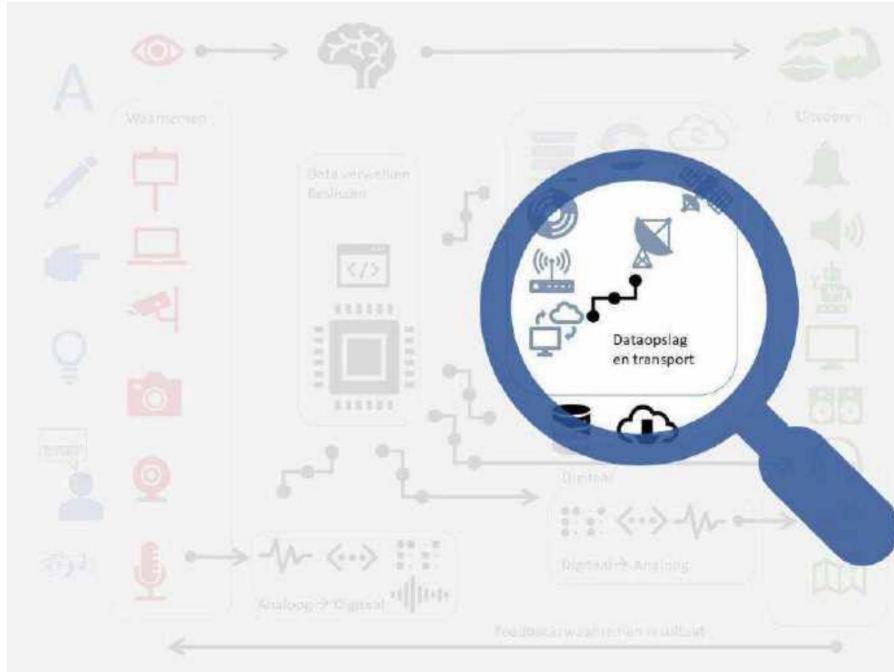


```
1 void setup() {
2   // Setup is een functie
3   // * Aan de beginnen van de functie is de pin 13 (led) uitgezet
4   // * In een apart functie (loop) als output-werken doen voor zowel de LED en de potentiometer
5   //
6   // D1
7   // 1. afbeelding, ledPin0 // D05 is een digitale poort, die moet als output gezet worden
8   //
9   //
10 void loop() {
11   /* Hier is een functie
12   * die waarde [....] leest en dat terugbrengt (waarde is waarde(12))
13   */
14   //
15   digitalWrite(13, HIGH); //poort D13 wordt aangezet (D09 = oef). Let op !
16   delay(1000); // Wacht voor 1000 millisecond(s)
17   analogWrite(13, 100); //poort D13 wordt uit geven (D09 = oef),
18   delay(1000); // Wacht voor 1000 millisecond(s)
19 }
```

van Q-strip in H7.

Figuur 4.11 Een stukje sketch (code) voor Arduino (in de taal C). Achter de // tekens staat uitleg.

0.6 Data opslag en transport



0.7 Leerdoelen

Kennis 5.1 Je kunt uitleggen dat er vuistregels zijn voor het ordenen van gegevens binnen bestanden, het ordenen van bestanden en dat het gebruiken van die vuistregels je tijd bespaart en de kans op fouten vermindert. 5.2 Je kunt uitleggen dat metadata de informatie bevat over wat er in bestanden te vinden is en hoe deze geordend zijn. 5.3 Je kunt uitleggen dat gegevens fysiek opgeslagen worden op harde schijven (in het klein) of in datacenters (in het groot) en hoe deze opslag functioneert 5.4 Je kunt uitleggen dat gegevens op verschillende manieren beveiligd kunnen worden door wachtwoorden en encryptie en dat de wiskunde hier een grote rol in speelt. 5.5 Je kunt uitleggen dat het internet een combinatie is van ip-adressen en TCP/IP protocollen en hoe dit werkt

0.8 Vaardigheden

5.6 Je hebt gewerkt met een apparaat dat informatie naar een website kan sturen zodat je op afstand mee kunt kijken met de sensoren die jouw digitale systeem bevat.

0.8.1 5.1 De ‘cloud’

Met alleen maar een mobiele telefoon heb je eindeloos veel informatie beschikbaar. Het versturen en ophalen van data gaat snel, draadloos en onzichtbaar via de ‘cloud’. Je deelt bestanden zonder dat je er iets extra’s voor hoeft te doen. De naam ‘cloud’ suggereert dat het ergens in de lucht gebeurt, maar niets is minder waar. Overal in de wereld staan grote en kleine gebouwen zonder ramen, vol met computers en harde schijven: datacenters. Digitale data is heel geschikt om snel en zonder verlies te transporteren. Dat gebeurt met draadloze wifi en 4G/5G signalen en kabels van je mobiel naar datacenters en weer terug. Goede beveiliging is ook vereist. Achter dataopslag en datatransport zit een heleboel techniek die we in dit hoofdstuk beter gaan bekijken.

0.9 Digitale signalen

Digitale technologie gebruikt de binair code die twee (= bi) waarden kan aannemen: 0 of 1. Dat is in een elektrische schakeling hoge (5 V) of lage (0 V) spanning, met licht door helder of zwak of

zwarte en witte lijnen of vlakjes. Informatie wordt gecodeerd in reeksen nullen en enen.

0.9.1 5.1 Vraag ““{exercise}

Met een rijtje van 2 binaire waarden kun je 4 combinaties maken: 00, 01, 10 en 11. Dat is 2^2 {}

Figuur 5.2 Een byte is uit een rijtje van 8 bits. Deze 24 bits zijn in 3 byte verdeeld. 8 bits wordt in discrete (stapgewijze) decimale waarden 0 tot 1024. (Ga eens na hoeveel bits er voor

Omzetting van analoog naar digitaal geeft wel wat verlies (immers: bij een reeks verschillende

Nauwkeurigheid

Precies meten van de temperatuur, bijvoorbeeld met twee cijfers achter de komma, betekent dat digitaliseren is. Maar ook: hoe meer rekenkracht de processor moeten hebben om een signaal te

5.2 Vraag ““{exercise}

““
Hoeveel data produceert een 16 -bit sensor in één dag? Geef je antwoord in Mb.

Data bewerken, opslaan en transporteren

! [] (https://cdn.mathpix.com/cropped/2024_12_20_510ffc175a3910aebf8dg -53.jpg?height=244&width=)

Figuur 5.3 Opslag en uitvoer van digitaal signaal

! [] (https://cdn.mathpix.com/cropped/2024_12_20_510ffc175a3910aebf8dg -53.jpg?height=281&width=)

Figuur 5.4 Verwerking van digitale gegevens in de processor

Het elke seconde uitlezen van de 8 -bit sensor levert 1 byte data per seconde, 300 byte in 5 m

Geheugen

Digitale systemen gebruiken geheugenchips en opslagmedia (harddisk, optische schijven, solid s

5.3 Opdracht

- Test eens uit wat de bandbreedte (bandwidth) is van de internetverbinding van een mobiel en
- Vergelijk de bandbreedtes met elkaar. Verklaar het (eventuele) verschil dat je ziet.

In de computerwereld draait alles om data, datatransport en dataopslag. Aan alleen maar een re ’ja’. Daar heb je weinig aan als je de Vraag ““{exercise}

““ en de context niet kent. Het antwoord wordt anders als je de Vraag ““{exercise}

““ weet: wil je suiker in je thee? Een database bevat tabellen met data én metadata. Daar kun

Een bestand (file) heeft allerlei informatie dat vertelt waar het bestand over gaat en hoe het

De computer schrijft en leest de hele tijd van werkgeheugen naar massageheugen (harde schijf o

Voor het heen -en -weer sturen van datapakketjes moeten computers verbinding met elkaar hebben

5.4 Vraag ““{exercise}

Hoeveel opslag heeft jouw mobiel? Maak onderscheid tussen werkgeheugen (RAM) en het opslag (ssd) geheugen. Hoe zit dat met je computer?

0.9.2 5.2 Verbindingen

0.10 Metaaldraad

Elektriciteit loopt via metaal en dat is dus ook de manier om elektrische digitale signalen te vervoeren. De snelheid waarmee dat gebeurt is pakweg 0,1x de lichtsnelheid. Een elektrisch signaal verzwakt over grotere afstanden, waardoor er versterkers nodig zijn. De standaard voor digitale elektrische signalen is de UTP-kabel, die bijvoorbeeld ook gebruikt wordt om in huis router en computer te koppelen.

5.5 Vraag ‘‘‘{exercise}

Hoe werkt die UTP kabel? Waarom heeft die meerdere draden?

Sneller: glasvezel

Het sneller en over langere afstanden transporteren van digitale informatie gaat tegenwoordig

[^0]Voor de glasvezel worden lasers gebruikt (die een digitaal signaal omzetten in lichtpulsen)
! [] (https://cdn.mathpix.com/cropped/2024_12_20_510ffc175a3910aebf8dg-55.jpg?height=772&width=1000)

Figuur 5.5 Kwaliteit van datatransport met licht in glasvezels

Meer bronnen over glasvezel:

<https://www.deingenieur.nl/artikel/nederlandse-bedrijven-pionieren-met-chips-op-licht>
<https://www.deingenieur.nl/artikel/doorbraak-silicium-zendt-licht-uit>

<https://nl.wikipedia.org/wiki/Glasvezel>

https://en.wikipedia.org/wiki/Laser_diode

Nog sneller? Draadloos.

Wifi, bluetooth, mobiele data (3, 4 en \$5 G\$) of satellietcommunicatie werken met radiogolven

5.7 Opdracht

Er is veel te vinden over draadloze communicatie. Als je in een groepje werkt: kies per groeps

5.3 Adresseren

Elke computer in een netwerk heeft een eigen adres: een toegewezen IP (\$v 4\$ of v6) en een MAC-adres. Deze adressen moeten over het netwerk vervoerd worden. Andere protocollen (als http://) sturen de datapakketten.

Het eerste knooppunt leest het IP adres en bepaalt naar welk volgend punt het verstuurd wordt.

Internet knooppunten

Het Internet Knooppunt Amsterdam, op de campus van de Universiteit van Amsterdam, was in de beginjaren van de 1990'er jaren een belangrijk knooppunt voor de wereldwijde verbindingen. De basis van het vrij toegankelijke internet is het nauwkeurig zonder selectie doorgeven van de bestemmingen.

5.8 Opdracht
 Vraag ‘‘‘{exercise}

““ aan je docent, TOA of ICT -beheerder of je op school kunt kijken waar de netwerkouters en -switches staan.

5.4 Datacenters

Voor alles wat je online met je mobiel doet zijn andere computers nodig. Ze worden 'servers' g

5.9 Opdracht
 Voor welke apps moet je online zijn? Welke data stuur je dan van je mobi

Rondkijken in een datacenter

Direct naast het rekencentrum van de Universiteit van Amsterdam staat een groot datacenter van cdn.s3.amazonaws.com/virtual - tours/Amsterdam AM4/index.htm.

Snelheid is alles in deze digitale wereld. Voor een webwinkel is het belangrijk dat bestellen

Digitale haven

Nederland is wereldwijd een belangrijke plek voor de digitale wereld. Een centrale verbinding Groningen.

5.10 Vraag ‘‘'{exercise}

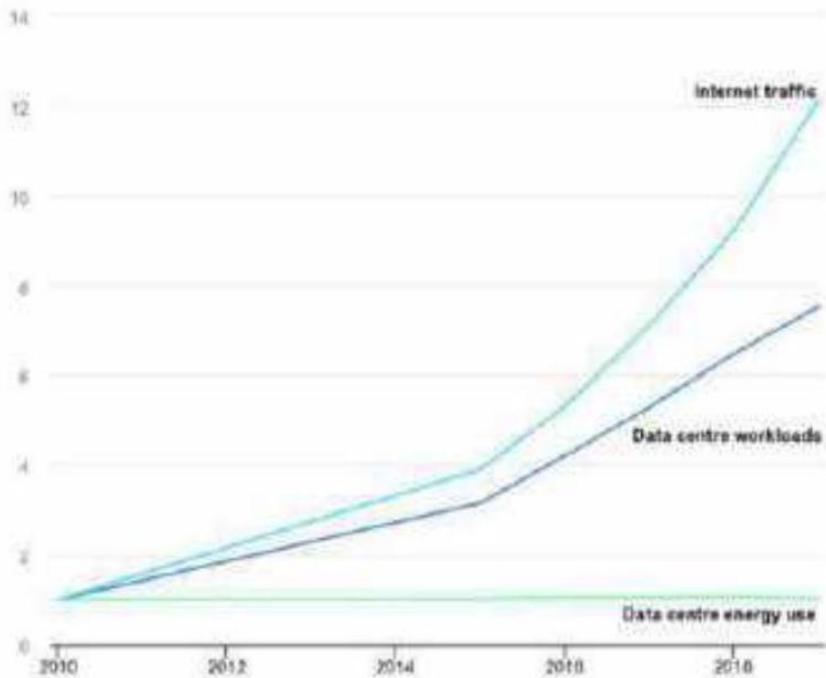
Noem een paar voorbeelden van webwinkels. Denk even terug aan je laatste online bestelling. Welke stappen (denk je) dat er zitten tussen de bestelknop en jouw digitale betaling zit (het moment dat het geld van je rekening af is)?

Veiligheid is ook alles in de digitale wereld. In een datacenter zijn de volgende zaken goed geregeld: constante voeding (stroom voor de servers en schijven, noodstroom als de netstroom uitvalt), koeling (een constante temperatuur rond de 20 graden), snelle verbindingen (glasvezels en korte afstanden tot andere servers) en een streng beveiligd gebouw (mensen kunnen er niet zomaar bij). Een datacenter biedt deze veilige voorziening en bedrijven kunnen daar hun computers neerzetten, door ruimte in het datacenter te huren. Datacenters die ruimte verhuren aan klanten voor het plaatsen van servers, worden ook wel colocatie datacenters genoemd. Weten hoe een datacenter werkt? Zie <https://www.dutchdatacenters.nl/datacenters/hoe-werkt-een-datacenter/>

Sommige techbedrijven hebben zoveel servers dat ze eigen datacenters bouwen, denk aan Google en Microsoft. Dit soort datacenters worden hyperscale datacenters genoemd. Wanneer een programma op een computer in een datacenter draait en de data daar ook staan, dan hoeft je op je eigen computer alleen maar wat instructies naar de computer van het datacenter te sturen. Je eigen computer hoeft dan niets op te slaan of te bewerken. Je kunt op die manier je telefoon gebruiken om allerlei plaatjes te bewerken, omdat het echte rekenwerk in het datacenter gebeurt. Dit werken in de 'cloud' gebeurt niet in de wolken, maar gewoon in een gebouw zonder ramen, ergens op een industrieterrein, een datacenter dus.

0.11 Elektriciteit

Datacenters gebruiken veel stroom. Zoveel zelfs dat ze windmolenvelden in de buurt hebben om groene stroom te leveren. Tegelijkertijd zien we dat het energieverbruik van datacenters al tien jaar stabiel is, terwijl het internetverkeer en de werkbelasting voor de datacenters enorm gestegen zijn. Dit is voor een belangrijk deel te danken aan het samenvoegen van kleine inefficiënte serverruimtes die bedrijven zelf ingericht hadden, tot professionele, grote en efficiëntere datacenters die we nu kennen. Ook het gebruik van meer Cloud-applicaties heeft bijgedragen aan het gelijk blijven van het stroomverbruik. Het restproduct van stroomverbruik in een datacenter is restwarmte. Naast besparing op



Figuur 5.6 Energieverbruik van datacenters, in vergelijking met dataverwerking en internetverkeer. stroomverbruik (bijvoorbeeld door harde schijven te vervangen door solid state drives) wordt ook gekeken of de restwarmte nuttig gebruikt kan worden.

0.11.1 5.11 Opdracht

Duurzaam maken van digitale technologie heeft veel te maken met het verbruik (en de gebruikte bronnen) van energie en grondstoffen en kringlopen. a. Welke manieren kun jij bedenken om digitale technologie zelf duurzamer te maken? Je kunt kijken naar het verminderen van negatieve effecten, zoals hoog stroomverbruik, digitaal afval en opraken van grondstoffen. Hierbij kun je uitgaan van een voorbeeld, zoals een datacenter, je (school)computer, telefoon, minicomputers als Arduino en allerlei printers of digitaal gereedschap. Ook kun je je richten op de samenhang in het hele systeem. b. Op welke manier draagt het gebruik van digitale technologie zelf bij aan duurzame ontwikkeling? Denk bijvoorbeeld aan online samenwerken, slim plannen van routes of apparaten die zichzelf kunnen uitschakelen als ze niet gebruikt worden.

0.11.2 5.5 Datacenters en ‘Cloud’

De standaardopslag in datacenters is vaak de 3,5 inch harde schijf, waarvan er tienduizenden samengepakt zitten in rekken, die samen aangestuurd worden door server-computers. De procesor kan wat meer verwerken dan in je computer thuis, maar de opbouw is hetzelfde. In plaats van één harde schijf heeft een server er tientallen. Ook thuis wordt data vaak apart van de computer opgeslagen op een zogenaamde NAS (Network Attached Storage), een kastje met een minicomputer, harde schijven en een netwerkaansluiting. Met wat handige software kan zo'n kastje ingericht worden om films, documenten en foto's op te slaan. Misschien wel het belangrijkste: door een serie harde schijven samen te voegen kan een array worden gemaakt, waarmee verlies van gegevens voorkomen wordt (in geval een harde schijf kapot gaat). Een systeem als RAID is een handige manier om met minder opslagruimte toch van alle data een kopie te hebben.

0.11.3 5.12 Opdracht

Zoek uit wat RAID is en hoe het werkt. Welke vormen van gegevensbescherming kom je nog meer tegen?

Data redundancy, ofwel voorkomen van dataverlies als er iets stuk gaat, is belangrijk voor veiligheid van data. Minstens zo belangrijk is het voorkomen van diefstal, brand, stroomuitval en andere zaken

waardoor data onbereikbaar wordt. Veel bedrijven zijn volledig afhankelijk van goed werkende servers die constant beschikbaar zijn. Een hokje met computers in een bedrijfspand is niet handig en het risico van uitval of dataverlies door calamiteiten is te groot. Vandaar dat bedrijven hun computers en schrappen het liefst in een veilige en optimale omgeving zetten: datacenters.

0.11.4 5.6 Beveiliging

Het sturen van kleine datapakketjes van en naar aparte computers is probleemloos als iedere computer op het internet netjes doet wat de bedoeling is. Helaas is de wereld niet zo eerlijk. Pakketjes worden onderschept en gelezen, je krijgt pakketjes aangeboden die schadelijk zijn en ga zo maar door. Via internet kan informatie uit een computer gelezen en gewijzigd worden.

We zijn zó afhankelijk van goed werkende computers, die onderling veilig data kunnen uitwisselen, dat beveiliging misschien wel het meest belangrijke onderwerp in de IT-wereld is. Hoe zorg je voor veilige computers?

0.11.5 5.13 Vraag “{exercise}

Poorten

Zoals een gebouw meerdere deuren heeft, geldt dat ook voor een router. Routers en computers be-

Wanneer je pakketjes, voordat ze verstuurd worden, door elkaar schudt (versleutelen, encryptie)

Voor de beveiliging wordt veel kennis van wiskunde gebruikt. Priemgetallen, reeksen willekeuri-

Hacken en cyberaanvallen

Binnendringen van een computer of webserver betekent dat je door de beveiliging moet zien te k

Maar: snelle computers kunnen ook zelf wachtwoorden samenstellen en uitproberen.

Wachtwoorden van meerdere karakters kunnen in heel korte tijd gemaakt en uitgeprobeerd worden.

5.14 Opdracht: Sterke wachtwoorden
 Wachtwoorden bestaan meestal uit de 256 verschillende

- Als een wachtwoord 5 tekens lang is, hoeveel verschillende wachtwoorden kun je dan vormen?
- Stel: een vlotte hack -computer kan 1 wachtwoord per seconde maken en uitproberen. Hoe lang

Via de voordeur binnengaan gebeurt ook vaker dan je zou willen. Een bericht met een bijlage o

5.15 Opdracht

Zoek op internet tenminste twee voorbeelden van hacken, falende beveiliging of dreiging van cy

Routers en beveiliging

Routers en hubs zijn ook kleine computers met chips en software en met algoritmen die door men

“‘d kopieën elders heen te sturen, is iets wat je van een router verwacht. Maar het is niet o

Je kunt dit zien als valspelende postbodes die brieven onderweg kopiëren of verwisselen en pa

In de tech -industrie is het maken van chips en apparatuur op een paar plekken in de wereld ge

Het nieuwe 5G netwerk Vraag “‘{exercise}

“‘t om heel veel nieuwe apparatuur: 5G zenders en snelle netwerkapparatuur om die enorme datas

5.16 Opdracht
 Probeer te achterhalen welke verdenkingen tegen aanbieders van 5G -tech

5.7 Keuzeopdrachten

a. Kleine arduino's maken samen big data

Fijnstof, partikels in de lucht van 1 tot 500 nm , kunnen met een fijnstofsensor gemeten worde

De Arduino die hiervoor gebruikt wordt, meet via de fijnstofsensor en geeft data draadloos doo

b. Je eigen Arduino meetproject.

Arduino is prima geschikt om metingen op te slaan en door te sturen. De thermometer die je in

“‘ gerust aan je docent of TOA!

Voorzie je Arduino -thermometer van een manier om de gemeten data te kunnen opslaan of verstuur

- Opslaan: voorzie je Arduino van een klok en een opslagmedium, zoals SD -kaartje
- Versturen: voorzie je Arduino van een klok en wifi of bluetooth

Zoek de juiste drivers en herschrijf de sketch voor je uitgebreide Arduino thermometer.

Gebruik je device om gedurende een aantal uren temperatuur te meten. Temperatuur in een klaslo

<https://www.arduino.cc/en/Guide/ArduinoUnoWiFiRev2>

<https://www.arduino.cc/en/Reference/WiFiNINA>

[https://store.arduino.cc/explore -iot -kit](https://store.arduino.cc/explore-iot-kit)

c. Satellieten en informatie over de aarde (GIS)

In bijlage 2 in de moduledatabase staat een opdracht over de digitale wereld achter aardobserv

d. Verder te verkennen / te onderzoeken toepassingen

Er is veel aan de orde gekomen in dit hoofdstuk, maar niet zo uitgebreid. Er valt nog veel te ! [] (https://cdn.mathpix.com/cropped/2024_12_20_510fffc175a3910aebf8dg-63.jpg?height=861&width=630)

Figuur 5.7 Landsat satelliet die dagelijks honderden foto's van het aardoppervlak maakt

Maak daarvan een presentatie, informatieve poster of een schriftelijk verslag.

- 5 G communicatie
- Glasvezel en optische communicatie
- Een digitaal geheugen
- Datacenter: hoe is dat ingericht
- Telefonie, videobellen
- Streaming media
- LORA netwerken

6 Uitvoer: de actuator

! [] (https://cdn.mathpix.com/cropped/2024_12_20_510ffc175a3910aebf8dg -64.jpg?height=1171&width=1000)

Leerdoelen

Kennis

- 6.1 Je kunt uitleggen dat digitale systemen invloed kunnen hebben op hun omgeving door middel van actuatoren.
- 6.2 Je kunt uitleggen dat actuatoren zoals lampjes, speakers, motoren, RF zenders, servomotoren en drukknopen zijn.

