BACHELORPROEF

Internet-of-Things in het onderwijs

Ontwikkeling van een IoT-cursus en een integratieproject waarin een leervolgsysteem wordt ontworpen

Bachelor	Toegepaste Informatica
Afstudeerrichting	Software & Systems Engineer
Academiejaar	2015 – 2016
Student	Daan Pape
Interne begeleider	Olivier Sourie
Externe promotor	Johan Coppieters

howest.be

Samenvatting

Internettechnologie wordt steeds belangrijker en zelfs onmisbaar in onze huidige leefwereld. Het Internet-of-Things, waarbij naast computers ook allerhande elektronische toestellen aan het internet worden gekoppeld, belooft in de komende jaren een volgende internetrevolutie te weeg te brengen. Het is dus onmisbaar om de IT-ers van morgen reeds vandaag kennis te laten maken van deze nieuwe technologie.

In dit document wordt beschreven hoe een inleidende IoT-cursus kan worden ingericht. Er wordt dieper ingegaan op het opzetten van een onderzoekslabo en er wordt afgesloten met enkele volledig uitgewerkte IoT-projecten. Dit document kan u zien als een leidraad voor het ontwikkelen van een vooruitstrevende Internet-of-Things opleiding binnen uw onderwijsinstelling en/of bedrijf.

Abstract

Internettechnology is getting ever more important and even essential in our society. The Internet-of-Things takes the internet one step further and not only connects humans to each other, but also electronic appliances and sensors. This techologoy promises to unleach the next internet revolution. It's therefore very important to start teaching IoT-technology to the programers and system engineers of tomorrow.

This documents describes how high-schools and universities can create their own IoT-course, it elaborates on what infrastructure and hardware is essential to have and what one can do with the hardware. Based on this thesis a teacher can develop a pioneering Internet-of-Things course to get people and students excited about the new technologies which IoT brings to us.

Verklarende woordenlijst

Woord/Afkorting	Verklaring
API	Een API of Application Programming Interface is een lijst van afspraken en functies waarmee een computerprogramma met een ander computerprogramma kan communiceren.
Binary blob	Een stukje gecompileerde computercode waarvan de broncode niet beschikbaar is
HTML	De HyperText Markup Language is een manier om via een tekst te structureren waarbij men via hyperlinks kan doorklikken naar andere HTML-pagina's.
IDE	Een Integrated Development Environment is een softwarepakket die de programmeur helpt om te programmeren
IoT	Het IoT of Internet-of-Things is een systeem waarbij naast mensen en computers ook andere toestellen via het internet met elkaar worden verbonden.
JSON	JavaScript Object Notation is een manier om javascriptobjecten te serialiseren en dit wordt zeer vaak gebruikt door computerprogramma's om met elkaar te communiceren.
LCD	Liquid Crystal Display is een bepaald type schermtechniek om platte schermen te bouwen.
LED	Een Light Emitting Diode of LED is een speciaal soort diode die licht uitstraalt als er stroom door loopt.
MITM	Een Man-In-The-Middle attack is een computeraanval waarbij de aanvaller tussen de zender en ontvanger het verkeer manipuleert.
MosFET	Een Metal-Oxide-Semiconductor Field-Effect Transistor is een modern type transistor, het beste te vergelijken met een elektronische schakelaar.
NFC	Near Field Communication is een techniek om via radiogolven met een chip te communiceren zonder dat de ontvanger een batterij nodig heeft.
Open source	Met open-source wordt aangeduidt dat de broncode en/of bouwplannen beschikbaar zijn op het internet en dat andere vrij zijn om deze aan te passen.
PCB	Een printplaat wordt ook wel Printed Circuit Board genoemd.
SBC	Dit staat voor Single Board Computer, dit is een kleine computer waarbij alle onderdelen op één enkele printplaat gemonteerd zitten.

Woord/Afkorting	Verklaring
True RMS	RMS staat voor Root-Mean-Square en duidt op de effectieve waarde. Deze aanduiding duidt bij meetapparatuur aan dat deze de echte waarde van wisselspanning kan meten.
UI	De User-Interface van een applicatie is dat deel wat de gebruiker te zien krijg op zijn scherm, de presentatie van het programma als het ware.
USB	Universal Serial Bus is een standaard voor snelle seriële communicatie, USB ik zeer herkenbaar aan de typische USB-poort.
VLAN	of Virtual Local Area Network is een techniek die het mogelijk maakt om meerdere LAN netwerken op één kabel te plaatsen en het netwerk te segmenteren.