Vaardigheden

- 6.3 Je hebt ervaren dat je in je programmeercode in je digitale systeem soms bepaalde bibliotheken moet gebruiken om actuatoren te kunnen gebruiken.
- 6.4 Je hebt ervaren dat je voor het aansturen en monteren van actuatoren online veel informatie kan vinden.
- 6.5 Je hebt kennismegemaakt met het aansturen van actuatoren en dat dit gevolgen heeft voor het gebruik van de verschillende soorten.
- 6.6 Je hebt ontdekt hoe goed jouw actuatoren functioneren en hun omgeving beïnvloeden en kunt deze resultaten vergelijken met de verwachtingen.

6.1 Invloed op de omgeving uitoefenen

Aan digitale apparaten hebben we weinig als ze niets in buitenwereld kunnen veranderen. Daar hadden we een belangrijke functie in.

Een digitaal systeem zonder actuatoren kan uiteindelijk niet communiceren of dingen doen. Pixelen en geluid zijn alleen maar informatie.

Digitale actuatoren zijn verwerkt in de machine en worden door de processor(en) aangestuurd. Ze kunnen verschillende soorten bewegingen uitvoeren.

Maar bewegingen en verplaatsingen, zoals zelfrijdende auto's of robotarmen, vragen veel meer energie dan een led.

6.1 Opdracht: Terugkoppeling

Teken een eenvoudig schema voor een zelfrijdende auto of een autonome robot. Benoem enkele sensoren die je kunt gebruiken.

Voorbeelden van actuatoren die in digitale systemen bruikbaar zijn

Fysieke verandering Actuator Toepassing
: - - - : - - - : - - -
Draaibeweging Stappenmotor
Lengteverplaatsing Lineaire motor
Licht geven Diode LCD, OLED: beeldpunt laat licht door of produceert licht
Beeldscherm Lineaire motor of Piezo (zet uit als er elektrische spanning op komt)
Klep bedienen Luidspreker: spoel en magneet Piezo -element
Geluid laten horen

Licht geven of selectief doorlaten

Met actuatoren die licht geven (LED) of blokkeren (vloeibare kristallen) als beeldelement (pixel).

6.2 Opdracht
 Kijk eens met een sterk vergrootglas naar het beeldscherm van je telefoon.

Een beeldscherm kan een matrix zijn van LED's (die zelf licht uitzenden), zoals in een OLED (OLED).

Een digitale projector (beamer) heeft kleine, doorschijnende LCD beeldschermmpjes voor de drie kleuren.

Figuur 6.1 Uitvoeren: de actuator

bioscoop gebruikt tegenwoordig digitale projectoren.

6.3 Opdracht

Zoek eens uit hoe een digitale projector in elkaar zit.

Digitale data wordt gebruikt om LCD pixels te schakelen. Voor zowel LCD als LED pixels wordt d

Er zijn schermen die niet constant ververst worden, namelijk E -ink of electronisch papier. De

De digitale data moet vertellen of een pixel aan staat, in welke kleur en welke helderheid. Om
Daarvoor is veel data nodig. Vooral bewegend beeld Vraag “{exercise}

“t veel bandbreedte, om schermen te
kunnen aansturen. Door de schnellere processoren, glasvezel en betere datacompressie lukt het be

Verplaatsen: motoren

! [] (https://cdn.mathpix.com/cropped/2024_12_20_510ffc175a3910aebf8dg-67.jpg?height=253&width=253)

Figuur 6.2 Digitaal - Analoog conversie

Een elektromotor zet stroom om in een draaibeweging. Het is prettig als je die draaiing onder

De andere manier is het gebruik van een stappenmotor. Een stroompuls zorgt dat de motor één st

Geluid maken

Een korte spanning kan ook weer gebruikt worden om een elektromagneet te activeren. Als je dit

Electromagnetische straling maken

De digitale pulsen kunnen ook gebruikt worden in een radiozender. Een spoel zendt radiogolven

Printen

Inkt of poeder (in de basiskleuren CMYK: cyaan (blauwig), magenta (rozerood), yellow (geel) en

[^1]

6.5 Opdracht

Onderzoek welke data een inkjet - of laserprinter nodig heeft om een afdruk te kunnen maken en
6.6 Opdracht: Andere digitale maakapparaten

Een 3D printer, plotter, CNC frees, digitale snijmachine of een lasersnijder werken technisch
Kies één van deze apparaten uit en zoek verder uit hoe deze (technisch) werkt. Onderzoek ook o

“ om uitleg daarover. Een Fablab of Makerspace heeft zulk digitaal maakgereedschap zeker in

Andere computers als actuator: datasystemen

Wanneer digitale apparaten data aan elkaar doorgeven, kunnen uitgebreide systemen worden gebou

Computers kunnen de gegevens uit de database lezen, en via programma's (algoritmen) nuttige in

Voor de makers

Ook voor actuatoren moet code gebruikt worden om ze goed aan te sturen. Dit soort informatie w

De professional kijkt op

<https://stackoverflow.com> of <https://www.reddit.com>. Als je digitale ontwerpen (zoals 3D teken

6.7 Opdracht: verdieping

Op welke manier worden digitale motoren aangestuurd? 3D -Printbestanden, snijfiles, robotprogr

6.2 Mechatronica in bedrijf

Mechatronica is het gebied waar mechanica (werktuigen) en electronica (digitale technologie) e

Opdracht 6.8: Kennismaken met bedrijven

Over diverse bedrijven zijn lesbrieven gemaakt om je een beeld te geven wat ze (met digitale t

1. ASML: hun chipmachines zijn de werkpaarden waarmee fabrikanten alle chips maken die in alle
2. Vanderlande: wereldwijd worden in luchthavens en sorteercentra de lopende banden van Vander
3. DAF: DAF werkt aan zelfrijdende vrachtwagens. Kunstmatige intelligentie moet helpen om de v
4. Philips: Bij Philips wordt medische apparatuur gemaakt, waaronder MRI scanners. Het herkenn

6.9 Opdracht: Actuatoren in context?

Er zijn talloze andere, aansprekende voorbeelden van de rol van actuatoren bij digitale technolo

- Robotica (Boston Dynamics, zelfrijdende auto, drone, ...)
- Robotarm (voor operaties, inbrengen hersenimplantaat, lassen of montage)
- Het knuffelrobotje voor dementerenden
- Beeldschermen die zich aanpassen aan de lichtomstandigheden en de gebruiker
- Robotbeestje om gedrag op video vast te leggen (Pinguinrobot met camera) Planetearth.
- Rat met actuator in brein - aardbevingsgebieden doorzoeken
- Microscopen en 3D beeldvorming

Maakopdrachten op verschillende niveau's. Met een serie maakopdrachten kan telkens een nieuw a

6.3 Maakopdracht met Micro:bit, Arduino of Raspberry Pi.

6.10 Opdracht

In bijlage 3 in de moduledatabase wordt de NeoPixel beschreven. Ga daarmee aan de slag, en exp! [] (https://cdn.mathpix.com/cropped/2024_12_20_510ffc175a3910aebf8dg-70.jpg?height=189&width=1000)

Programmeer vervolgens de Arduino (of een ander device), zodat een serie NeoPixels gebruikt ku

6.11 Keuzeopdracht

Er zijn talloze actuatoren en meer dan genoeg interessante systemen om te onderzoeken of te bo

““ je docent / TOA welke materialen er op school beschikbaar zijn:

Beeldscherm
CNC machine
Lasersnijder
Plotter / snijplotter
3D printer
Voedselprinter
Inkjet
Poederstraal printer
Thermische printer
Holografische projector
Beamer
Robotproject Leaphy
Karretje
Motoren

Zelfrijdende auto
Robotarm
Kleppen
Schakelaars (bijvoorbeeld een schakelaar om voor 220 V apparaten met Arduino te bedienen)
Beweging en licht
Temperatuur verandering (verwarmer)
Menger
Drukvat, compressor (aansturen zware machines)
Stembediening voor je lampen, gordijnen, koffiemachine.

0.12 Eindopdracht: Digitaal device ontwerpen

0.13 Leerdoelen

7.1 Je leert de ontwerpcyclus beheersen en toepassen om een werkend prototype te kunnen bouwen dat een probleem oplost. 7.2 Je krijgt inzicht hoe innovaties in de tijd veranderen 7.3 Je maakt kennis met opdrachtgevers en realistische opdrachten uit het bedrijfsleven

0.14 Projecten

0.14.1 7.1 Digitale technologie in huis: weerstation voor het binnenklimaat.

In huis wil je een veilig en gezond binnenklimaat hebben. Fijne temperatuur, geen geurtjes, tocht of vocht, en de waarden voor CO₂ en CO binnen de normen. Deze parameters zijn goed meetbaar met digitale sensoren. Met je kennis van digitale microcontrollers kun je een gecombineerd device bouwen en programmeren, dat deze waarden voor je in de gaten houdt. Maak gebruik van de standaardsensoren voor vocht, licht en temperatuur. Fijnstof (rook), CO en CO₂ vragen extra sensoren. Zoek op internet wat er verkrijgbaar is of Vraag je docent/TOA naar de beschikbaarheid op school.

0.15 Opdracht:

Maak met digitale elektronica een weerstation voor in huis. Meet een combinatie van licht, temperatuur, vocht, fijnstof en tocht (tenminste drie van deze waarden). Programmeer het apparaat zo dat de afwijking ten opzichte van de comfortwaarde (optimale waarde) wordt bepaald. a. Eerste stap: als er iets niet deugt, kan het device via LEDs signalen afgeven. Groen betekent: alle meetwaarden vallen binnen de grenzen (beneden-bovengrens). Geel betekent: één of meer waarden wijken gering af. Rood betekent: één of meer waarden wijken sterk af. Op een display moet blijken welke waarde afwijkt / afwijken en hoe sterk. b. Een stapje verder: het kan zijn dat het apparaat jou informeert via een schermpje, geluiden of een bericht op je mobiel. Zorg dat je apparaat één van deze extra opties kan gebruiken. c. Nog een stap verder: je kunt het device ook signalen laten versturen naar regelapparaten. Zorg dat je apparaat verwarming, verlichting of ventilatie kan in- of uitschakelen (alsof je een lamp aan- of uitschakelt). Laat je device ook een alarm geven wanneer er schadelijke concentraties stoffen in de lucht zijn.

0.16 Aanwijzingen

0.17 1. Vochtmeting

Voor Arduino zijn diverse vochtsensoren beschikbaar. Een standaardsensor is bijvoorbeeld geschikt om in een bloempot te zetten en te meten of de plant water moet hebben. Met een kleine aanpassing kan hier een apparaat van gemaakt worden dat vocht in muren meet.

0.18 2. Fijnstofmeting

Bij de module Fijnstof zit een pakketje met fijnstofsensor (lasergestuurd) die gekoppeld is aan fijnstofmeting in de lucht. Met kleine aanpassingen zou dit geschikt zijn om fijnstof in huis te meten. 3. Lichtmeting

Arduino heeft diverse lichtsensoren. Hiermee is redelijk eenvoudig de lichtsterkte te bepalen. Met geschikte sensor ook in Lux. 4. Temperatuurmeling (ook temperatuur/vochtigheid)

Arduino heeft sensoren voor temperatuur. Prima geschikt als thermometer voor een ruimte. 5. Tocht / convectie

Voor tocht zou een schoopenrad gebruikt kunnen worden (op een rotatiesensor). Rook is wat ingewikkelder. 6. Gehalte **CO₂** in de lucht /Gehalte schadelijke gassen in de lucht (gecombineerde sensor) Voor deze metingen zijn speciale sensoren beschikbaar.

0.18.1 7.2 Wetropolis: sensoren en actuatoren voor waterbeweging

Nederland heeft een ervaring van eeuwen met het leven met water. Toch is het nodig dat we onderzoek blijven doen hoe we moeten omgaan met het veranderende klimaat. Meer droogte, hevige regenval in korte tijd, stijging van de zeespiegel: het vraagt aanpassingen. Wat we het beste kunnen doen, wordt onderzocht met modellen. In het Verenigd Koninkrijk zijn vaak problemen met wateroverlast. In Leeds zorgde de rivier Aire ervoor dat de stad in korte tijd blank stond. Wat er moest gebeuren om dat te voorkomen wist niemand. Onderzoekers hebben toen een model gebouwd van de rivier en de stad, om te onderzoeken wat de beste aanpak was. Dat model werd Wetropolis genoemd. In Nederland zijn we dit model verder gaan ontwikkelen om ook in het onderwijs te gebruiken. In het model stroomt water, kunnen overstromingen ontstaan en wordt gekeken hoe regenval en waterafvoer het waterpeil in het landschap verandert. Om het model realistisch te laten werken is het nodig om regenval te simuleren (dus: water op het model laten regenen), het waterpeil in een rivier, sloot of bodem te meten, stroomsnelheid te bepalen.

Ontwikkel op basis van je kennis over digitale sensoren en meetprincipes een systeem waarmee:
 1 Het waterpeil in het model nauwkeurig en automatisch kan worden gemeten
 2 De stroomsnelheid van water automatisch kan worden gemeten
 3 De afvoersnelheid en afgevoerde hoeveelheid water door buisje (regenpijp, riool) kan worden gemeten in de tijd
 4 De aangevoerde hoeveelheid rivierwater kan worden geregeld en gevarieerd
 5 Regenval nagebootst kan worden. Alle sensoren en actuatoren moeten uitgelezen worden en data moet op één plaats verzameld worden. Kijk voor informatie over technieken van sensoren en modelbouw op wetropolis.nl.



Figuur 7.1 Originele Wetropolis model voor rivierwater in de stad

0.18.2 7.3 Digitale devices die een oplossing zijn bij opwarming in de stad

In de grote steden zal bij klimaatverandering de temperatuur harder toenemen dan in het omliggende landelijk gebied. De NLT module “summer in the city” gaat hier helemaal over. Voor sommige van die gevolgen zou je een digitaal device kunnen maken dat de gevolgen hiervan verminderd.

Opdracht: Ontwerp een apparaat dat ons leven fijner maakt als de steden flink gaan opwarmen.
 Voorbeelden zijn:

- Vogelhuisjes met airco

- Drinkbakjes voor dieren die door een sensor water geven als het bakje leeg is en er een dier wil drinken
- Drinkbekers die meten hoeveel er gedronken wordt en een signaal geven als je te weinig drinkt
- Zweetdetectiesysteem in je kleding om je te waarschuwen als je teveel hebt gezweet en mogelijk gaat stinken
- Een systeem dat meet welke vuilnisbakken in het park te vol zitten en dus mogelijk erg gaan stinken. Vervolgens geeft het systeem een waarschuwing af zodat er gericht geleegd kan worden
- Een systeem dat de troebelheid in vijvers meet (met mogelijk in de zomer blauwalgrisico) en waarschuwt als het te troebel wordt (zie figuur 7.2).

Er zijn nog veel meer dingen te bedenken. Bovenstaande apparaten zijn ook al echt door leerlingen



gebouwd, dus het kan echt.

Figuur 7.2 Een device dat leerlingen gemaakt hebben om troebelheid in vijvers te meten

0.19 Stappenplan

1. Bedenk een concreet probleem dat met technologie opgelost kan worden.
2. Stel de eisen op die je aan het apparaat stelt.
3. Breek het ontwerp op in stukken:

Welke zaken wil je meten met sensoren? Wat voor effect moet je apparaat hebben? Hoe moet de data opgeslagen worden? Moet de data verzonden worden? Door eerst elk probleem apart op te lossen kom je uiteindelijk sneller bij een totaaloplossing. 4. Test het ontwerp en pas het aan tot je tevreden bent

Wellicht kunnen jullie in de klas een wedstrijd doen wie het beste ontwerp bouwt voor een probleem. Op die manier wordt het ook belangrijk dat je je eigen ideeën geheim houdt.

0.19.1 7.4 De Q-strip: meten van transpiratievocht

0.19.2 7.4.1 Introductie

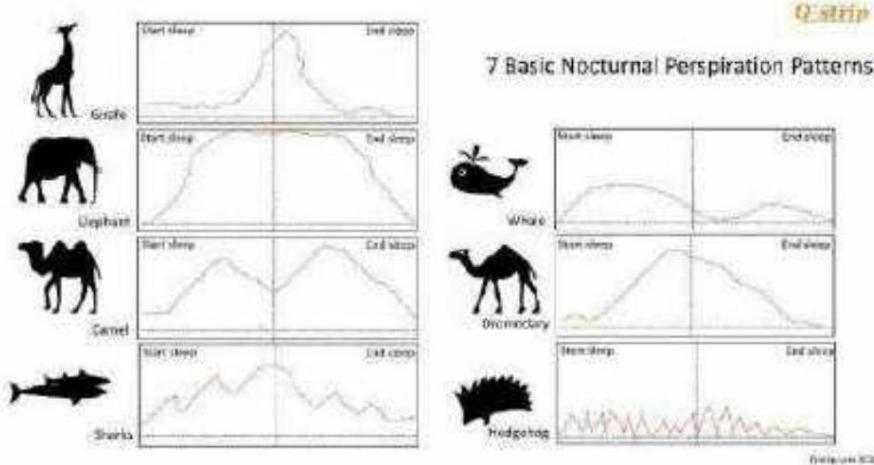
0.20 Meten aan zweten

Het meten van vocht heeft veel zinvolle toepassingen. Eén daarvan is het meten van het vocht dat we zelf afgeven via onze huid: transpiratievocht. Nuttig om te kunnen afkoelen als we het warm hebben, lastig als we fietsen in een waterdicht regenpak of 's nachts in ons bed badend in het zweet wakker schrikken.

Het kan bijvoorbeeld voorkomen dat je ziek wordt, bijvoorbeeld door griep. Als je lichaamstemperatuur omhoog moet, probeer je zoveel mogelijk warmte vast te houden. Je gaat rillen en krijgt kippenvel. Je kruipt in bed. Als je het te warm hebt gaan je lichaam flink transpireren (zweten). Dat gebeurt ook als de koorts weer zakt of als je een paracetamol of aspirine hebt geslikt.

Tijdens de nacht produceren we vocht door transpiratie, soms tot wel een liter per nacht. Dat gaat niet heel gelijkmatig en is ook niet bij iedereen hetzelfde. Als het gelijkmatig en niet hinderlijk is, spreken we over nachtelijke transpiratie. Daarin zijn bij mensen zo'n zeven patronen te herkennen, waarbij de vorm naar een diersoort genoemd is die er op lijkt.

Als je er last van hebt en er bijvoorbeeld wakker van wordt, praten we over nachtzweten. Ook bij



veel diersoorten blijkt

Figuur 7.3 Patronen voor nachtelijk zweten, gemeten met Q-strip nachtelijke transpiratie in patronen voor te komen.

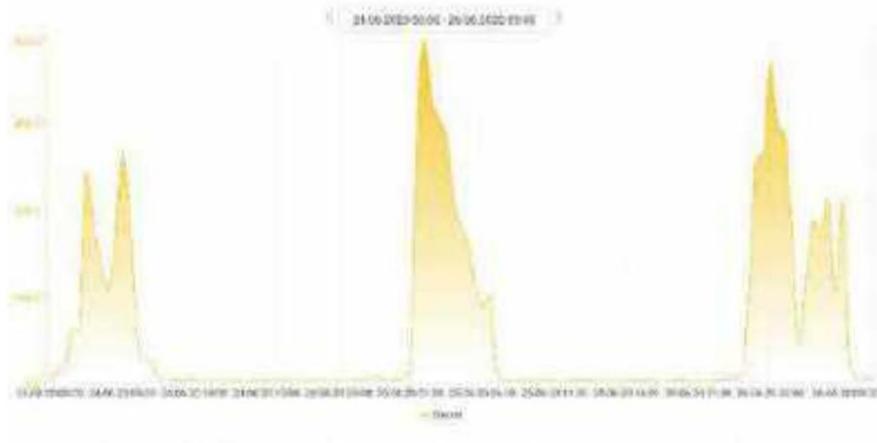
Nachtzweten is ook een symptoom van verschillende ziekten. Het komt voor bij sommige vormen van kanker (zoals leukemie). Overmatig zweten komt ook voor in de dagen voorafgaand aan een hartinfarct. Zweert is daarom een steeds populairdere informatiebron voor de medische wetenschap. Andere informatiebronnen zijn bijvoorbeeld bloed, urine, speeksel en tranen. Naar de laatste twee vormen wordt nog veel onderzoek gedaan.

Momenteel worden wereldwijd diverse instrumenten ontwikkeld om de samenstelling van zweet te kunnen meten. Inmiddels is bekend dat je diverse stoffen in het zweet terug kunt vinden: ionen als natrium, kalium, chloride en ammonium, alcohol, melkzuur, peptiden en eiwitten (Bron: Sweat diagnostics). Deze stoffen noemen we biomarkers en vertellen ons veel over het functioneren van het lichaam. Als een waarde afwijkt van wat normaal is, kan dat betekenen dat iemand ziek is. Biomarkers zijn dus heel belangrijk bij het ontdekken en bestrijden van ziekten. In Nederland wordt veel onderzoek gedaan naar biomarkers.

Naast de samenstelling van het zweettrechter is ook van belang op welke momenten dat zweet wordt geproduceerd. Het gaat dan niet om nachtzweten, waarvan je wakker wordt, maar de normale onbewuste zweetpatronen. Die kunnen gemeten worden met de Q-strip. Dat is een eenvoudige sensor van papier en geleidend materiaal, die je op het matras kunt leggen. Een kleine computer meet de geleiding en daarmee de hoeveelheid en moment van zweetproductie.

0.21 Onderzoek doen

Dat zweetproductie een informatiebron is weten we, maar wat die zweetproductie ons kan vertellen is nog niet duidelijk. Er is nog nauwelijks wetenschappelijk onderzoek naar gedaan, voornamelijk omdat meten aan zweten heel complex is. In het verleden werd dat gedaan door proefpersonen te laten slapen in kamers met een temperatuur van 30 graden Celcius (pfff). Met de Q-strip



Figuur 7.4 Meetresultaten Q-strip wordt dat een heel stuk eenvoudiger, omdat die gevoelig is voor de normale zweetproductie bij de normale slaapttemperatuur.

De Q-strip is een eenvoudige sensor die door iedereen zelf te maken is, en geschikt is voor het zelf meten en data verwerken. Ook de meetcomputer is zelf in elkaar te zetten en te programmeren, en via het internet kunnen de data verzameld en geanalyseerd worden. Daarmee is de Q-strip een ideaal digitale technologie-project. Het onderzoek dat je hiermee doet kan een belangrijke bijdrage leveren aan het voorkomen en genezen van talrijke nare ziekten. Doe je mee?