Inhoudsopgave

1 Partners en opdrachtgevers

Dit document is opgesteld in opdracht van Howest. Howest is een sterk groeiende hogeschool met campussen in Brugge en Kortrijk. De onderwijsinstelling scoort zeer hoog met zijn kwaliteitsvolle informaticarichtingen. Dit maakt de instelling zeer geschikt om als eerste in Vlaanderen een volwaardige Internet-of-Things cursus op te richten. De hands-on cursus vervat in dit document is door Howest reeds in de praktijk verwezenlijkt.

Als hardware partner voor deze cursus werd gekozen voor het Belgische DPTechnics. Deze hardware en software start-up heeft een uniek IoT-platform ontwikkeld dat zowel voor commerciële als educatieve doeleinden kan worden gebruikt. De bedrijfscultuur is open-source waardoor de studenten het volledige systeem kunnen begrijpen.

2 Onderzoeksvraag

Internet-of-Things is een opkomende technologie waarvan zeker is dat deze een volgende internetrevolutie teweeg zal brengen. Bedrijven en onderwijsinstellignen moeten voorbereid zijn op deze nieuwe golf van technologie. Dit document geeft antwoord op volgende vraag:

Hoe kan een basiscursus Internet-of-Things worden ingericht waarbij de cursist zelf IoT applicaties kan ontwerpen?

Deze vraag wordt met een hands-on aanpak beantwoord en aan de hand van verschillende voorbeelden wordt een inleiding tot IoT gegeven. Het is de bedoeling dat dit naslagwerk zonder veel moeite kan worden omgevormd tot een volwaardig handboek dat ook door de cursist kan worden gebruikt.

3 Cursusbenodigdheden

3.1 De DPTechnics IoT-kit

Speciaal voor deze cursus heeft DPTechnics een Internet-of-Things kit samengesteld, deze is te zien op figuur 1. De onderdelen in deze kit laten toe alle opdrachten in dit document uit te voeren en daarna verder te experimenteren. Deze mogelijkheid om ook projecten die niet in dit document beschreven staan uit te voeren is heel belangrijk. Het opbouwen van de hardware is namelijk iets dat veel moet worden gedaan vooraleer dit vlotter gaat. Hierbij is afwisseling belangrijk omdat steeds het zelfde project opnieuw bouwen zeer demotiverend werkt in tegenstelling tot het zelf ontwerpen van een nieuwe schakeling.

Alle onderdelen van de IoT-kit zitten in een handige opbergkoffer. De cursisten kunnen deze gemakkelijk naar eender welk leslokaal meenemen. Voor de hands-on projecten in dit document is geen internetverbinding vereist. Het is echter wel aan te raden om een bekabelde of draadloze internetverbinding te voorzien om de software van de IoT-kit te upgraden. Ook indien men geconnecteerde projecten wil bouwen aan de hand van het BlueCherry platform is een internetverbinding noodzakelijk.

Figuur 1: De DPTechnics Internet-of-Things kit

De kit bevat volgende handige onderdelen:

- DPT-Board+
- 20x man-man jumper kabel
- 40x vrouw-vrouw jumper kabel
- 400 punts breadboard