In de opdracht is een aantal onderdelen te herkennen:

- Standaardopdracht: maken en meten
- Uitzoeken hoe de sensor optimaal gemaakt kan worden
- Meerdere sensoren gebruiken
- Wearables voor dataverzameling overdag
- Verzamelen en transport van data
- Analyse van meetdata om te ontdekken wat de transpiratiepatronen betekenen



Figuur 7.5 Plaatsing van een Q-strip sensor op matras

- Privacy en beveiliging van data

0.22 Zelf maken van een Q-strip sensor

A Opdracht: Maak en test een kleine Q-strip-sensor Benodigdheden:

- Keukenrol (geribbeld, 1300 × 40 ribbon)
- Koolstofvezel 4K (200-400 TEX)
- Uierzalf (gebruik géén vaseline)
- Wateroplosbare (knutsel)lijm
- Schaar (scherp)
- Strijkijzer
- Papier

Smeer de koolstofdraden dun in met uierzalf. Dit zorgt ervoor dat de 7 micrometer dunne koolstofvezeltjes aan elkaar blijven kleven. Ook kun je zo een bundel koolstofvezel splitsen



Figuur 7.6 Een Q-strip sensor en meetdevice (Arduino en vochtsensor ingebouwd) in dunne bundels (800 TEX kan gesplitst worden in 4 draden van 200 TEX). [NB: Een droge bundel koolstofvezel laat bijna onzichtbare dunne draadjes los. Ze kunnen kortsluiting geven in elektrische apparaten, maar ook in je Q-strip, waardoor die niet meer werkt. Het dragen van adembescherming wordt aangeraden. Als je met een botte schaar knipt ontstaan er ook losse vezeltjes].

Draai bundels in elkaar om een gelijkmatige draad te krijgen. De uierzalf helpt ook om goede geleiding tussen papier en koolstof te krijgen voor een goed werkende Q-strip.

De DIY Q-strip is een sensor die je maakt van een strook papieren tissue met 2 dunne koolstofdraden over de volledige lengte van de strip. De koolstofvezels moeten een afstand hebben tussen 22 en 24 mm . Het is de kunst om de koolstofdraden te stabiliseren, alsof ze deel uitmaken van het tissuemateriaal.

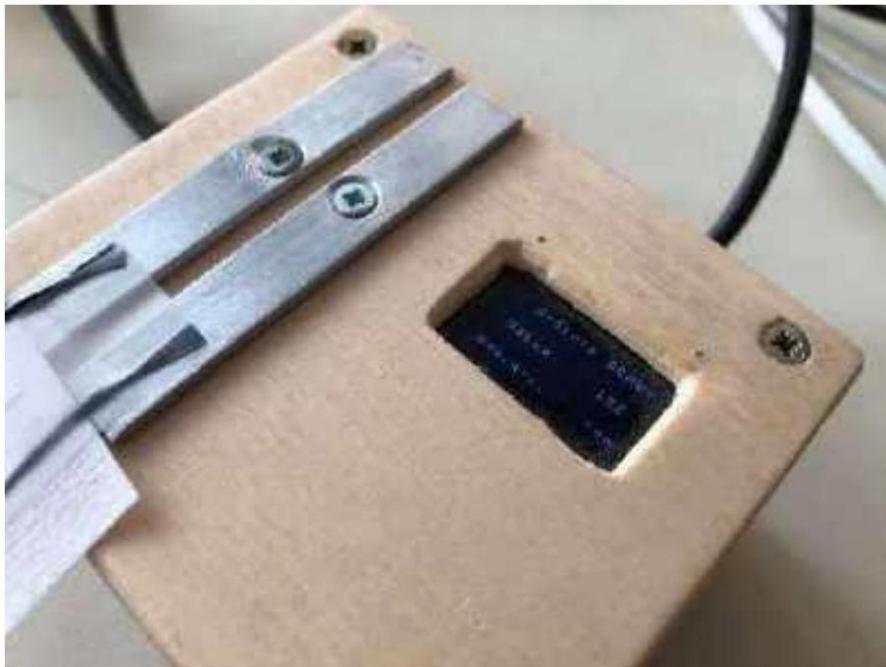
Het maken van de Q-strip sensor Vraagt enige oefening. Begin met één vel keukenpapier en knip daarvan een strook van 9 cm breed. Vouw de randen van de strook van 9 cm naar binnen, zodat een 3 cm breed lintje ontstaat.

Leg de koolstofdraden in de strip van keukenpapier (22 - 24 mm afstand). Zorg dat de draden van koolstofvezel aan een zijde enkele centimeters uit het papier steken. Breng lijm aan tussen de koolstofdraden. Vouw de strip dicht en druk stevig aan (de lijm trekt in het gevouwen papier). Leg een vel papier op de strip en strijk papier, lijm en vezel tot één geheel met een droog strijkijzer (warmste stand). Leg voor het strijken een vel papier op de strip en zorg voor een warmtebestendige ondergrond!. [Een andere manier om de lijm nauwkeuriger aan te brengen is als volgt: Knip ook een strookje tissue van 2 cm breed. Breng daarop ruim lijm aan en verdeel de lijm over het hele oppervlak van het strookje. Leg vervolgens het strookje keukenpapier met lijm tussen de koolstofvezels, vouw de strip dicht en strijk vast.]

Gebruik de Q-strip testunit, of een vochtigheidssensor, met een display om te testen. Druk (of plak met een plakbandje) de uiteinden van de koolstofdraden op de metalen strips van de meetunit en lees de geleiding af. [De geleidingswaarden zitten in deze range: droog 0, vochtig 50-350 en nat 600-800].

Als de draden goed geïsoleerd zijn, kan er alleen stroom van de ene naar de andere draad komen als er vocht in de tissue terecht komt. De Q-striptranspiratiesensor is dan klaar voor gebruik.

De koolstofvezel kan met geïsoleerd metaaldraad verlengd worden (solderen

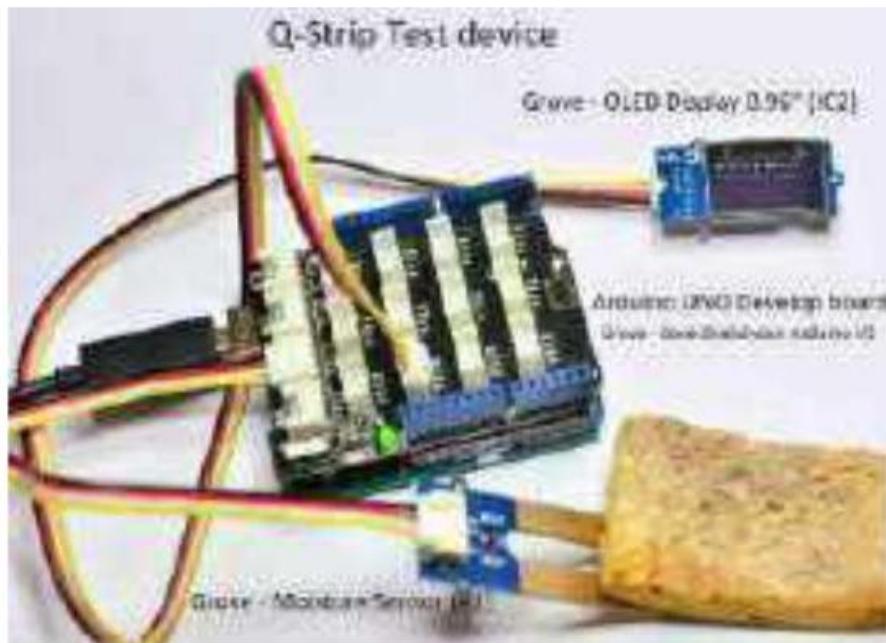


Figuur 7.7 Een testdevice voor vochtsensor: de metalen strips maken contact met de Arduino-vochtsensor in het kastje en het display laat de geleidbaarheid zien. Het binnenwerk van deze device kun je zelf bouwen met de instructies. is onmogelijk). Vlecht de draden koolstof en metaal in elkaar en zet ze vast met lijm of tape, een krimpkous of elektrisch geleidende verf. Bevestig de metaaldraad aan de vochtsensor van de Arduino. [Tip: Je kunt ook werken met diverse andere materialen om (kleine) sensoren te maken, in combinatie met koolstofvezel. Denk aan filtreerpapier, gewoon plakband, voorgelijmd papierplakband, tissue, stofvrije tissue, papierpulp (te maken van tissue of toiletpapier).]

B Opdracht: Maak een lange Q-strip Gebruik dezelfde werkwijze, maar plak nu drie tot vier stroken (9 cm breed) aan elkaar tot een lint van 1 meter lengte. Maak koolstofvezels van 90 cm en verbind ze met geïsoleerde metaaldraad. De overgang koper-koolstof zit ongeveer op 10 cm van het einde van de strip. Het sensorgedeelte van de strip is dan zo breed als een matras.

Voor langdurig gebruik kan de Q-strip in een drager geschoven worden die voldoende stevigheid geeft om de Q-strip onder een laken te kunnen leggen en gedurende de nacht te gaan meten. Wij gebruiken de strip nu zonder extra drager.

De Q-strip zal niet super precies zijn, maar je kunt hele behoorlijke resultaten behalen als je een strip gedurende langere tijd (zeg 14 dagen) gebruikt en kun je iets zeggen over het nachtelijke ritme van de transpiratie van je lichaam.



Figuur 7.8 Arduino met Groove shield, vochtsensor en display. Dit zit ingebouwd in de testdevice van figuur 7.3.5, en kun je zelf bouwen met de instructies.

0.22.1 7.4.2 Het Q-strip-meetsysteem bouwen

0.23 Benodigdheden testdevice

1. Arduino UNO
2. Groves Base shield voor Arduino (v2) HAT
3. Grove OLED display 0.96 (IC2), 128 × 64 display
4. Groove Moisture Sensor (A1)
5. Voeding (5V) USB via lader of computer
6. Code: Q-strip.ino

0.24 LoRaWAN gekoppeld meetdevice

1. Arduino TTN Uno (standaard Arduino met LoRaWAN Chipset)
2. Groves Base shield voor Arduino (v2) HAT
3. Groove Moisture Sensor (A1)

0.25 Methode

1. Maak verbinding tussen Q-strip en pootjes van de vochtsensor (bv: klem, drukknoop, vleugelmoer en oogje)
2. Leg verbinding aan met het LORA netwerk (Internet of Things) en The Things Network.
3. Stel gateway in voor de Q-strip-database.

C Opdracht Gebruik het Q-strip-device De sensor voor het Q-strip-device zijn de lange koolstof-electroden. De weerstand tussen de electroden hangt af van de isolatie tussen de electroden. Als het tissuepapier tussen de electroden droog is, dan is de weerstand hoog. Hoe vochtiger, hoe beter de geleiding, dus hoe lager de weerstand. De electrodestrip wordt aangesloten op een vochtsensor van de Arduino. Normaal gesproken bestaat die sensor uit metalen electroden die, bijvoorbeeld, in de grond worden geprikt. Nu zijn de metalen electroden vervangen door de koolstofdraden in de sensorstrip. De verwerking van de electrodendata met de Arduino gebeurt op dezelfde manier.

Om een relatie te kunnen leggen tussen gemeten waarde en de hoeveelheid opgenomen vocht, is het nodig de Q-strip te kalibreren. Breng een hoeveelheid vocht (bv 1 mL) op de strip aan, wacht tot het vocht verspreid is en noteer de geleiding. Breng telkens meer vocht aan tot de strip verzadigd is. Maak hiervan een kalibratiegrafiek.

Om metingen uit te voeren gedurende de nacht, installeer je de Q-strip onder het laken van je bed, ter hoogte van je borstkas. Bevestig het meetkastje en zorg dat deze elke 10 minuten een meting doet. Vaker of minder vaak kan uiteraard ook. De gegevens kun je opslaan op een geheugenkaartje (SD) of laat je doorsturen via het LoRa netwerk (LoRaWAN) naar The Things Network.

D Opdracht: Verwerken van meetdata

1. Ophalen van data uit de database met metingen
2. Opslaan van lokale data op een SD kaartje
3. Analyse van meetgegevens

0.25.1 7.4.3 Onderzoek en toepassing

0.26 E Verdiepingsopdracht

Gebruik het Q-strip-systeem om onderzoek te doen naar transpiratie in de nacht. Denk daarbij aan de volgende zaken:

- Proefpersonen (hoeveel, welke leeftijd, verschil en overeenkomst)
- Data verzamelen en verwerken
- Bescherming van privacy
- Wat betekent een transpiratiepatroon?
- Hoe kan het Q-strip systeem handig ingezet worden?

F Verdiepingsonderzoek: De vorm en techniek van de vochtsensor

- Is het mogelijk de sensor in een continuproces te maken (vanaf de rol)
- Welke papiersoort is het meest geschikt om de sensor te produceren?
- Werken kleinere /anders gevormde vochtsensoren ook?
- Kan de sensor van stof (bijvoorbeeld katoen) gemaakt worden?
- Welke soort lijm is het meest geschikt voor het plakken van de sensor?
- Is de vochtsensor draagbaar te maken, om bijvoorbeeld in te bouwen in een shirt, onder de armen, buik, rug, voorhoofd?

0.27 G Verdiepingsopdracht: Meten met de Q-strip

- Kan het meetsysteem in bed gebruikt worden om meer gegevens te verzamelen, bijvoorbeeld draaien, lichaamstemperatuur, luchtvochtigheid, omgevingstemperatuur, vochtdoorlatendheid van matras.
- Kan de meetdata van de sensor gekoppeld worden aan andere draagbare sensoren (bijv een meethorloge, meetring of -armband?)
- Hoe vaak moet de meetdata verzameld worden voor een goed beeld?
- Hoe snel reageert de sensor op de transpiratie / temperatuurverandering?
- Kan met de Q-strip méér gemeten worden dan alleen vochtigheid?

Informatiebron: <https://washingtoncitypaper.com/article/221338/straight-dope-do-you-really-sweat-one-liter-each-night/>

0.27.1 7.5 Andere thema's voor de eindopdracht

De thema's waar je aan kunt werken zijn eindeloos. Enkele suggesties hiervoor zijn:

- Een digitaal gestuurde spuitenkomp (bij de module Lab on a chip)
- Meten met licht in een Lab on a chip (met een led, fotodiode en Arduino)
- Stuur een 3D-printer aan met Arduino
- Maak een zelfrijdend wagentje of robot, op basis van een microcontroller als Arduino, Micro:bit, Raspberry Pi of ander platform.
- Wellicht heb je zelf een idee waar je aan wilt werken. Vraag je docent wat er mogelijk is.
- Je kunt bij verschillende modules die je gedaan hebt (of misschien nog gaat doen) een ontwerp maken waar je digitale technologie voor gebruikt.

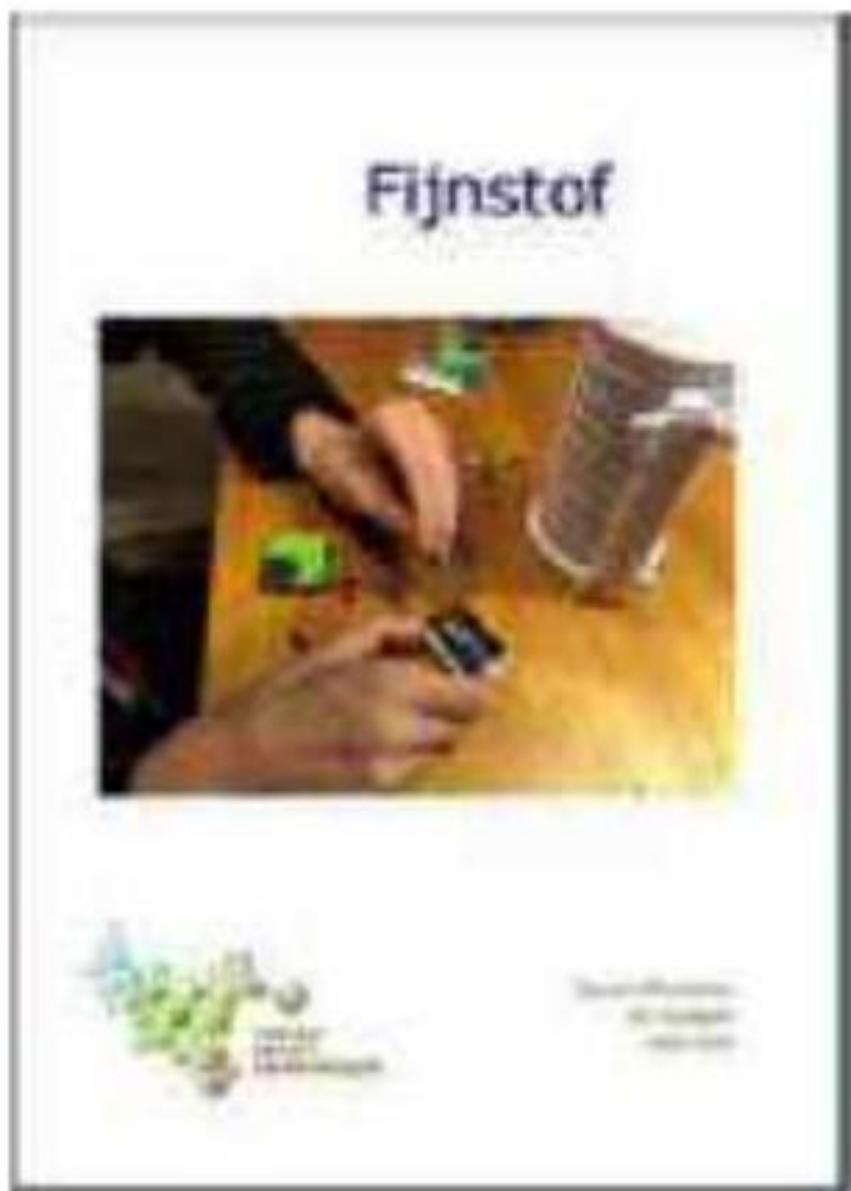
Niet alle opdrachten kunnen we in de module plaatsen. Daarom worden er ook na verschijnen van de module losse eindopdrachten beschikbaar gesteld.

0.28 8 Verder met digitale technologie

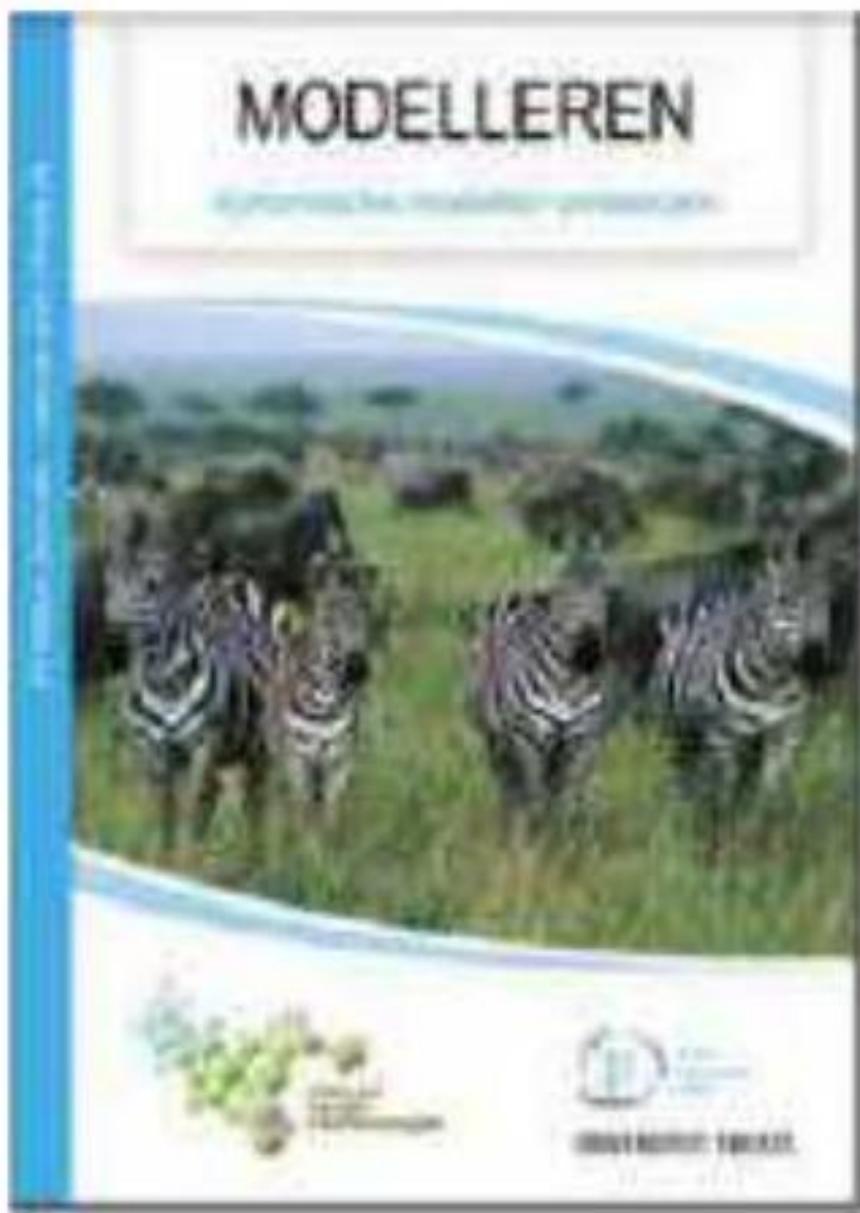
Sommige van onderstaande lesmodules nlt gaan in zijn geheel over een onderdeel van digitale technologie. Bij anderen wordt digitale technologie ingezet in een context. Ook kun je bij een groot deel van de nlt modules iets maken of onderzoeken waar digitale technologie bij gebruikt wordt. Vraag er gerust naar bij je docent of TOA.

0.29 Fijnstof

Nederlanders overlijden gemiddeld 9 maanden vroeger door fijnstof. Hoe zit het met de luchtkwaliteit rondom je school? Is dit aanleiding tot een vervolgplan van deze burgerwetenschapapak? Samen op zoek naar oplossingen voor een betere luchtkwaliteit en een steentje bijdragen aan het Schone Lucht Akkoord. GLOBE Nederland en het RIVM ontwikkelden deze lesmodule om op middelbare scholen de fijnstofconcentratie te meten in hun omgeving. Doe mee met de fijnstofcampagne!



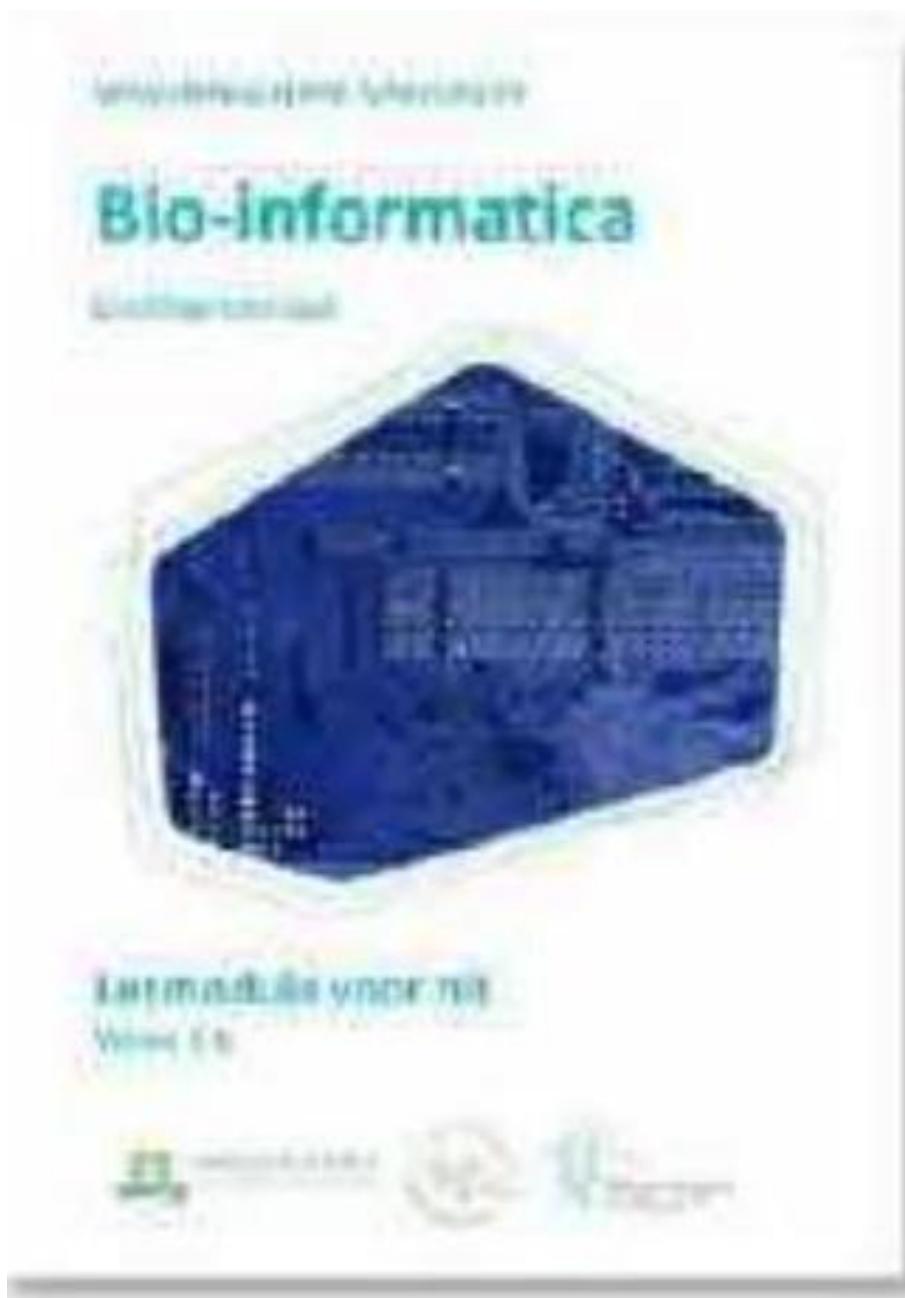
0.30 Modelleren



Modellen helpen de mens om voorspellingen te doen en bedreigingen beheersbaar te maken. In deze module ontdek je hoe situaties en processen met behulp van dynamische modellen beschreven kunnen worden. De contexten waar je aan werkt zijn de verspreiding van corona onder de bevolking, de waterhuishouding in Nederland, de groei van de wereldbevolking en/of het saneren van een vervuilde bodem.

0.31 Bio-informatica

Het concept genexpressie (transcriptie, splicing, translatie) wordt vanuit programmeren in Python benaderd. Hierbij wordt het biologisch concept vertaald naar praktische programmeer opdrachten. Via bestaande tools worden beginselen van het lezen van DNA geoefend. In een individuele eindopdracht wordt naar een onbekend gen(product) en de genetische oorzaak van een afwijkend genproduct gezocht.



0.32 De toekomst van de landbouw



Nederland staat aan de vooravond van een grootschalige herinrichting van de land- en tuinbouw. Het moet slimmer, efficiënter, schoner, met minder energie, minder landoppervlak en vooral ook meer ecologisch. In deze lesmodule verdiep je je in technologische innovaties, die in de landbouw ingezet worden. Robots die planten automatisch herkennen en verwijderen, spuiten tegen insecten, bemesten en watergeven gebeurt op basis van BigData van sensoren van o.a. drones. Je bouwt een eigen visie op over het complexe systeem van schakels binnen de voedselproductie en -distributie. Vanuit die visie werk je toe naar een toekomstplan, waarin al deze aspecten een rol spelen. Zo draag je positief bij aan de toekomst.

0.33 Cybersecurity

Cybersecurity gaat over het beveiligen van digitale informatie en digitale diensten die een bepaalde waarde hebben. Die waarde zorgt ervoor dat je de informatie of dienst wilt afschermen. Uiteindelijk gaan diensten over het lezen of wijzigen van digitale gegevens: de gegevens over je bankrekening, al je emails of de inhoud van je Facebook pagina. ‘Cyber’ in cybersecurity benadrukt extra dat het gaat om security in cyberspace, de virtuele wereld die bestaat uit computers en de netwerken tussen



computers.

0.34 Artificiële Intelligentie (AI)

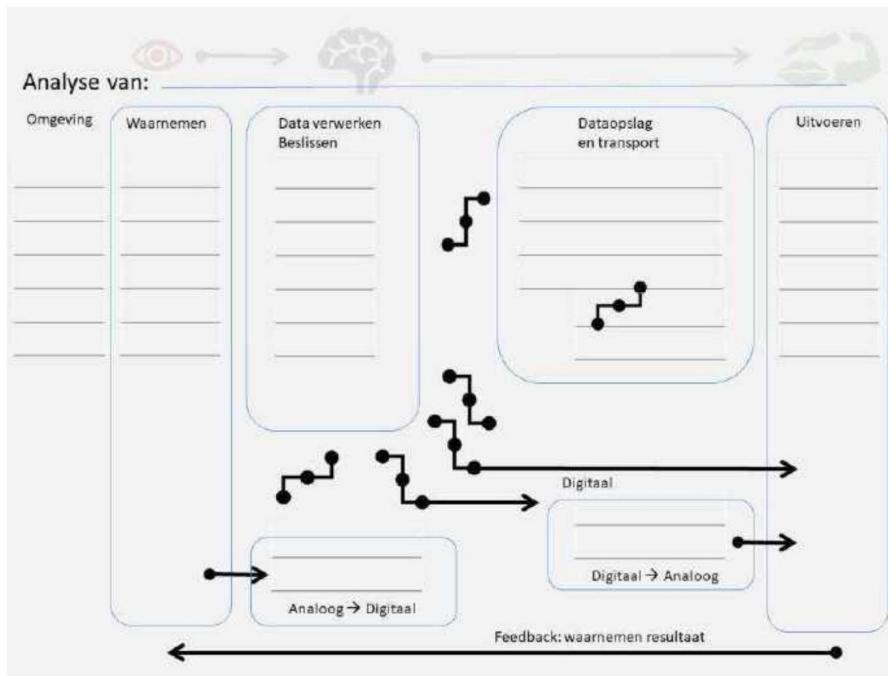
Een lesmodule waarin de leerlingen een positief project doen: ze zetten AI in om de wereld iets beter



te maken. Na afloop van de module heeft de leerling geleerd dat AI een breed vakgebied is, dat gaat van het programmeren van algoritmes tot het doen van medisch onderzoek. Het is een samenspel van verschillende disciplines: spraakherkenning, machine learning, beeldherkenning en natural language processing. Kunstmatige intelligentie wordt gebruikt als je een verzekering wilt afsluiten, als je zit te 'netflixen', als je via social media contact houdt met je vrienden en op nog veel meer manieren.

0.35 Leeg analyseschema

Ook als los werkblad beschikbaar in de moduledatabase (<https://module-database.betavak-nlt.nl/>).



0.36 Bijlagen verkrijgbaar in de moduledatabase

- Bijlage 1: Uitgebreide uitleg bij onderdeel C van H4
- Bijlage 2: Uitgebreide uitleg bij onderdeel C van H5
- Bijlage 3: Uitgebreide weergave van achtergronden en verdieping zoals

Fabricage van computerchips, printplaten Hoe transistoren werken Computergeheugen Logische schakelingen Sensoren, actuatoren Neopixel Verschillende microcontrollers

- Bijlage 4: Uitgebreide begrippenlijst
- Bijlage 5: URL lijst

Er is voor de leerlingen ook veel informatie te vinden op de leerlingenwebsite:
https://maken.wikiwijs.nl/171772/Schakelmodule_Digitale_Technologie_online_materiaal

0.37 Appendix

Online documenten

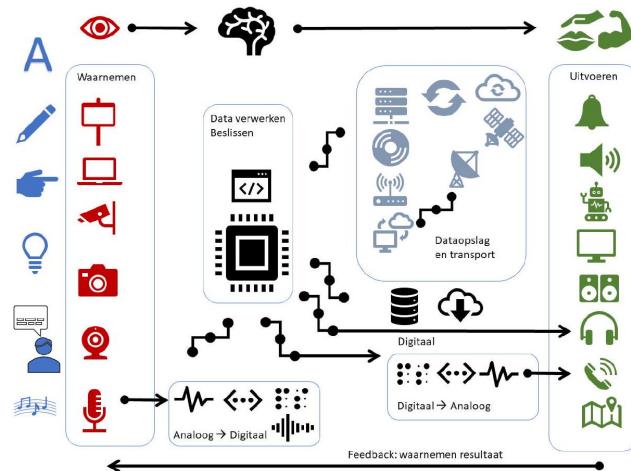
0.37.1 Inleiding

Doeel van de module

Tijdens deze module ga je bekend raken met digitale techniek. Digitale techniek is overal.

Structuur van de module

Digitale techniek bestaat uit een aantal onderdelen. Dat wordt schematisch weergegeven in figuur 1. Vrijwel alle systemen nemen iets uit hun omgeving waar. De koelkast meet de temperatuur in de koelkast, een satelliet in de ruimte neemt bepaalde golflengtes licht waar, een telefoon neemt een signaal van de zendmast waar etc. Dat waarnemen gebeurt met **sensoren**. *Hoofdstuk 3* van de schakelmodule gaat over de waarnemingskant van een digitaal systeem. Vervolgens wordt die waarneming omgezet in een digitaal signaal. Dat betekent dat dat wat gemeten wordt omgezet wordt naar informatie waar computers mee overweg kunnen, 1'en en 0'en, wat we digitale data noemen. Die data worden vervolgens **verwerkt** (*hoofdstuk 4*) en **opgeslagen en getransporteerd** (*hoofdstuk 5*). Verwerken betekent dat er op basis van de data vervolgacties worden vastgesteld. Als de koelkast te warm is, moet de koelmotor aan, als een telefoon het signaal verliest, gaat hij op zoek naar een nieuw signaal, als het niveau in de brandstoftank te laag wordt, gaat het lampje aan etc. Die vervolgactie, de uitvoer, vindt plaats door **actuatoren**. Dat zijn apparaten die in de echte omgeving een effect hebben. Een lampje dat aangaat, een motor die gaat draaien, een speaker die geluid geeft, dat zijn allemaal actuatoren die aan het werk worden gezet. Deze uitvoer ('output') van een digitaal systeem komt in hoofdstuk 6 aan bod.



Alle onderdelen van digitale techniek. Dit figuur gaat vaak terugkomen in deze module wikiwijs

Combinatie met andere modules

Omdat er in veel andere vakgebieden digitale systemen zijn te vinden gaat de informatie van deze module je ook helpen bij andere modules. Bij de modules over het hart (Hart en vaten & Leef met je hart) is er bijvoorbeeld het ECG. Door middel van sensoren wordt er informatie verzameld over je hart en dit kan op een scherm bekijken worden. Maar digitale systemen kunnen ook oplossingen bieden voor problemen. Zo is er een module Ruimte voor de rivier, waarbij je natuurlijk met slimme elektronische systemen sluizen open en dicht kunt zetten om zo het waterpeil in een rivier te reguleren. Bij Summer in the city kun je met behulp van een slimme regenton (zie de link hieronder) je tuin beter voorbereiden op weersextremen. De mogelijkheden zijn eigenlijk eindeloos.

Daarnaast is deze module een zogenaamde schakelmodule. Deze module helpt om alle leerlingen, ongeacht hun achtergrond in dit onderwerp, op een basisniveau te krijgen zodat in een verdere module daarop voortgebouwd kan worden.



[Slimme regenton] (<https://slimmeregenton.nl/>)

Leerdoelen

Tijdens dit hoofdstuk leer je:

- Dat ontwerpen niet in 1 x volmaakt zijn, maar volgens het proces van de ontwerpcyclus stap voor stap verbeteren
- Dat je door de ontwerpcyclus goed toe te passen je een overzichtelijker verslag van je ontwerpproces krijgt en een beter ontwerp
- Dat ontwerpen vaak een kwestie is van gewoon doen en dan goed evalueren en dat fouten maken eigenlijk cruciaal is om tot een goed ontwerp te komen
- Dat je leert welke praktische stappen er nodig zijn om van een ontwerp op papier naar een echt werkend prototype te komen
- Dat je leert nadenken over technische oplossingen voor problemen en deze oplossingen kunt realiseren.

Studeren en werken in de digitale technologie: kunstmatige Intelligentie filmpjes van Daan Geijs

Daan Geijs heeft over zijn opleiding en werk vier filmpjes gemaakt. Hij vertelt over zijn opleiding en onderzoek, en waar hij later aan de slag kan met zijn kennis van digitale technologie.

- 1 zijn vakken op de middelbare school en zijn studie aan Universiteit Twente
- 2 wat Kunstmatige Intelligentie is, en waar het voor gebruikt kan worden
- 3 welke rol kunstmatige intelligentie in het ziekenhuis kan hebben: promotieonderzoek bij het Radboud UMC
- 4 welk werk Daan hierna kan gaan doen

Introductie digitale technologie. Daan Geijs vertelt over zijn studiekeuze en opleiding

Wat is kunstmatige intelligentie? Daan Geijs vertelt wat er met digitale technologie mogelijk is.

Kunstmatige intelligentie in het ziekenhuis. Daan Geijs vertelt over zijn onderzoek aan het herkennen van tumoren met de computer.

Daan Geijs over het werk dat hij in de toekomst kan gaan doen

0.37.2 Ontwerpen en ontwerpcyclus

Jullie gaan tijdens deze module tot een prototype komen. Daar is een bepaalde manier voor om dat zo goed mogelijk te laten verlopen.

Probleemstelling

Als eerste moet je vaststellen voor welk probleem je een oplossing gaat ontwerpen. Stel: je kat gaat iedere vakantie dood van de honger omdat je weg bent en er niemand eten kan geven, en jij wilt een apparaat bouwen dat de kat automatisch eten geeft. Of: de vogels in het vogelhuisje in de tuin raken oververhit in de zomer en je wilt iets bouwen dat het vogelhuisje koelt als de temperatuur te heet wordt. Of: Je wilt een wekker bouwen die licht geeft als de alarmtijd aangebroken is.

Je ziet het al: het doel van je ontwerp (nml. het oplossen van een probleem) bepaal je zelf. Het is handig om het probleem zo concreet mogelijk te omschrijven. Als je doel bijv. ‘wereldvrede’ is, dan is dat veel te breed en te vaag om een goed startpunt te zijn voor een ontwerpcyclus. Vaak kom je in een ontwerpcyclus later nog terug op het probleem als je er in eerste instantie achter komt dat het probleem te groot is om met je ontwerp op te lossen. In dat geval stel je je doel bij (tenzij je een opdracht hebt gekregen van een opdrachtgever, dan moet je van tevoren inschatten of het haalbaar is).

Pakket van eisen

De tweede stap is dat je gaat vaststellen aan welke eisen je ontwerp moet voldoen om te zorgen dat je probleem ook echt opgelost wordt. Een voorbeeld: als je een apparaat wil bouwen dat automatisch onder water de troebelheid meet als waarschuwingssysteem voor blauwalg, moet het natuurlijk wel waterdicht zijn.

Het kan ook hierbij het geval zijn dat je gedurende je ontwerpcyclus erachter komt dat je een eis mist, die voeg je dan in de volgende ronde toe. Let op dat ook eigenschappen als prijs of afmetingen van het ontwerp belangrijke eisen kunnen zijn.

Deeluitwerkingen

Een ontwerp is vaak complex en heeft verschillende aspecten en moet aan verschillende eisen voldoen. Er is de kwestie van de stroomtoevoer, het systeem moet dingen waarnemen, maar ook output geven. De code moet kloppen en de code bevat verschillende onderdelen om de verschillende onderdelen van je systeem aan te sturen. Om dit alles op een ordelijke manier te ontwerpen helpt het als je je ontwerp opbreekt in kleine stukken. Zo kun je ook makkelijker de taken verdelen binnen een team.

Bij de fase van ‘deeluitwerkingen opstellen’ breekt je ontwerpproces dus op in kleine stukken die je op zichzelf kan maken en testen waarna je daarna alle deeluitwerkingen (stap voor stap) samenvoegt. Dat samenvoegen is een complexe stap, en daarom moet je dit stap voor stap doen. Als je namelijk in 1 x alles bij elkaar zet en er is iets fout, dan weet je namelijk niet welk onderdeel de fout veroorzaakt.

Ontwerpvoorstel

In het ontwerpvoorstel ga je beschrijven welke deeluitwerkingen je samenvoegt voor je ontwerp. Dus stel dat je bij een wake-up light in de 1e ronde alleen de code schrijft die de klok laat lopen en op een display laat verschijnen, dan beschrijf je dat. Op die manier kun je hier bij je evaluatie op terugkomen. (in de praktijk voer je deze en de volgende stap tegelijk uit)

Prototype

Nu ga je aan het bouwen. Je zorgt dat je de stappen die je hebt gemaakt werkend hebt (dus als je een code hebt die de klok laat lopen en op een display laat verschijnen, dan laad je niet alleen de code op de Arduino, maar je sluit ook alle draadjes aan voor de display, zodat het systeem ook daadwerkelijk kan werken). Dit is de leukste stap, maar ook het meest frustrerende, omdat er vaak dingen niet werken of door jou slecht zijn uitgedacht of voorbereid. Dat hoort erbij, deal with it.

Evaluatie

Dit is een hele belangrijke stap, en een stap waarvan het heel belangrijk is dat je al je waarnemingen goed opschrijft. Je beschrijft nu namelijk wat er van je ontwerp wel en wat niet werkt. Vaak is dit in het begin meer niet dan wel. Fail and try again.

Om te zorgen dat je niet steeds dezelfde problemen tegenkomt is het belangrijk dat je je evaluatie gestructureerd aanpakt. Zorg dat je de datum vermeld, wat werkte er wel, wat werkte er niet, en je vermoedens waar het aan ligt.

Ga in deze fase niet eindeloos lopen prutsen, maar pak de verbetering gestructureerd aan door weer terug te gaan naar je pakket van eisen, deeluitwerkingen etc. en een goed plan te maken voor de volgende verbetering. Zo ga je dus nog een keer de ontwerpcyclus in. Bij dit project kun je zo 10-15 ontwerpcycli doorlopen. Het is dus ook slim om je codes van je Arduino programma een versienummer te geven, zodat je altijd achteraf kunt laten zien wat er steeds is veranderd, of dat als je in versie 5 iets blijkt gesloopt te hebben wat in versie 4 nog werkt, je altijd weer terug kan gaan naar de laatst werkende versie.

Contexten

Er zijn ontzettend veel contexten waarvoor je een digitaal systeem kunt ontwerpen. Digitale systemen zijn namelijk overal om je heen!

In het voorbeeld in hoofdstuk 8 zie je hoe de auteur van de site een ontwerpopdracht aanpakt rondom de context van een wake-up light. Eigenlijk zijn alle systemen waarin iets van een klok of tijdwaarneming zit interessante systemen. Je moet namelijk een interessante code schrijven, er iets van een input nodig, en je hebt output op basis van die tijd.

Een andere interessante context is corona. Hoewel de meesten er natuurlijk zat van zijn, kan technologie ons helpen er beter mee te leven. Zo kun je bijvoorbeeld een apparaat bouwen waarmee je met een ultrasone sensor de afstand tot objecten om je heen kunt meten? Stel dat je met zo'n apparaat om je middel door de kantine loopt? Hoe vaak is er dan iemand binnen 1,5m? Of kun je dan een waarschuwingssignaal laten klinken als feedback?

Ook kun je op basis van een lichtsluis meten hoeveel mensen er in een winkel zijn. Is het maximum bereikt kan er een stoplicht op rood springen.

Ook sport is interessant: stel je gaat hardlopen en je meet de tijd dat je daarover doet. Misschien kun je een apparaat maken dat jou feedback geeft over je rondetijden? Je hebt misschien wel eens bij het schaatsen op TV gezien dat als iemand een wereldrecord lijkt te schaatsten dat je dan op het ijs een lijtje geprojecteerd ziet van die tijd.

De mogelijkheden zijn eindeloos. De uitdaging zit hem erin vooral snel iets te gaan bouwen en niet te lang bezig zijn om de perfect context te vinden. Voor deze module is het namelijk vooral belangrijk ermee aan de gang te gaan en in de praktijk te leren.

0.37.3 Hardware voor de verwerking van informatie in een digitaal systeem

Arduino Uno

In deze module gebruiken we vooral de Arduino Uno. Een Arduino (Uno) is een klein board met programmeerbare elektronica. Met een Arduino kan je verschillende elektrische componenten aansturen. Denk aan lampjes, schakelaars, motortjes en verschillende sensoren.

De Arduino Uno Revision 3 - vaak Arduino Uno r3 of Arduino Uno rev3 genoemd - is het meest gebruikte board van Arduino. Dit board is geschikt voor beginners en gevorderde gebruikers. Mede daardoor is de Arduino Uno zo populair.

De Uno is gebaseerd op de ATmega328 chip en heeft 14 digitale input/output pinnen. Verder zijn er 6 analoge input pinnen en een USB-aansluiting. Het board is eenvoudig aan te sturen door een programmeercode te schrijven (zie elders in deze module).



Figure 1: Arduino uno

Je ziet in de afbeelding hierboven in rood vierkant de genummerde digitale poorten. Deze worden vaak gebruikt om aangesloten outputs (lampjes, motortjes e.d.) aan of uit te zetten). Let op: poort 0 en 1 hebben een speciale functie.

Verder zie je in een groen vierkant de analoge poorten. Waar een digitale poort alleen 1 of 0 als waarde kan aannemen, kan een analoge poort veel meer waardes aannemen. We komen hierop terug bij de verschillende sensoren.

In blauw hebben we de 2 verschillende spanningen. Soms heb je systemen die om 5 V vragen en soms om 3.3V (dit is de plus).

In geel heb je de aarde (of de min) aangegeven met GND.

Arduino Nano

De Arduino Nano is vergelijkbaar met de Uno. Hij is wat kleiner, wat een voordeel is in ontwerpen die heel klein of licht moeten zijn. Het nadeel is dat hij wat minder krachtig is en minder digitale poorten heeft.

Arduino Mega

De Arduino Mega 2560 Rev3 is een microcontroller board gebaseerd op de ATmega2560. Het heeft 54 digitale input / output poorten en 16 analoge poorten. Daarmee heeft hij meer mogelijkheden dan de

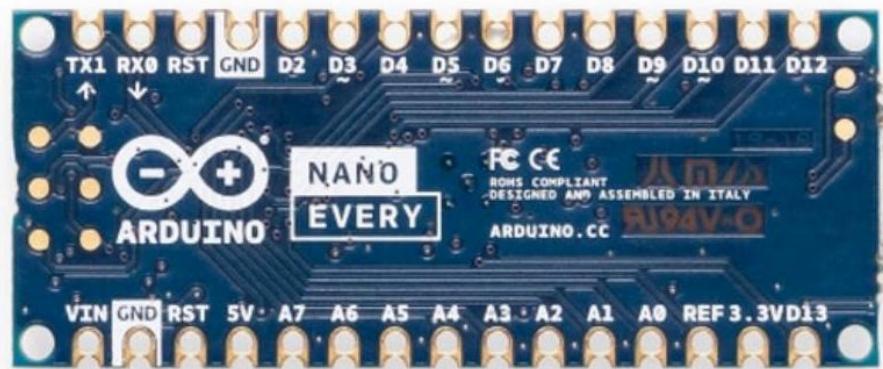


Figure 2: Arduino Nano

Uno. Omdat we meestal bij onze projecten niet zoveel poorten nodig hebben gebruiken we deze niet standaard.