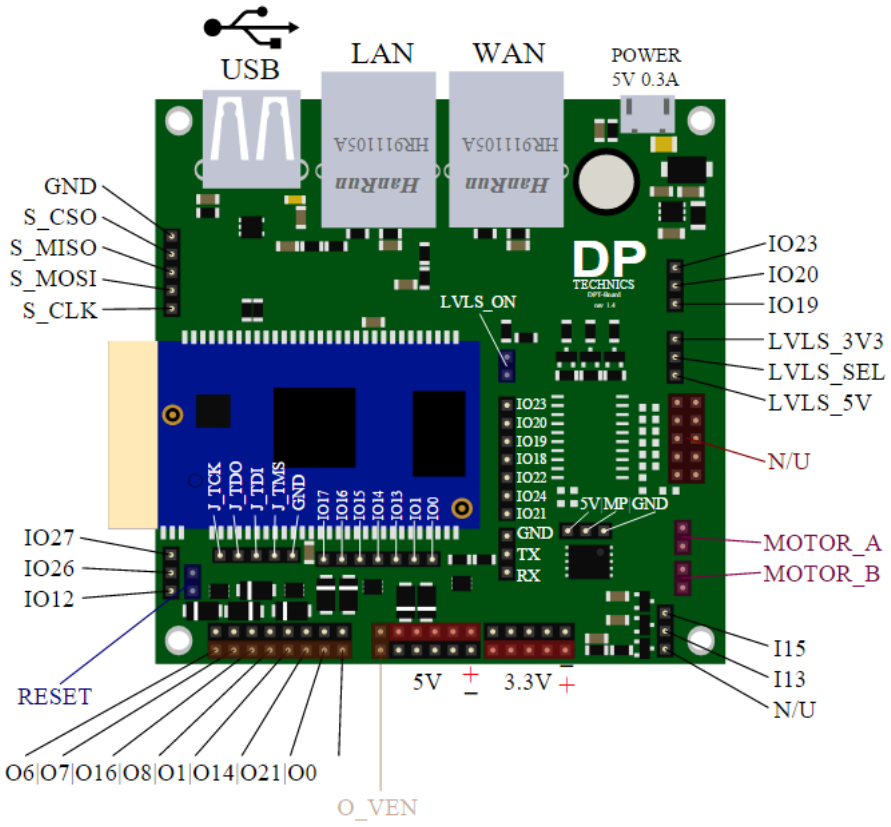
- 5 rode LED's
- 5 groene LED's
- 5 gele LED's
- 10 $\times$ 220Ω weerstand
- 10 $\times$ 1kΩ weerstand
- 10 $\times$ 10kΩ weerstand
- 10 $\times$ 100kΩ weerstand
- 2 LDR sensoren
- 6 drukknoppen
- US-100 digitale afstand en temperatuur sensor
- 16x2 I^2C LCD
- 2 digitale nabijheidssensoren
- 9 grams micro servo
- Opbergkoffer

3.1.1 Het DPT-Board

De basis van de IoT-kit is het DPT-Board+. Dit is een open source Internet-of-Things modem ontwikkeld door DPTechnics en het wordt zowel in consumentenhardware, hobby projecten als in het onderwijs gebruikt.

Het DPT-Board, afgebeeld op figuur 2, is een volledige computer met veel digitale aansluitingen. Zo heeft het board twee 'fast ethernet' poorten en WiFi g/n aan boord om rechtstreeks met het internet te verbinden. Naast de verschillende netwerkaansluitingen is het board voorzien van veel voorkomende digitale interfaces:

- **8 n-mosFET uitgangen**: deze uitgangen kunnen lasten tot 20 volt tot maximaal 1 ampère schakelen. Deze uitgangen zijn daarenboven voorzien van een vrijloopdiode waardoor er ook inductieve lasten zoals motoren en spoelen mee kunnen worden geschakeld.
- **3 beveiligde ingangen**: veel SBC's (Single Board Computers) kunnen slechts tot 5 volt of slechts 3,3 volt op hun ingangen verdragen. Het board van DPTechnics heeft ingangen die beveiligd zijn door middel van diodes en een RC-filter. Zodoende kunnen ook digitale sensoren die bijvoorbeeld op 12 of 24 volt werken rechtstreeks worden ingelezen.
- **3 level-shifted communicatiepoorten**: veel sensoren zoals gyroscopen, afstandsensoren, GPS, magnetometer, nabijheidssensoren, ... worden digitaal uitgelezen. Veel van deze sensoren werken op 5V of 3,3V en deze twee spanningen zijn niet compatibel met elkaar. Intern werkt het DPT-Board op 2,5V en er moet dus gelevel-shift worden voor beide types van sensoren. Veelal kan dit worden gedaan met een simpele weerstandsdeler maar sommige protocollen zoals I^2C vereisen toch wat complexere logica. Het DPT-Board is daarom voorzien van 3 mosFET-shifters die alle communicatieprotocollen zoals I^2C, SPI, 1-wire, ... ondersteunen.
- **Hoge snelheid SPI**: SPI is het protocol bij uitstek wanneer er een snelle bus tussen IC's nodig is. Het DPT-Board heeft een zeer snelle, tot 50MHz, SPI poort welke



Figuur 2: Schematisch overzicht van het DPT-Board

wordt gebruikt om het flash geheugen aan te sturen. Daar het besturingssysteem echter in het RAM geheugen draait kan deze poort ook worden gebruikt om andere snelle randapparatuur zoals een scherm aan te sturen.