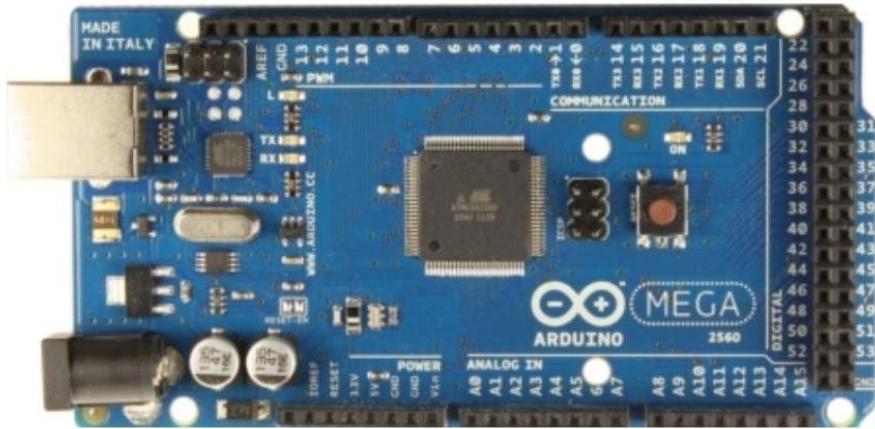


Figure 3: Arduino Mega

Micro:bit

tekst

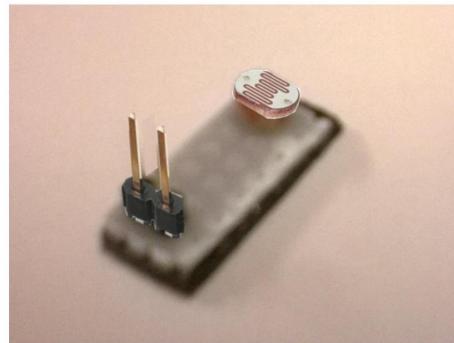
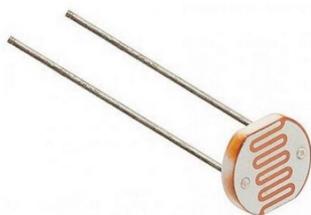
Raspberry Pi

tekst

0.37.4 Hardware voor de input van een digitaal systeem

LDR sensor

Een lichtgevoelige weerstand of LDR (light-dependent resistor) is een elektrische component waarvan de weerstand beïnvloed wordt door de hoeveelheid licht die erop valt. De weerstandswaarde van een LDR wordt kleiner, naarmate de LDR sterker wordt belicht. Dat betekent dat als je een LDR aansluit op je analoge poort én de poort programmeert met Pull-Up naar Vcc, je de waarde van de spanning zult zien afnemen als er meer licht op de LDR schijnt. Wil je dat de spanning stijgt bij grotere lichtsterkte, dan heb je een Pull-Down weerstand naar GND nodig. Die heeft de Arduino niet aan boord dus die plaats je als fysieke weerstand extern.



LDR

Omgekeerd evenredige spanning met lichtsterkte

De LDR sluit je aan door een draad van de GND aan het ene uiteinde te verbinden (hier zwart) en het andere uiteinde met een willekeurige analoge poort te verbinden (hier rood). Poort programmeren met Pull-Up.

Evenredige spanning met lichtsterkte

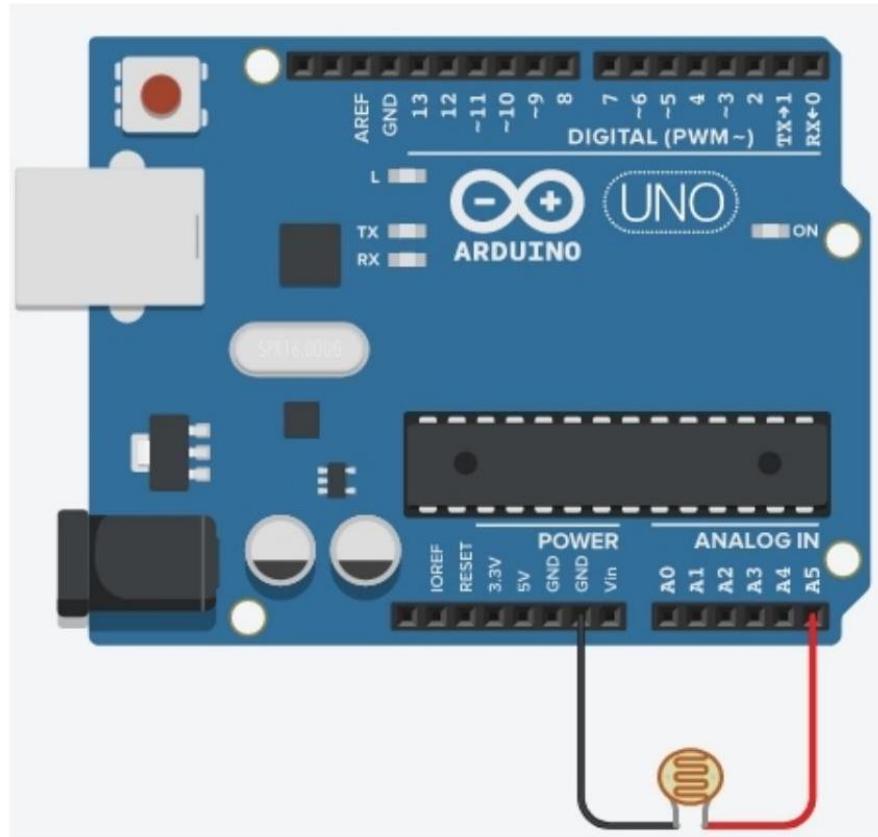
De LDR sluit je aan door een draad van de Vcc aan het ene uiteinde te verbinden (hier rood) en het andere uiteinde met een willekeurige analoge poort te verbinden (hier groen). Ook plaats je een fysieke weerstand tussen de analoge poort en de GND.

Momentdrukknop

Een drukknop is een veelgebruikte input. Je drukt op een knop en er gebeurt iets. De drukknop is gebaseerd op dat er als de knop ingedrukt is er een stroom doorheen kan lopen die minder weerstand ondervindt dat de route direct terug naar de min draad. In de schakeltekening zie je dat als de knop niet is ingedrukt, de stroom niet door de knop kan en dan via de weerstand met de zwarte draad naar de min gaat. Als de knop wel is ingedrukt gaat het door de groene draad (via een weerstand, anders verbrandt de LED) naar de LED, die dan dus aan gaat.

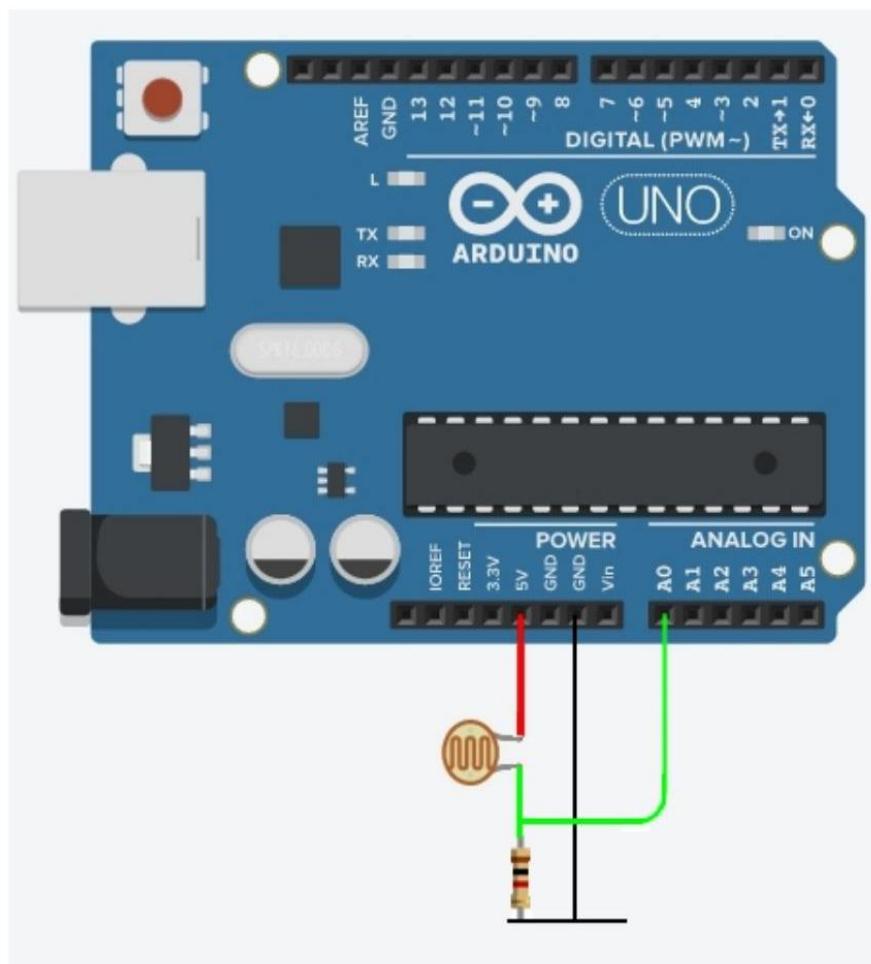
Je moet in de code aangeven in welke poort het signaal van de drukknop binnengaat (in de schakeling is dat digitale poort 2). Hieronder zie je ook een voorbeeldcode passend bij de schakeltekening.

Een schakeling waarmee een drukknop een LED bedient

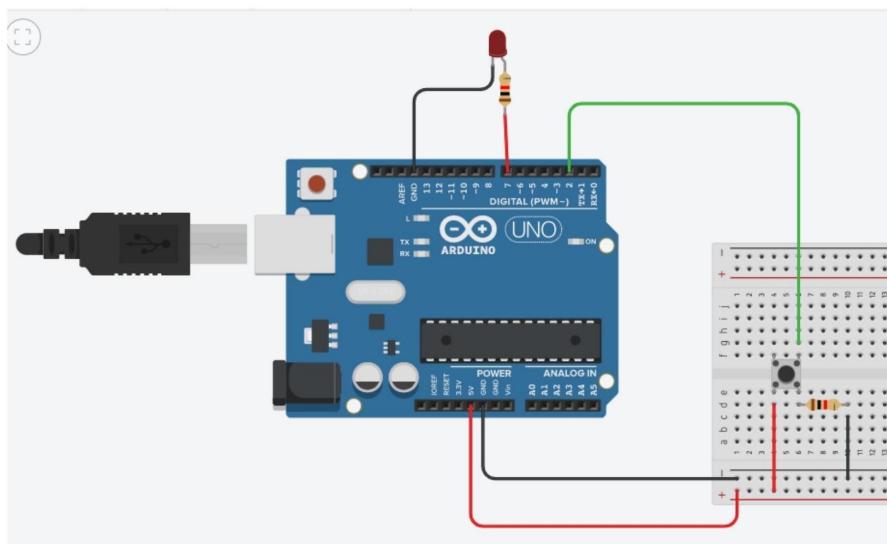
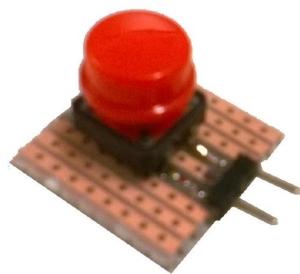


Opstelling

```
// Pin 2 has an pushbutton connected on most Arduino boards.
// give it a name:
int pushButton = 2;
int led = 7;
// the setup routine runs once when you press reset:
void setup() {
    //initialize serial communication at 9600 buts per second
    Serial.begin(9600);
    //make the pushButton's pin an input:
    pinMode(pushButton,INPUT);
    pinMode(led, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
//read the input pin
    int buttonState = digitalRead(pushButton);
    Serial.printIn(buttonState);
    delay(1);
//read the output pin
    if(buttonState > 0 ){
        digitalWrite(led,HIGH);
        delay(1000);
        digitalWrite(led,LOW);
        delay(1000);
    }
    else
    {}
}
```



Spanningsdeler



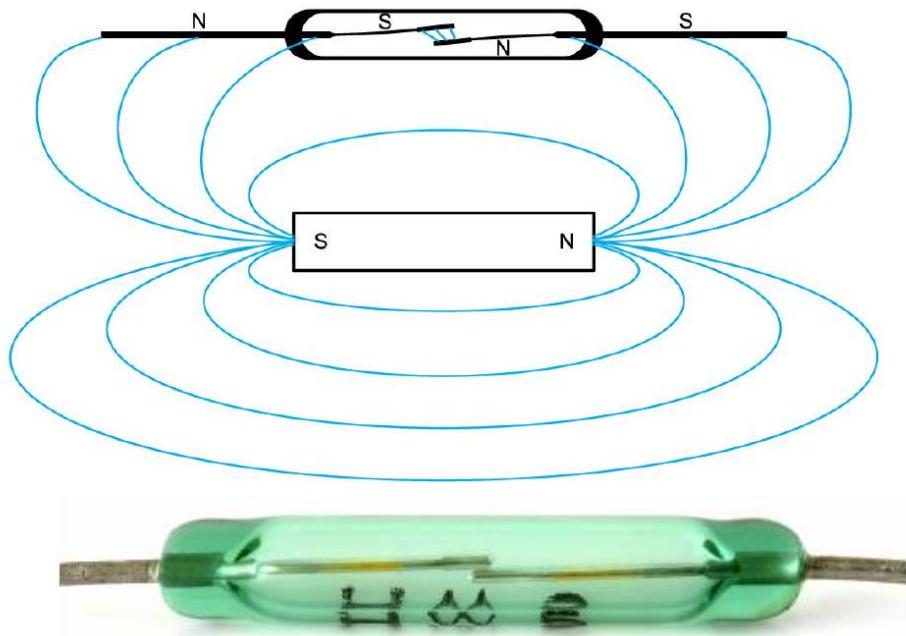
drukknop

{}

Magneetcontact

Een magneetcontact (ook wel magneetschakelaar of reedswitch genoemd) is een schakelaar die gevoelig is voor magnetisch veld. De schakelaar kent twee posities; hij is Open of Dicht, oftewel Hoogohmig of Laagohmig (zoals je misschien weet is Ohm de eenheid voor weerstand bij elektriciteit). Dat betekent dat er een open dan wel gesloten circuit is. De werking is niet anders dan de momentdrukknop. het contact is gesloten zolang het magneetveld aanwezig is.

Het magnetisch veld breng je aan in de vorm van een magneet. In onderstaande afbeelding zie je de werking van het magneetveld. Het contact zal zich sluiten door de aantrekkingskracht.



Magneetcontact.

Waar je bij de drukknop nog een tweede route voor de elektriciteit moet bieden, hoeft dat bij het magneetcontact niet. De ene kant kan aangesloten op de GND, en de andere op een digitale poort. Het nummer van die poort is dan in je Arduino code de locatie voor de input.

Thermometer (LM35)

De LM35 is vergelijkbaar met de TMP36 temperatuursensor. Je ziet hieronder een afbeelding van de drie aansluitingen die de sensor heeft: voedingsspanning, ground én het sensorsignaal.

Er zijn sensoren die voorzien zijn van een lange aansluitkabel. Rood gemerkte draad = +5Volt. Zwart gemerkte draad = GND. Draad zonder markering (of andere kleur) = Signaal. In de datasheet vind je de verhouding van het signaal t.o.v. de temperatuur. Bekijk ook de link naar een voorbeeldschakeling, waar je ook de code kunt zien. Er is een TMP36 sensor aangesloten en 3 LED's. Als je op simulatie starten klikt, kun je door op de sensor te klikken de temperatuur verhogen en zien dat de LED's bij hogere temperatuur één voor één aan gaan.



Tinkercad TMP36 [https://www.tinkercad.com/things/94VFdq4ienj-copy-of-tmp36-temperatur e-sensor-with-arduino/editel?tenant=circuits](https://www.tinkercad.com/things/94VFdq4ienj-copy-of-tmp36-temperatur-e-sensor-with-arduino/editel?tenant=circuits)

De code die je kunt gebruiken:

```
int baselineTemp = 0;
int celsius = 0;
int fahrenheit = 0;
void setup()
{
    pinMode(A0, INPUT);
    Serial.begin(9600);
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
}
void loop(){
    baselineTemp = 40;

    celsius = map((analogRead(A0) - 20) * 3.04, 0, 1023, -40, 125);
    fahrenheit = ((celsius * 9) / 5 + 32);
    Serial.print(celsius);
    Serial.print(" C, ");
    Serial.print(fahrenheit);
    Serial.printIn(" F");

    if (celsius < baselineTemp) {
        digitalWrite(2, LOW);
        digitalWrite(3, LOW);
        digitalWrite(4, LOW);
    }
    if (celsius >= baselineTemp && celsius < baselineTemp + 10) {
        digitalWrite(2, HIGH);
        digitalWrite(3, LOW);
        digitalWrite(4, LOW);
    }
    if (celsius >= baselineTemp + 10 && celsius < baselineTemp + 20) {
        digitalWrite(2, HIGH);
    }
}
```

```

    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
}
if (celsius >= baselineTemp + 20 && celsius < baselineTemp + 30) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
}
if (celsius >= baselineTemp + 30) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
}
delay(1);
}
}

```

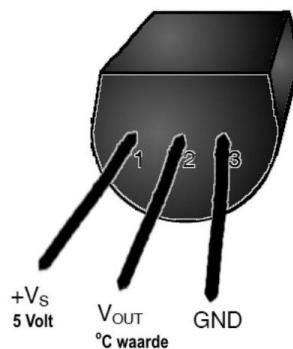
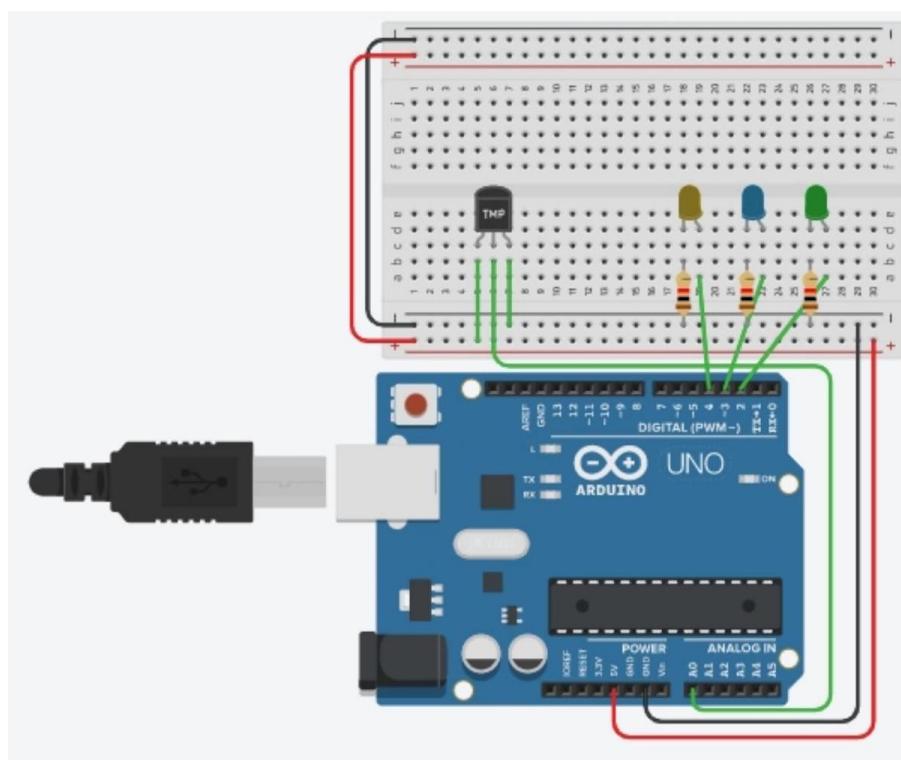


Figure 4: LM35 onderaanzicht



Figure 5: 4 LM35 Datasheet

Potentiometer

De potentiometer is een weerstand waarvan je de waarde kunt variëren. Wanneer je aan de as draait verandert de positie van een loper op de weerstandsbaan (vaak carbon). Met de draaiknop kun je de spanning aan de loper regelen. De werking hiervan en het aansluiten ervan wordt in de onderstaande video uitgelegd.

Aan de potmeter die wij gebruiken is een extra weerstandje toegevoegd om te voorkomen dat de potmeter verbrand bij verkeerd aansluiten (er zou dan kortsluiting kunnen ontstaan).

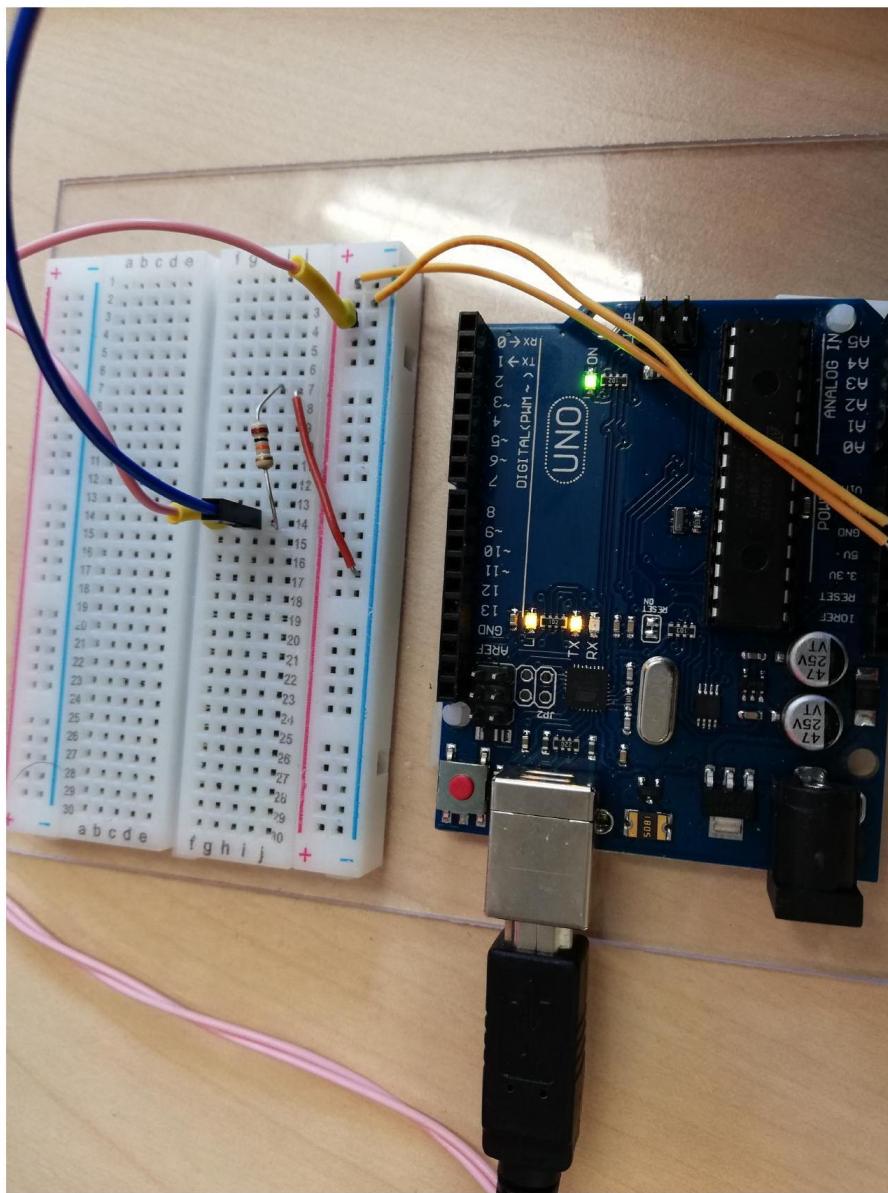
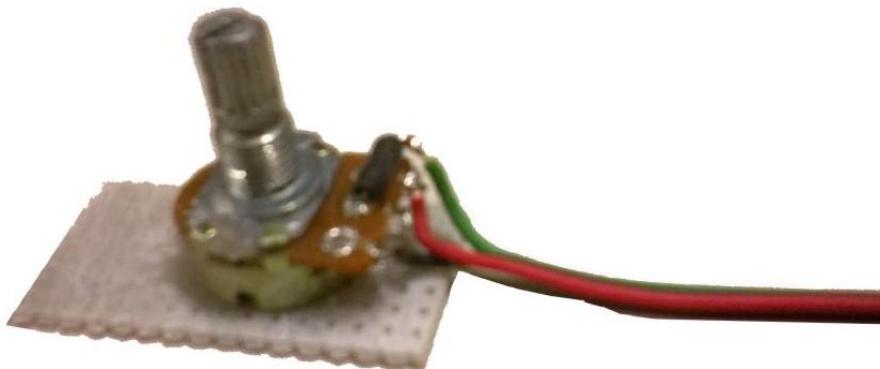


Figure 6: Opstelling



Aansluitingen

De groene draad is verbonden met de loper (via extra weerstandje van 2 kOhm). De rode is verbonden met de uiterste kant met de klok mee (maximum). De zwarte draad zit aan de nulpositie, de uiterste kant tegen de klok in (minimum).

Onze potmeter heeft een waarde van 10kOhm met de loper van 2kOhm.

Microfoon

Een microfoon is een sensor (of opnemer) die gevoelig is voor geluidsgolven. Wij maken gebruik van een module met het MAX9814 circuit met de volgende mogelijkheden:

- vast instellen van versterking
- automatische versterking met instelbare attack / release
- nauwkeurige opname in het frequentiegebied tussen 20 Hz en 20 kHz

Bekijk de datasheet van de MAX9814 voor een schat aan informatie over de mogelijkheden.

https://maken.wikiwijs.nl/userfiles/4/44fb5180baa209c6ad024ef2108610_8b1fb911c1.pdf

Hieronder zie je een aansluitschema voor de microfoon. Een microfoon is eigenlijk een vrij simpele analoge input, en dat zie je ook aan de code. Er wordt een analoge input uitgelezen en die wordt

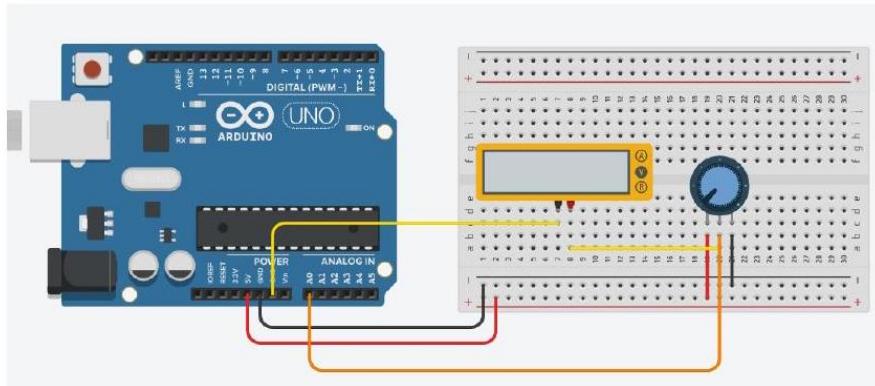


Figure 7: Aansluiting van de de potentiometer met willekeurige kleuren draad(!)

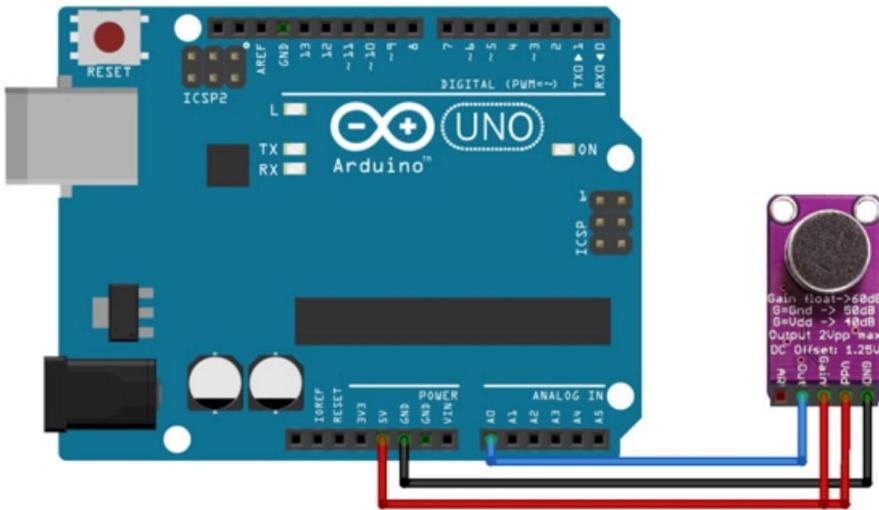


<https://maken.wikiwijs.nl/userfiles/4/44fb518018b1fb911c1.pdf>

Figure 8: Opstelling

in deze code geplot op de seriële plotter (bij hulpmiddelen, let op: de y-as wordt steeds automatisch geschaald). Let op: wat je hier ziet is de spanning die de microfoon afgeeft aan de Arduino, de golven die je ziet verschijnen zijn niet de geluidsgolven maar een elektronische representatie daarvan.

Als je echt de microfoon als input wilt gebruiken zul je nog een flinke bewerking met de gegevens moeten uitvoeren. Dat kan vrij complex zijn, en je moet wat weten over geluid. We zullen hier t.z.t. meer informatie over opnemen.



Een aan/uitschema voor de microfoon die wij gebruiken

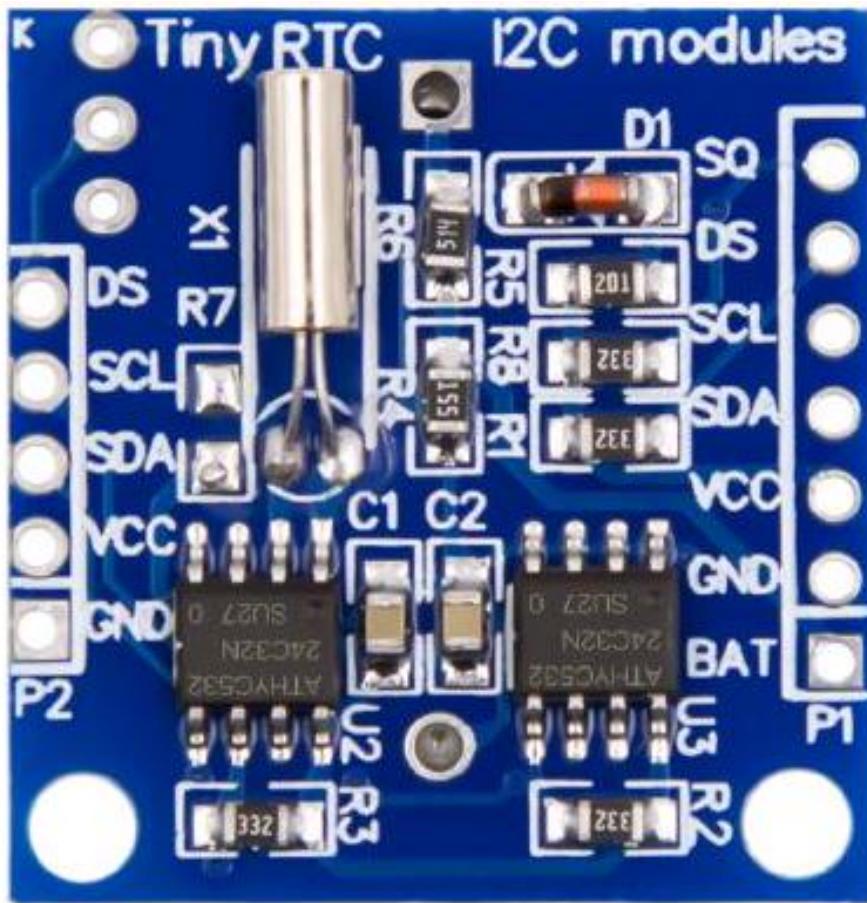
Klok

De Arduino beschikt over een ingebouwde klok die is aan te roepen met het commando `millis()`. Deze klok is handig om incidenteel een korte periode te meten, een wachttijd in te stellen of een teller op te hogen.

Wil je nauwkeurig een tijdseenheid kunnen toepassen en wil je daarin flexibeler zijn dan is het verstandig om een externe klok te gebruiken, een Real Time Clock.

Met zo'n klok beschik te over de werkelijke tijd en vaak ook de datum en zelfs een meerjarenkalender. Je leest hem uit op het moment dat je de gegevens nodig hebt zodat je je in de programmacode met belangrijker zaken kunt bezighouden. Met een ingebouwde batterij (knoopcel o.i.d.) zorg je ervoor dat de klok blijft werken als de Arduino is uitgeschakeld.

Wij werken met de DS1307 chip en communiceren met de module via het i2c protocol.



Ultrasone sensor

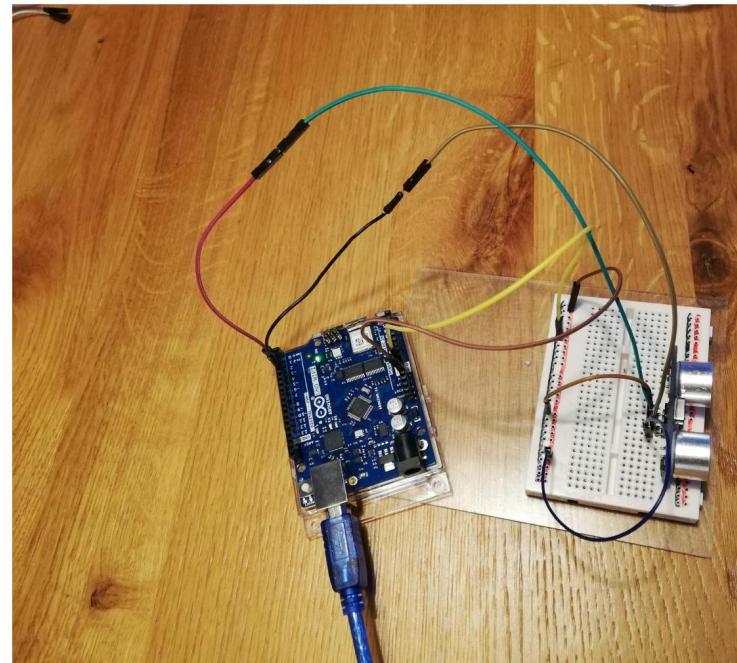
Met de ultrasone sensor kun je de afstand tot een object meten. Het is een apparaat dat erg eenvoudig in gebruik is. Op de onderstaande site kun je voor het type HC-SR04 lezen hoe je hem moet aansluiten en gebruiken. Dit is het type dat wij op school hebben. Er zijn ook andere typen. Je ziet ook een filmpje hieronder met de ultrasone sensor in gebruik en een foto van hoe het er in de praktijk uit ziet.





Ultrasone sensor

Tutorial voor de ultrasone sensor <https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6>



wikiwijs

Een voorbeeld van hoe een aangesloten ultrasone sensor eruit ziet

```
int trigPin = 9; // TRIG pin Serial.begin (9600); void loop() {
digitalWrite(trigPin, HIGH);
duration_us = pulseIn(echoPin, HIGH);
delay(500);
```

Vochtsensor

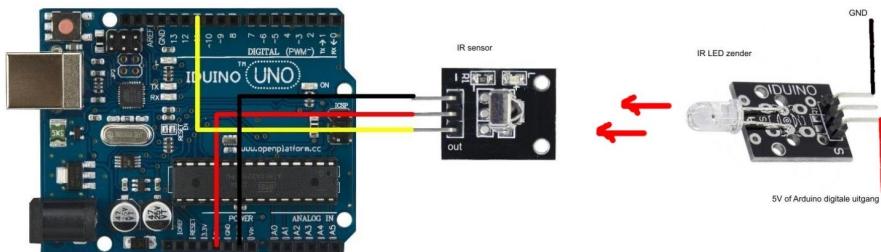
De vochtsensor is een sensor die gebaseerd is op de geleiding van stroom door (normaal) water. Als er meer water is, gaat die geleiding makkelijker, is het droog gaat de geleiding moeilijker.

Lichtsluis

Een lichtsluis is een hulpmiddel om aanwezigheid van objecten te kunnen detecteren. Een voorbeeld van een hele grote lichtsluis vind je bij een liftdeur. In de deuropening zit aan de ene zijde een lichtbron

(al dan niet met sensor) en aan de andere zijde een reflector of een lichtsensor. Zodra een persoon zich tussen lichtbron en sensor bevindt zal de deur niet sluiten. Het licht is meestal onzichtbaar (infrarood).

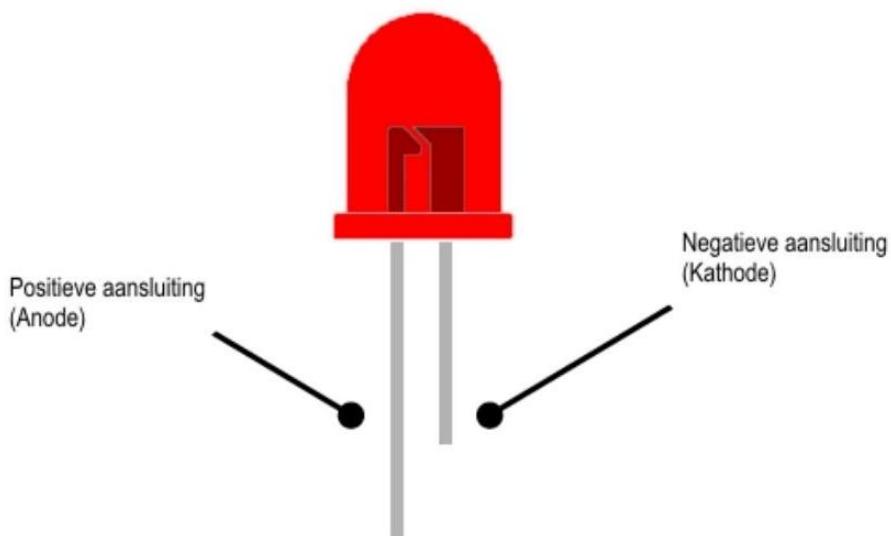
In het aansluitvoorbeeld zie je hoe je een lichtsluis kunt toepassen i.c.m. een Arduino.



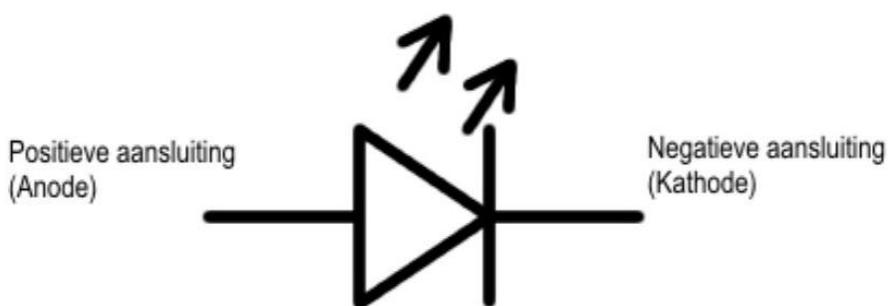
0.37.5 Hardware voor de output van een digitaal systeem

LED

Een LED is een diode die licht kan uitzenden. De afkorting staat voor Light Emitting Diode. Een diode is een elektronische component, een zogenaamde halfgeleider.



De LED wordt geleidend en gaat licht uitzenden zodra er een stroom in de juiste richting gaat lopen: de LED heeft een positieve en een negatieve aansluiting. Als de LED verkeerd om wordt gemonteerd loopt er geen stroom, de LED staat dan in sperstand.



Schematisch Symbool van de LED

Waar moet je op letten bij gebruik van een LED?

- Een LED is kwetsbaar, overschrijd de maximum stroom niet! Gebruik daarom ALTIJD een voorschakelweerstand in serie met de LED. Deze zal de stroom begrenzen op de gewenste waarde en de grootte van de benodigde weerstand is afhankelijk van aangeboden spanning. Het maakt niet uit of je de weerstand in de pluslijn of in de minlijn plaatst.

Als vuistregel mag je de volgende waarden hanteren:

- Aangeboden spanning 5 Volt d.c. ->voorschakelweerstand 150 Ohm.
- Aangeboden spanning 12 Volt d.c. ->voorschakelweerstand 560 Ohm.

Soms krijg je een LED waarbij al een weerstand in serie is gemonteerd, zie de foto voor een voorbeeld hiervan.

weerstand met waarde 560 Ohm

LED RGB

Neopixel LEDstrip

Neopixel LEDs zijn fantastisch. Het zijn LEDjes die alle kleuren en intensiteit kunnen aannemen en je kunt heel makkelijk een heleboel LEDs aansturen met een paar draadjes en wat slimme code.

Om Neopixels te gebruiken moet je een library installeren op je laptop. In die library zitten een heleboel commandos die speciaal zijn gemaakt voor de Neopixel LEDs. Hoe je de library installeert zie



je op onderstaande website.

Arduino tutorial neopixel <https://create.arduino.cc/projecthub/robocircuits/neopixel-tutorial-1ccfb9>

Op de bovenstaande site vind je ook het schema hoe je de Neopixel aansluit en hoe je met je code de verschillende ‘adresseerbare’ LEDs aan of uit zet en welke kleur en intensiteit ze moeten aannemen.

Met adresseerbaar bedoelen we dat je met de code verschillende LEDs kunt aansturen: pixels.setPixelColor(i, pixels.Color(0, 150, 0)); Met bovenstaande code wordt nr i aangezet met een kleur die beschreven wordt met het pixels.Color commando. De 0,150,0 staat voor de intensiteit van de kleuren Rood Groen en Blauw (RGB). Dit zal dus een LED een milde groene kleur laten aannemen. De intensiteit kan van 0 tot 255 lopen.

Dus met: pixels.setPixelColor(8, pixels.Color(255, 0, 0)); zetten we LED nr 8 op de strip aan met een felrode kleur. Hieronder zie je een voorbeeld van een code waarmee je een Neopixel bestuurt. Dit betreft een Neopixel van 10 LED’s, maar ook voor een Neopixel van 100 LED’s kun je deze code gebruiken (alleen het getal 10 vervangen door 100).

```
#include <Adafruit_NeoPixel.h>
#ifndef _AVR
#include <avr/power.h>
#endif
#define Neopixel 7

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, Neopixel, NEO_GRB + NEO_KHZ800);
int delayval = 50;
```

```
void setup() {
    pixels.begin();
}
void loop() {
    for(int i=0;i<NUMPIXELS;i++){
        pixels.setPixelColor(i, pixels.Color(0,150,0));
        pixels.show();
        delay(delayval);
    }
    for(int i=0;i<NUMPIXELS;i++){
        pixels.setPixelColor(i, pixels.Color(0,0,0));
        pixels.show();
        delay(1);
    }
}
```



Neopixel voorbeeld

LCD display

De processor (Arduino) in jullie ontwerp gaat acties uitvoeren aan de hand van signalen of gegevens die je laat invoeren (input). Is het op enig moment nodig om visueel informatie aan de buitenwereld kenbaar te maken dan kan dat met een eenvoudig lichtsignaal. Maar soms is de hoeveelheid informatie zo groot dat er meer nodig is. De uitkomst is dan een beeldscherm. Jullie kunnen daarvoor een LCD gebruiken. LCD staat voor Liquid Crystal Display. Dat is een stapeltje glasplaatjes waartussen vloeibare kristallen kunnen bewegen. De richting waarin de kristallen zich richten verandert door een kleine lading aan te brengen. Daardoor ontstaat een polariserend filter dat licht blokkeert. Een goed voorbeeld is je rekenmachine of een kwartshorloge; je ziet op dat schermpje cijfers verschijnen in

zwart/wit.

Praktisch

De LCD die jullie kunnen gebruiken bevat 4 regels (20x04) waarop karakters kunnen worden getoond. Schrijven gebeurt via je programmacode en het is nodig dat je telkens aangeeft op welke regel en welke positie je begint met schrijven. Omdat een standaard LCD wordt aangestuurd met een behoorlijk aantal signalen is er een I₂C circuit toegevoegd die dat voor zijn rekening neemt. Daardoor zijn er voor het opbouwen van het display slechts vier draden nodig: Voedingsspanning, Ground, SCL kloklijn en SDA datalijn.

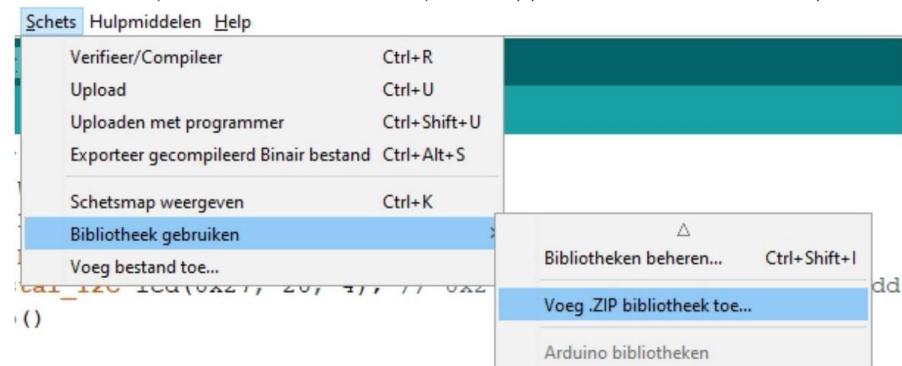
Hieronder vind je een code die werkt. Het blijkt in de praktijk wel tricky te zijn om zo'n display werkend te krijgen, er staan namelijk een hoop voorbeelden op het web waar fouten in zitten. Daarnaast heb je twee libraries nodig. Een library is een collectie van programmeercommando's die samen helpen om bepaalde taken uit te voeren. De libraries die je hier nodig hebt zijn wire.h en LiquidCrystal_I2c.h.

Wire.h is standaard geïnstalleerd, maar die andere niet. Via de link hieronder vind je een zip bestand. In de Arduino omgeving kun je zo'n bibliotheek inladen vanuit een zip bestand (zie screenshot hieronder).

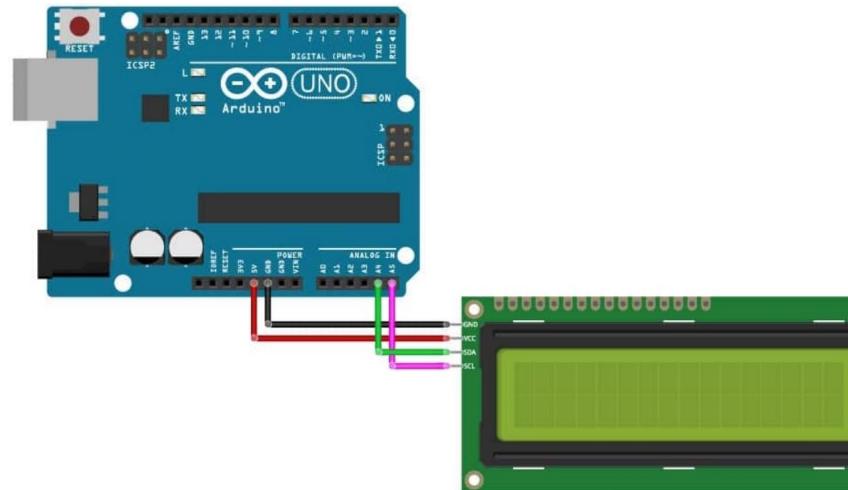


Een "kale" display met veel aansluitingen.

link naar liquidcrystal_i2c (datum link: 7-2-2022) <https://downloads.arduino.cc/libraries/github.com/marcosc...>



ystal I2C-1.1.2.zip



Toevoegen van een zip-bibliotheek (library)

Display voorzien van i2c interface: aansluitvoorbeeld.

Een werkende code

Onderstaande code start de display, en laat de tekst 'hello world' zien. Daarna start er een teller die vrij rap oploopt (het geprinte 'nummer' wordt in elke loop met 1 opgehoogd).

```
int nummer = 0;
#include "Wire.h" // For I2C
#include "LiquidCrystal_I2C.h" // Added library*
//Set the pins on the I2C chip used for LCD connections
LiquidCrystal_I2C Icd(0x27, 20, 4); // 0x27 is the default I2C bus address
void setup()
{
// Set off LCD module
Icd.begin (20, 4); // 20 x 4 LCD module
// Turn on the blacklight and print a message.
//lcd.setBacklight(3,POSITIVE); // BL, BL_POL
Icd.setBacklight(HIGH);
Icd.setCursor(0, 3);
Icd.print("Hello, world!");
}
void loop()
{
nummer = nummer+1;
Icd.setCursor(2,2);
Icd.print(nummer);
}
```

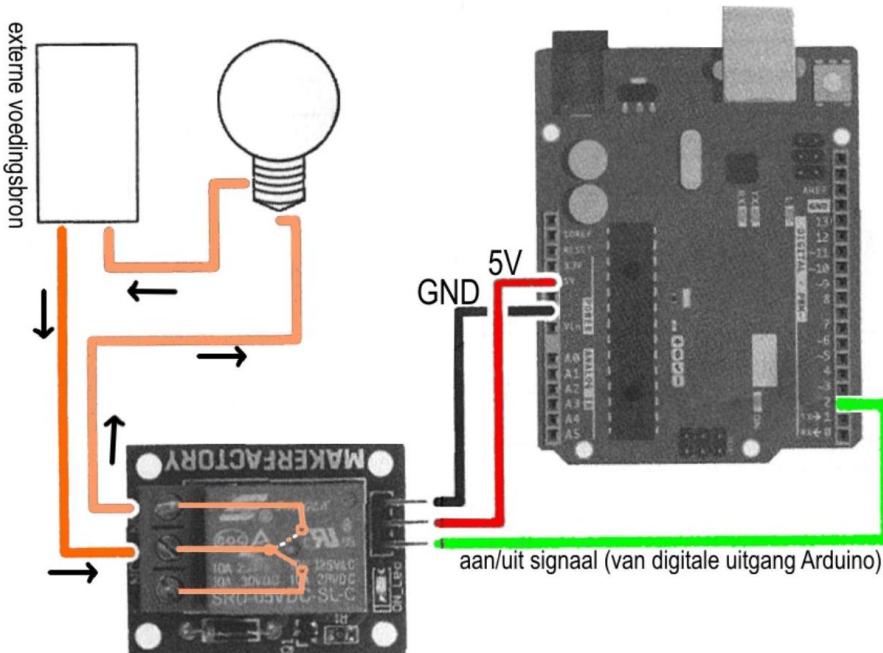
Let op: bij mijn laptop blijkt ik de LCD display te moeten aansluiten op de 3,3V aansluiting ipv de 5V. Ook kan er een probleem ontstaan als je een verouderde versie van Arduino op je laptop hebt staan.

Relais

Relais

Een relais is een schakelaar om actuators te kunnen inschakelen en uitschakelen.

Je gebruikt een relais wanneer de Arduino zelf niet genoeg vermogen kan leveren voor de actuator. Ook kun je een relais gebruiken om een circuit galvanisch gescheiden van de Arduino te kunnen schakelen, bijvoorbeeld wanneer een element op netspanning werkt, of op een andere spanning dan 5



Volt.

Aansluiten

Een relaismodule is beschikbaar met één relais of met meerdere relais. De module wordt bediend met een digitaal stuursignaal vanaf een uitgangspin van de Arduino (HOOG / LAAG ->5V / 0V).

Omdat het relais bekraftigd moet worden moet je via een extra draad 5 Volt voeding aanbieden. Dat kan via de 5 V pin van de Arduino. Let wel op dat deze pin begrensd is in maximale stroomcapaciteit. Gebruik je dus meerdere relais of gelijktijdig ook andere verbruikers op 5 Volt, dan kun je die beter rechtstreeks voeden uit een externe 5 Volt bron.

De relais schakelaar wordt magnetisch bekraftigd en heeft in ons geval 3 aansluitpunten. Het is een wisselschakelaar. In onbekraftigde toestand (in rust) is het moedercontact (middelste aansluitpunt) geschakeld aan contactpunt N.C. (Normally Closed). Zet je het relais "aan" dan schakelt het moedercontact over naar contactpunt N.O. (Normally Open). Op de printplaat zijn deze punten herkenbaar gemerkt.

Pulse Width Modulation PWM

Elektronische schakelaar in plaats van mechanisch Relais

In bepaalde situaties wil je een actuator (lamp / motor / klep o.i.d.) schakelen. Dat kan vaak met een relais (elektromagnetisch bekraftigde schakelaar).

Maar er zijn toepassingen waarbij het niet met een relais kan. Bijvoorbeeld het trapsgewijs schakelen naar een hoger vermogen lukt niet met een mechanische schakelaar, die staat of AAN óf UIT, ALLES of NIJS.

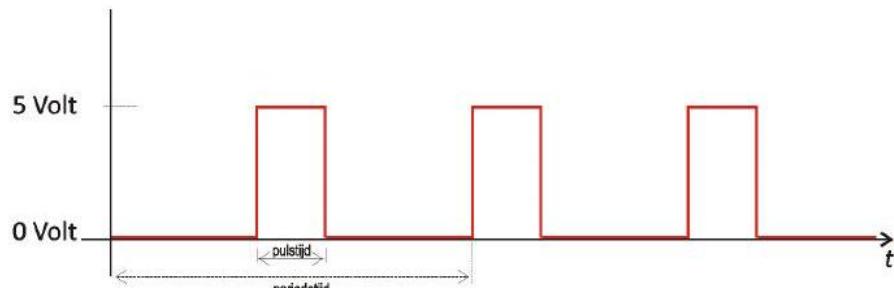
Een elektronische schakelaar kan veel sneller schakelen, bevat geen mechanische onderdelen en is dus heel geschikt om middels korte pulsen te schakelen.

Dat geeft nieuwe mogelijkheden! Door met korte pulsen de schakelaar in- en uit te schakelen kan het gemiddelde vermogen geregeld worden. Zo'n pulserend regelsignaal heet pulsbreedtemodulatie, oftewel Pulse Width Modulation PWM.

Door bijvoorbeeld iedere seconde afwisselend aan- en uit te schakelen wordt er het halve vermogen toegevoerd, want 50% van de tijd is de schakelaar AAN en 50% van de tijd is deze UIT. Men zegt dan: de Duty cycle is 50%.

Maar: iedere halve seconde is een erg lage frequentie. Dan zou een lamp bijvoorbeeld erg vervelend knipperen, wel handig als knipperlicht, niet handig als je de lamp juist zou willen dimmen.

Dus, een bruikbaar PWM signaal heeft een veel hogere frequentie, want dan krijg je een heel goed



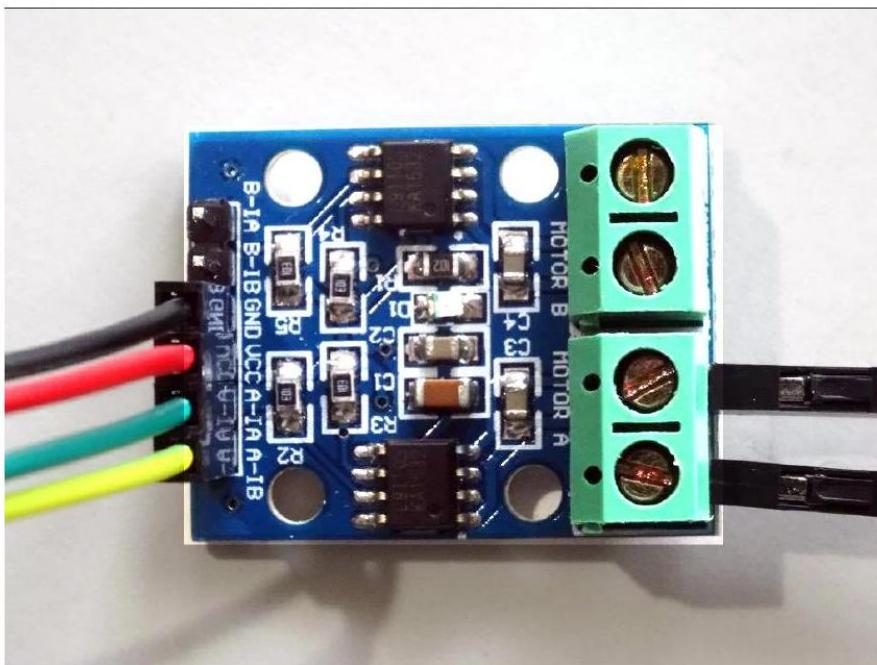
regelbaar signaal.

Hiernaast zie je een PWM signaal. Een digitale uitgang van de Arduino wordt op een vaste frequentie geschakeld tussen laag en hoog. De verhouding tussen de duur van laag en hoog bepaal je zelf. De duur van de puls (hoog) noem je de PulsBreedte (PulseWidth). In dit geval is de Dutycycle 33%.

PWM poorten

De Arduino Uno bezit een paar poorten die je kunt gebruiken om een PWM signaal te genereren. Deze poorten werken automatisch op commando en staan op een vaste frequentie.

Arduino Uno pin 3, 5, 6, 9, 10 en 11 kunnen als PWM uitgang gebruikt worden. (Pin 5 en 6 werken met een 980 Hz frequentie. Poorten 3, 9, 10 en 11 werken met 490 Hz.)



Solid State Switch (Elektronische Schakelaar) om 2 motoren te regelen.

Code voorbeeld Arduino UNO

Met het commando analogWrite(naam,waarde) geef je de opdracht om het signaal aan de uitgangspoort beschikbaar te maken. naam Is de naam die je aan de poort hebt gegeven. Let er op dat niet alle uitgangen van de Arduino geschikt zijn voor een PWM signaal. waarde Is de gemiddelde waarde die je wilt dat het signaal heeft. Deze is instelbaar tussen 0% (UIT) en 100% (AAN).

formule: Duty cycle = (waarde / 255) [%]

Duty cycle 0% = 0

Duty cycle 50% = 127

Duty cycle 100% = 255

onderstaand een voorbeeldcode voor de combinatie Arduino / Chip L9110 / DC motor Het IC L9110 heeft 2 ingangen waarmee je 2 uitgangen schakelt (de twee draden die naar de motor gaan). Het IC schakelt volgens een waarheidstabell: inputA inputB outputA outputB motor 0000 uit

1010 rechtsom

0101 linksom

1111 uit

```
// pwm analogWrite op pin D05 490Hz
// pwm analogWrite op pin D06 490Hz
const int LINKS = 5; // motor pwm signaal op pin D04
const int RECHTS = 6; // motor pwm signaal op pin D05
void setup()
\{
pinMode(LINKS, OUTPUT); // pinMode(RECHTS, OUTPUT); //
\}
void loop()
\{
analogWrite(LINKS,0);
analogWrite(RECHTS,0);
delay(500); // wacht xx ms
analogWrite(RECHTS,125);
delay(2000); // wacht xx ms
analogWrite(RECHTS,200);
delay(2000); // wacht xx ms
analogWrite(RECHTS,255);
delay(2000); // wacht xx ms
analogWrite(LINKS,0);
analogWrite(RECHTS,0);
delay(500); // wacht xx ms
```

Ventilator

DC motor (lineaire motor)

DC motor

Een standaard DC motor (gelijkstroom elektromotor) wordt toegepast in allerlei soorten aandrijvingen. Denk aan een accuboormachine of een polijstmachientje bij de pedicure bijvoorbeeld.

De motor schakel je niet rechtstreeks vanuit de poort van de Arduino maar je gebruikt tussen Arduino en motor een relais of een solid-state-switch. Hiermee schakel je dan een externe voeding



aan- of uit.

Voorbeeldschakeling met Arduino, Relais en Lamp. https://maken.wikiwijs.nl/152360/NLT__Digitale_Techniek
7105 936

Linksom of Rechtsom?

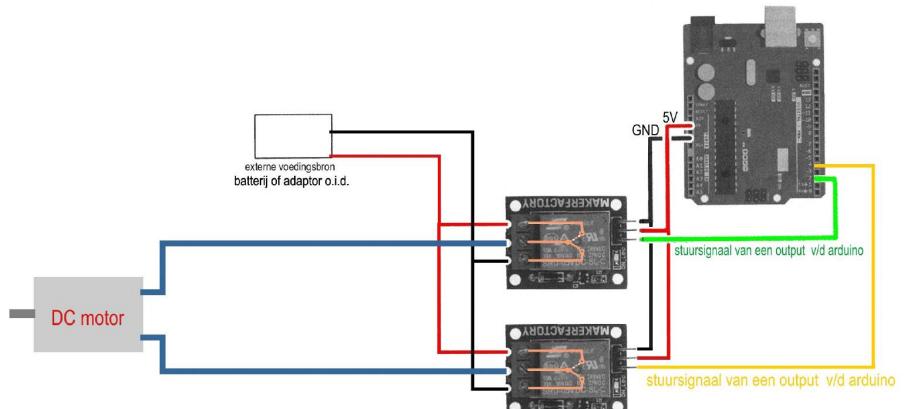
De dc-motor draait altijd in één richting, afhankelijk van de polariteit (+ pool en - pool) van de aangeboden voedingsspanning. Draai je de polariteit om, dan gaat de motor in de andere richting draaien.

Door het toepassen van twee relais wordt het mogelijk om de draairichting van de motor te kiezen. Zie onderstaande voorbeeldschakeling. Naast de twee relais heb je uiteraard ook een externe voeding nodig én je gebruikt een extra uitgang van de Arduino.

Hoe werkt het?

In rust staan beide relais in de ruststand. De motor is dan met beide aansluitingen verbonden met de pool van de voeding.

Er gebeurt dan dus niks. De truck zit hem in het afzonderlijk aansturen van één van de relais. Dan krijgt de motor energie toegevoerd, via het ene relais zal de motor met de klok mee draaien, met het

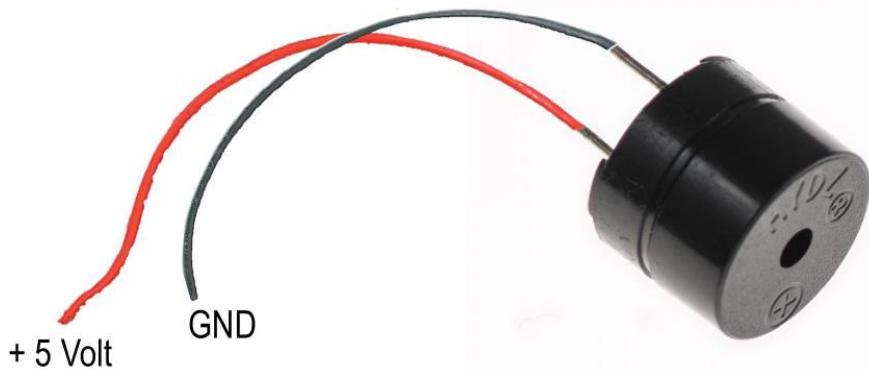


andere relais tegen de klok in!

Stappenumotor**Pomp****Buzzer**

De buzzer is een soort luidspreker. Deze kan een geluid produceren. Vaak een pieptoon of een zoemtoon, die ontstaat doordat een membraan gaat trillen onder invloed van een spanning of stroom.

De buzzer die hier als voorbeeld is afgebeeld kan direct worden aangestuurd door een digitale uitgang van de Arduino, 5 Volt gelijkspanning, dus let op de positieve en negatieve (GND) aansluiting.



Miniatuurbuzzer

Elektromagneet

0.37.6 Hardware voor de opslag en het transport van data

Draadjes (en ze bevestigen)

i2c

Sensoren en Actuatoren slimmer aansluiten in de praktijk

Wanneer je al wat schakelingen hebt bedacht en gebouwd aan de hand van de oefenopdrachten, zul je gemerkt hebben dat deze kunnen uitmonden in een wirwar aan verbindingen en draadjes. Vooral als je wat meer met de Arduino gaat doen dan alleen een temperatuur meten of een lampje aan- of uitzetten.

Zodra je project groeit krijg je te maken met ingewikkelder sensoren, meer ingangssignalen en ook meer uitvoer.

Voor weergave van gegevens kun je bijvoorbeeld gebruik maken van een display. Voor opslag op een geheugenkaart zijn er mooie insteekmodules. Typisch onderdelen die veel verschillende signalen nodig hebben om te kunnen werken.

Om je project overzichtelijk te houden, slimmer op te zetten, beter te kunnen onderhouden en makkelijker te kunnen ontwerpen ga je gebruik maken van een communicatieprotocol. Hiermee zijn een heleboel moeilijke ontwikkelstappen al gedaan en kun je op relatief eenvoudige manier grote slagen gaan maken.

Het communicatieprotocol waarvoor de Arduino al is voorbereid is het i2c protocol (spreek uit als "i kwadraat c").

Dit systeem werkt met maar 2 signaallijnen om gegevens uit te wisselen tussen processor en sensor of actuator. Samen noemen we deze twee draden de "bus". In dit geval dus de i2c bus. i 2 c is een seriële bus, omdat alle informatie in blokken van een aantal bits achter elkaar wordt getransporteerd. Eén draad is de referentie en voert een kloksignaal, de kloklijn: SCL (Serial CLock). De andere draad voert de informatie, de data en heet de datalijn: SDA (Serial DAta). Beide lijnen kennen twee niveaus, een logische "1" en een logische "0".

Een heel groot praktisch voordeel van deze communicatiebus is dat er een hele grote hoeveelheid verschillende sensoren en actuatoren parallel aan deze twee draden kunnen worden aangesloten. Dat scheelt enorm veel werk en bespaart veel hardware (bedrading).

Het protocol zorgt er voor dat alle aangesloten eenheden herkenbaar zijn aan een adres. Bij i2c bestaat elk adres uit 7 bits (een getal bestaande uit 7 nullen en/of enen). Met deze methode kun je dus maar liefst 128 sensoren/actuatoren verbinden met de Arduino en dat met maar twee draden! (Ga na of je begrijpt waarom er maximaal 128 unieke adressen bestaan bij i2c).

Geschikt voor i2c ?

Niet alle randapparaten die wij beschikbaar hebben zijn voorbereid voor i2c. Let hier op door van elke sensor of actuator die je wilt gebruiken bijbehorende informatie te zoeken! (uit de datasheet of andere bronnen).

Master en Slave

wikiwijs i2c werkt op basis van een Master-Slave constructie. De processor (Arduino) is de baas, de Master. De sensor of actuator is de slaaf (Slave). Dat houdt in dat een sensor of actuator pas actie onderneemt nadat de master daarvoor de opdracht heeft gegeven. Het zou anders een onbegrijpelijke brei van signalen worden op de bus.

Pull-Up

De beide buslijnen hebben in de ruststand een hoog niveau: logische "1". (5 Volt of 3,3 Volt, afhankelijk van de gebruikte voedingsspanning). Daarom moeten er altijd pull-up weerstanden worden gebruikt tussen de buslijnen en de voeding. Deze weerstanden plaats je bij een fysiek korte bus dicht bij de Arduino (je kunt ze ook via de code programmeren, dan hoef je geen externe weerstanden te paatsen).

Als je een langere bus gebruikt (meerdere meters) óf deze zich bevindt in een omgeving met veel ruis (sterke stoorsignalen) dan plaats je meerdere weerstanden, verspreid over de totale lengte.

Als je I2C apparatuur aan gaat sluiten krijgt elk apparaat een uniek adres. Dit is nodig omdat alle info naar alle aangesloten apparaten via hetzelfde draadje moet. In je code moet je dan dus aangeven van elk aangesloten I2C apparaat wat zijn adres is. Maar hoe weet je dan het adres? Met onderstaande code scan je je systeem voor alle adressen van aangesloten I2C apparatuur. Voor een LCD display is het bijv. 0x27.

```
/  
// Arduino I2C Scanner  
// Re -writed by Arbi Abdul Jabbaar  
// Using Arduino IDE 1.8.7  
// Using GY -87 module for the target  
// Tested on 10 September 2019  
// This sketch tests the standard 7 -bit addresses  
// Devices with higher bit address might not be seen properly.  
-----  
#include <Wire.h> //include Wire.h library  
void setup()  
{  
    Wire.begin(); // Wire communication begin  
    Serial.begin(9600); // The baudrate of Serial monitor is set in 9600  
    while (!Serial); // Waiting for Serial Monitor  
    Serial.println("\nI2C Scanner");  
}  
void loop()  
{  
    byte error, address; //variable for error and I2C address  
    int nDevices;  
    Serial.printIn("Scanning...");  
    nDevices = 0;  
    for (address = 1; address < 127; address++)  
    {  
        // The i2c_scanner uses the return value of  
        // the Write.endTransmisstion to see if  
        // a device did acknowledge to the address.  
        Wire.beginTransmission(address);  
        error = Wire.endTransmission();  
        if (error == 0)  
        {  
            Serial.print("I2C device found at address 0x");  
            if (address < 16)  
                Serial.print("0");  
  
            Serial.print(address, HEX);  
            Serial.printIn(" !");  
            nDevices++;  
        }  
        else if (error == 4)  
        {  
            Serial.print("Unknown error at address 0x");  
            if (address < 16)  
                Serial.print("0");  
            Serial.printIn(address, HEX);  
            if (nDevices == 0)
```

```

Serial.println("No I2C devices found\n");
else
Serial.println("done\n");

delay(5000); // wait 5 seconds for the next I2C scan
\}

```

(micro) SD card reader

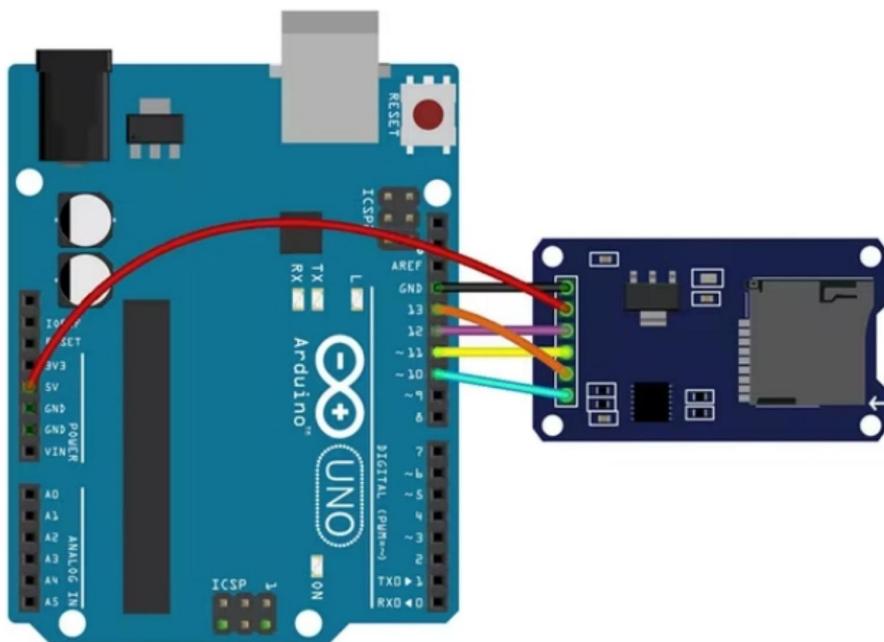
Als je systeem gegevens verzamelt wil je die bewaren. De makkelijkste manier om dit te doen is door het weg te schrijven naar een (micro)SD kaart. Deze zijn makkelijk aan te sluiten op de Arduino en met een cardreader kun je vervolgens de gegevens op de computer bekijken.

Het vergt wel wat voorbereiding. De gegevens schrijf je namelijk weg, zoals je gegevens naar de seriële monitor schrijft. Op de SD kaart wordt een tekstbestandje opgeslagen met daarin de tekst die je naar dat bestand toeschrijft.

Stel dat je een reeks metingen wilt wegschrijven is het bijv. belangrijk om elke meting steeds op een nieuwe regel te beginnen, en andere informatie die belangrijk is voor de meting (bijv. het tijdstip) op dezelfde regel op te schrijven. Verschillende soorten informatie kun je scheiden door spaties, maar beter is het om dit te doen door puntkomma's. Je kunt dan in Excel het bestand openen en de kolommen scheiden op basis van puntkomma's (wordt uitgelegd).

Het goed gebruiken van de SD kaart is een van de pittigste onderdelen van je systeem, maar het is het wel het onderdeel dat het systeem meer maakt dan een dom apparaatje. Nu kun je er namelijk onderzoek mee doen en daarmee je apparaat veel nauwkeuriger evalueren, wat een belangrijke stap is in de ontwerpcyclus.

Daarnaast is de code voor het wegschrijven naar de SD kaart best complex (zie onder). Het vergt enige oefening om al je andere code op de juiste plek in de code neer te zetten waardoor alle verschillende onderdelen samen werken. In hoofdstuk 8 ga je hier vanzelf een voorbeeld van zien. Omdat jullie allemaal iets anders gaan maken, zal voor iedereen de uiteindelijke oplossing anders zijn.



Aan/uitschema voor een micro-sd cardreader

```

#include <SPI.h>
#include <SD.h>
File myFile;
void setup() {
    // Open serial communications and wait for port to open:
    Serial.begin(9600);

```

```
while (!Serial){  
; // wait for serial port to connect. Needed for native USB port only  
}  
Serial.print("Initializing SD card...");  
if (!SD.begin(10)) {  
    Serial.printIn("initialization failed!");  
    while (1);  
}  
Serial.printIn("initialization done.");  
// open the file. note that only one file can be open at a time, // so you have to close it  
myFile = SD.open("test.txt", FILE_WRITE);  
// if the file opened okay, write to it:  
if (myFile) {  
    Serial.print("Writing to test.txt...");  
    myFile.println("This is a test file :)");  
    myFile.printIn("testing 1, 2, 3.");  
    for (int i = 0; i < 20; i++) {  
        myFile.printIn(i);  
    }  
    // close the file:  
    myFile.close();  
    Serial.printIn("done.");  
}  
else {  
    // if the file didn't open, print an error:  
    Serial.println("error opening test.txt");  
}  
}  
void loop() {  
// nothing happens after setup  
}
```

Wifi chip

0.37.7 Voeding voor een digitaal systeem

Arduino, sensoren en actuatoren voeden

USB stekker (5v)

Digitale systemen instructies geven (programmeren)

Programmeren

Een Arduino code noemen we een sketch. Deze sketch wordt omgezet naar computertaal in enen en nullen (de binaire code, zie hierboven). Wij hoeven ons niet druk te maken over de binaire code, maar moeten wel weten hoe we zo'n sketch schrijven.

Een sketch maakt gebruik van objecten. Wat zijn objecten? Een object is iets dat je in een code gedefinieerd hebt en dat een geheugenplaats inneemt. Als je bijvoorbeeld in je code iets wilt doen met een LED (aanzetten of uitzetten) moet je wel tegen die Arduino vertellen dat er zoiets als een LED is. Dat slaat de Arduino dan op in zijn geheugen zodat hij op een later moment weet wat jii bedoelt met het woordje LED. Het woordje LED snapt de Arduino niet, dus je moet hem vertellen dat het woordje LED iets te maken heeft met de poort waar de LED op aangesloten is, en dat dit een OUTPUT is. Er komt namelijk licht uit. Hier komt de structuur van een sketch om de hoek kijken.

De structuur van een code ziet er globaal zo uit:

1. <ruimte voor definiëren van objecten >
2. <ruimte voor het duidelijk maken wat de objecten zijn, dit wordt eenmalig uitgevoerd >
3. <ruimte waar verteld wordt wat er met de objecten moet gebeuren, dit wordt steeds opnieuw uitgevoerd >

Het eerste gedeelte heeft geen naam, het 2^e gedeelte noemen we de void setup, het 3^e gedeelte noemen we void loop. We gaan dit verduidelijken met behulp van onderstaande code:

```
const int LMMP = 8;
const int KNOP = 2;
void setup()
{
    pinMode(LMMP, OUTPUT);
    pinMode(KNOP, INPUT);
}
void loop()
{
    int KnopStatus = digitalRead(KNOP);
    if(KnopStatus==1) {
        digitalWrite(LMMP, HIGH);
    }
    else {
        digitalWrite(LAMP, LOW);
    }
    delay(100);
}
```

Deel 1 - Objecten definiëren

Je ziet in regel 1 en 2 dat er wordt verteld dat de LAMP gelijk is aan 8 en de KNOP gelijk is aan 2 . Aan het begin van de regel staat het woordje const, dat betekent dat de waarde van dit object niet kan veranderen tijdens het programma. Stel dat je een plekje in het geheugen wilt inruimen voor iets dat wel kan veranderen, bijvoorbeeld een object waarin de waarde van een thermometer steeds

opnieuw opslaat, dan moet er natuurlijk geen const voor staan. We komen later terug op het woordje int.

Deel 2 - Void setup

In het tweede gedeelte zie je het commando pinMode. Dat is een commando waarmee je de Arduino vertelt wat voor soort apparaat is aangesloten op de pin (er zitten genummerde pins op de Arduino). Je ziet dat het object LAMP een OUTPUT is, dat is logisch: er komt licht uit. Je ziet ook dat KNOP een INPUT is. Dat is ook logisch: een knop verwerkt een invloed van buitenaf (jij drukt wel of niet op de knop). Er zijn nog veel meer commando's die in void setup gebruikt kunnen worden, maar die volgen later. Je ziet ook dat er na het commando void setup een openingsaccoade staat { en aan het einde van void setup een sluitingsaccoade }. Dat betekent dat alles wat tussen die accolades staat bij void setup hoort.

Commando's waarbij accolades gebruikt worden (zoals bij void setup, maar ook bij if en ifelse) hebben geen ; aan het einde van de zin, alle andere zinnen wel. Als je die ; niet plaats denkt de computer dat je zin nog niet af is en krijg je in de volgende regel een foutmelding. Eerste tip bij foutmeldingen is dus even te checken of je overall ; hebt staan waar dat hoort.

Deel 3 - Void loop

In regel 12 wordt een object gedefinieerd met de naam KnopStatus. Je kunt ook in dit gedeelte van de code objecten definiëren, dat gebeurt dan steeds opnieuw omdat de void loop steeds opnieuw wordt uitgevoerd. Stel dat je in regel 3 (in het eerste gedeelte) KnopStatus al zou definiëren zou deze regel hetzelfde zijn, maar dan zonder int ervoor (dus: KnopStatus = analogRead(KNOP);). In deze regel wordt met het commando analogRead de waarde uitgelezen van de KNOP. Het systeem checkt dus of de knop ingedrukt wordt of niet. Als dat zo is, geeft analogRead een 1, anders een 0 . Die 1 of 0 wordt met deze regel opgeslagen in het object KnopStatus en dat gebeurt dus steeds opnieuw. Omdat dit een hele kort void loop is, gaat het heel snel en wordt er dus eigenlijk continu gekeken of de knop wordt ingedrukt.

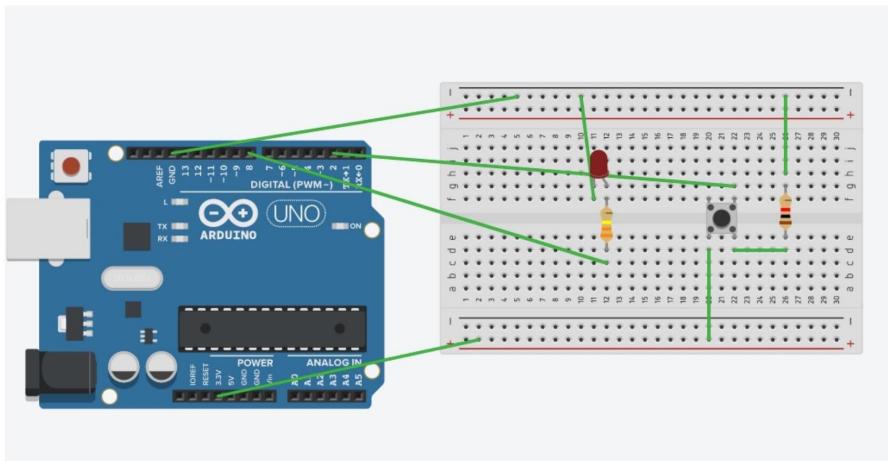
In regel 12 staat een if commando. Dat commando kijkt of de waarde tussen de haakjes erachter waar is of niet. In dit geval checkt hij of de waarde van KnopStatus gelijk is aan 1 .

Als je goed kijkt zie je dat hier == staat. == betekent het dat de computer moet kijken of het geen links en rechts van het == teken gelijk zijn. = betekent dat je een waarde toekent aan een object. = gebruiken binnen de haakjes van if levert dus een foutmelding op.

Als datgene wat tussen de staat haakjes waar is, wordt dat wat tussen de accolades staat van if uitgevoerd. Net als bij void setup en void loop eindigt een if niet met ; maar staan er accolades. Dit kan bij complexere code ingewikkeld worden omdat je dan een if binnen een if binnen een if kunt krijgen en dat je erg goed moet bijnouden hoe het zit met de accolades. Een tip hierbij is om de accolades die bij elkaar horen op dezelfde afstand van de kantlijn te plaatsen. Je ziet bijvoorbeeld dat de accolade op regel 11 even dicht bij de kantlijn staat als de accolade in regel 19. Die van regel 19 sluit dus die van regel 11 af. Vaak wordt de openingsaccoade direct achter de if of else geplaatst. De openingsaccoade van regel 13 hoort bij de sluitingsaccoade van regel 15 en de openingsaccoade van regel 15 hoort bij de sluitingsaccoade van regel 17.

Tussen de accolades van if staat het commando digitalWrite(LAMP, HIGH); Dat betekent dat er een signaal naar de lamp wordt gestuurd en wel een 1, dat zorgt er dus voor dat de lamp aangaat. De lamp blijft dan aanstaan tot dat er een 0 naar de lamp wordt gestuurd. Vergelijk het met een koelkast. Je stuurt nu een pak melk de koelkast in. Dat pak blijft in de koelkast totdat je het er actief uit haalt. Doe je niets, blijft het pak melk in de koelkast (en wordt het op een gegeven moment slecht, maar dat hoor niet bij deze module).

In regel 15 staat het woordje else. Dat zorgt ervoor dat er ook iets gebeurt als de waarde tussen de haakjes bij if niet waar is. In dit geval stuurt hij dan een 0 naar de lamp en gaat de lamp uit.



Deze schakeling hoort bij de bovenstaande code

Verwerkingsopdracht

Hierboven wordt aan de hand van een voorbeeld een stuk code uitgelegd. Ook wordt er een schakeling getoond die bij deze code hoort. De opdracht is het volgende: zorg dat je de schakeling aanpast zodat er nog een led wordt toegevoegd en nog een knop die deze led aan op uitzet. Je moet dus én de schakeling aanpassen én de code. Doe dit in je werkdocument in TinkerCad.

Gegevens die verzameld zijn moeten vaak opgeslagen worden om later te kunnen gebruiken. Ook zullen ze soms getransporteerd moeten worden via een netwerk. Het internet is zo'n netwerk en steeds meer apparaten staan in verbinding met het internet (zogenaamde "Internet of things" (IOT)). Daar hoort ook beveiliging bij. Dit hoofdstuk gaat over hoe gegevens opgeslagen worden en hoe systemen met elkaar communiceren.

Tijdens dit hoofdstuk leer je het volgende:

- Dat er vuistregels zijn voor het ordenen van gegevens binnen bestanden het ordenen van bestanden en dat het gebruiken van die vuistregels je tijd bespaart en de kans op fouten vermindert.
- Dat metadata de informatie bevat over wat er in bestanden te vinden is en hoe deze geordend zijn.
- Dat databases georganiseerd zijn op basis van de metadata en dat dat het zoeken in databases, en het samenvoegen of splitsen van databases mogelijk maakt
- Dat gegevens fysiek opgeslagen worden op harde schijven (in het klein) of in datacenters (in het groot) en hoe deze opslag functioneert
- Dat gegevens op verschillende manieren beveiligd kunnen worden door wachtwoorden en encryptie, en dat de wiskunde hier een grote rol in speelt.
- Dat het kraken van codes aan de wieg heeft gestaan van de ontwikkeling van computers
- Wat de rol van Alan Turing was in het ontstaan van computers
- Dat het internet een combinatie is van ip-adressen en TCP/IP protocollen en hoe dit werkt
- Hoe een wifi-verbinding werkt en hoe je zelf een wifi-ontvanger en zender kunt bouwen met een Arduino en daarmee gegevens kunt doorsturen naar andere digitale systemen
- Dat je informatie van een apparaat ook naar een website kunt sturen zodat je op afstand mee kunt kijken met de sensoren die jouw digitale systeem bevat.

Als er door jouw systeem 'besloten' wordt dat er iets in gang gezet moet worden (motor aan, koeling aan, ventilator uit etc.) moet dat natuurlijk wel gebeuren. Het is allemaal geen magie. In dit hoofdstuk gaan we in op hoe je je digitale systeem daadwerkelijk invloed kunt laten hebben op zijn omgeving.

Tijdens dit hoofdstuk leer je het volgende:

- Dat digitale systemen invloed kunnen hebben op hun omgeving door middel van actuatoren
- Dat actuatoren zoals lampjes, speakers, motoren, RF zenders, servomotoren in veel verschillende digitale systemen voorkomen en dat je die (relatief) eenvoudig zelf kunt bouwen

- Dat je in je programmeercode in je digitale systeem soms bepaalde bibliotheken moet gebruiken om actuatoren te kunnen gebruiken
- Dat je voor het aansturen en monteren van actuatoren online veel informatie kunt vinden op websites als stackoverflow en reddit.
- Dat het aansturen van actuatoren gevolgen heeft voor het stroomverbruik van je digitale systeem en dat ontwerpkeuzes dus invloed hebben op de duurzaamheid van je systeem
- Dat je door te kijken hoe goed jouw actuatoren functioneren en hun omgeving beïnvloeden kunt beoordelen of jouw digitale systeem voldoet aan de ontwerpeisen die je hebt gesteld.

Een voorbeeld

Het bovenstaande kan allemaal erg onhaalbaar lijken. Daarom wordt hier een voorbeeld uitgewerkt. Ikjzelf (de auteur: Rudy Jonker) ga tijdens de eerstvolgende ronde van deze module gelijktijdig met de leerlingen een ontwerp maken. Ik deel alle stappen hier op deze plek zodat je een beeld krijgt hoe zo'n ontwerp tot stand komt en welke problemen worden tegengekomen.

Ronde 1

Doelstelling ontwerp:

Ik wil een wake-up light maken. Dat wil zeggen een wekker waarbij als de ingestelde tijd is aangebroken er een lamp steeds feller en steeds witter gaat branden. Als het goed is wordt de persoon die slaapt dan ook wakker daardoor. Ik wil kunnen meten of dit ook zo is.

Pakket van eisen:

- Er moet een tijd lopen op een klok
- Er moet een alarmtijd ingesteld kunnen worden
- Er moet een lamp aan kunnen gaan met een variabele felheid en kleur
- De persoon die wakker wordt moet ergens op kunnen drukken om aan te geven dat hij/zij wakker is.
- Dit moet door het systeem opgeslagen worden op een SD-kaart, zodat ik de prestaties van de lamp kan onderzoeken.

Deeluitwerkingen

Ik begin met de klok die moet lopen. Ik wil de uren en minuten laten zien, en een uur moet ook echt een uur duren, anders is de klok niet lang 'houdbaar'. Ik start met de code en check deze in de seriële monitor van Arduino. Daardoor hoef ik me nog even niet druk te maken over de display.

Ontwerp:

```
long int mins = 0;
long int hrs = 0;
long int seconds = 0;
void setup() {
// put your setup code here, to run once:
    Serial.begin(9600);
}
void loop() {
    delay(1000); //hier verstrijkt 1000ms, dat is dus 1 seconde
    //long int t1 = millis(); //deze regel staat nu even uit
    seconds = seconds +1;
    if(seconds==60){
        seconds = 0; // als het aantal seconde 60 wordt, reset hem naar 0 en doe de minuten +1
        mins = mins+1;
    }
}
```

```
if(mins==60){  
    mins = 0; // als het aantal minuten 60 wordt, reset hem naar 0 en doe de uren +1  
    hrs = hrs+1;  
}  
if(hrs==24){  
    hrs=0; //als het aantal uren 24 wordt, reset hem naar 0  
}  
//long int t2 = millis(); //deze regel staat nu even uit  
//long int calctime= t2 -t1; //deze regel kan ik aanzetten als ik wil weten wat de tijd is  
Serial.print (hrs);  
Serial.print(" ");  
Serial.print(mins);  
Serial.print(" ");  
Serial.print(seconds);  
Serial.print(" ");  
//Serial.println(calctime);  
}  
}
```

De klok in werking

Evaluatie

De klok doet het. Ik wilde eerst iets doen met het vastleggen van een tijd, en dan als die tijd een minuut geleden was, de minuten +1 doen, maar dat bleek veel te veel geklooi op te leveren. Daarom heb ik gekozen voor een delay. Mocht de code later nou zo ingewikkeld worden dat het uitvoeren van de code lang duurt (waardoor de tijd van 1 seconde op de klok dus 1000ms PLUS de verwerkingsstijd van de code zou duren) heb ik twee regels ingebouwd waarmee ik kan meten hoe lang de code duurt. Als het dan nodig is kan ik de delay verkleinen om voor de verwerkingsstijd van de code te corrigeren. Op naar de display en de knopjes!

0.37.8 Testfile

```
a = 2
b = 3
print(a+b)

import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 100)
y = np.sin(x)

plt.figure()
plt.xlabel('x')
plt.ylabel('y')

plt.plot(x, y)
plt.show()
```

0.37.9 Wall of Fame

Omdat deze module door veel scholen gebruikt wordt en daar allemaal leerlingen coole ontwerpen maken zijn er ongetwijfeld prachtige ontwerpen om hier te presenteren. Wil je jouw (beste) ontwerp op deze site? Stuur dan een stukje tekst met wat foto's of evt. een linkje naar een youtube filmpje. Zorg dat het doel van je ontwerp duidelijk is en voor welke oplossingen je gekozen hebt.

Over dit lesmateriaal

Colofon Auteurs: Digitaal Redacteur; Rudy Jonker; H. Dirks Team: Schakelmodule Digitale Technologie Laatst gewijzigd: 24 februari 2023 om 13:06 Licentie: De Internationale Creative Commons 4.0 licentie waarbij de gebruiker het werk mag kopiëren, verspreiden en doorgeven en afgeleide werken mag maken onder de voorwaarden: Naamsvermelding en Gelijk Delen, zie <http://creativecommons.org/licenses/by-sa/4.0/>. Meer informatie over de CC Naamsvermelding-GelijkDelen 4.0 Internationale licentie licentie.

Aanvullende informatie over dit lesmateriaal Van dit lesmateriaal is de volgende aanvullende informatie beschikbaar:

Eindgebruiker leerling/student

Studiebelasting 4 uur en 0 minuten

Bronnen Slimme regenton <https://slimmeregenton.nl/>

1 Introductie digitale technologie. Daan Geijs vertelt over zijn studiekeuze en opleiding <https://youtu.be/H94a2UCkYXY>

2 Wat is kunstmatige intelligentie? Daan Geijs vertelt wat er met digitale technologie mogelijk is. <https://youtu.be/p9vdykaXKg4>

3 Kunstmatige intelligentie in het ziekenhuis. Daan Geijs vertelt over zijn onderzoek aan het herkennen van tumoren met de computer. https://youtu.be/VqiOqBVe_08

4 Daan Geijs over het werk dat hij in de toekomst kan gaan doen <https://youtu.be/3t5nQqr28II>

Tinkercad TMP36 <https://www.tinkercad.com/things/94VFDq4ienj-copy-of-tmp36-temperature-sensor-witharduino/editel?tenant=circuits>

Potentiometer <https://youtu.be/H3hSQgZxNe0>

Ultrasone sensor https://youtu.be/eYJg_Yp-Fys

Tutorial voor de ultrasone sensor <https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6>

Arduino tutorial neopixel <https://create.arduino.cc/projecthub/robocircuits/neopixel-tutorial-1ccfb9>

Neopixel voorbeeld <https://youtu.be/ANoG6DoSFHA>

link naar liquidcrystal_i2c (datum link: 7-2-2022) <https://downloads.arduino.cc/libraries/github.com/marcosc/I2C-1.1.2.zip>

Voorbeeldschakeling met Arduino, Relais en Lamp. https://maken.wikiwijs.nl/152360/NLT_Digitale_Techniek_page-7105936

De klok in werking <https://youtu.be/nmrz2QJYan4>