- **JTAG poort**: JTAG is een standaard die in de jaren 80 in het leven is geroepen om de almaar complexere elektronica op een gemakkelijke manier te kunnen testen. De JTAG standaard maakt het mogelijk de processor, een Qualcomm Atheros AR9331, tot op instructieniveau te debuggen of om rechstreeks in het flashgeheugen te werken.
- **USB 2.0**: de voor iedereen bekende USB aansluiting maakt het mogelijk om allerhande extra toestellen toe te voegen zoals webcams, 3G/4G sticks, geluidskaarten, ...
- **Dubbele H-brug**: DPTechnics heeft speciaal voor het onderwijs ook een robot ontworpen. Dit platform bevat twee motoren om de robot aan te drijven en te besturen. Het DPT-Board+ kan deze twee motoren rechtstreeks aansturen aangezien de motorcontroller ingebouwd zit.

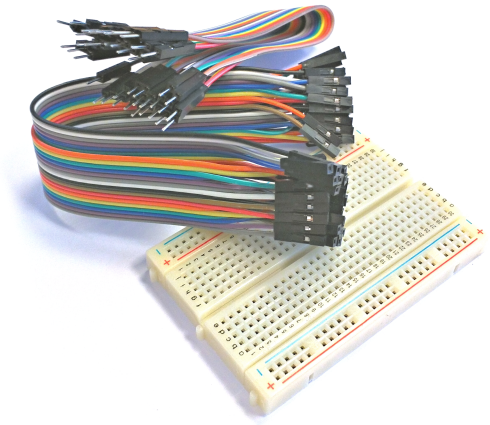
De SBC van DPTechnics heeft een efficiënte schakelende voeding aan boord waardoor het board met ingeschakelde WiFi en 100% CPU belasting slechts 0,45 watt verbruikt. Een

Raspberry Pi bijvoorbeeld heeft geen schakelende maar een lineaire spanningsregelaar aan boord, samen met de snellere processor maar zonder WiFi verbruikt deze 2 watt. Een IoT-device is meestal bedoeld voor 24/7 operatie en een laag verbruik is dus van kapitaal belang. Ook als men het toestel op batterijen wil laten werken is dergelijk verschil zeer belangrijk. Zie ook hoofdstuk xxx waarin het verschil in autonomie wordt berekend.

De software die op het DPT-Board draait is volledig open source en kan worden teruggevonden op de github pagina van dptechnics: github.com/dptechnics. Het besturingssysteem, DPT-Board-OS, is gebaseerd op OpenWRT Linux. Deze Linux distributie is geoptimaliseerd voor embedded toepassingen en volledig open source, er zijn geen verborgen binary blobs. Met meer dan 4000 verschillende softwarepakketten beschikbaar voor het systeem kan zonder zelf te programmeren reeds veel worden ontworpen.

3.1.2 Jumper kabels en breadboard

Elke Internet-of-Things ontwikkelaar moet een basiskennis van electronica hebben en de beste manier om deze op te bouwen is door zelf te experimenteren. Het breadboard, zie figuur 3 is een manier om zonder solderen toch elektronische schakelingen op te bouwen.



Figuur 3: Breadboard met mannelijke en vrouwelijke jumper kabels

Trough-hole elektronicacomponenten zoals weerstanden, condensatoren, LED's, ... houden zich veelal aan de standaard om pinnetjes om de 2,54mm te plaatsen. Een breadboard maakt hier handig gebruik van door een 2,54 mm aan te bieden waarbij contacten onderling doorverbonden zijn op een welbepaalde manier.

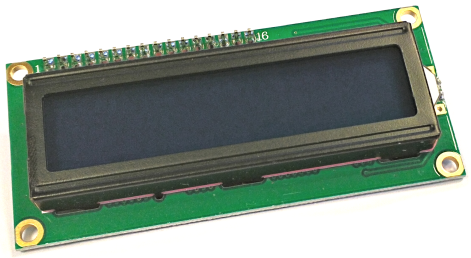
De jumper kabels maken het gemakkelijk om externe sensoren en schermen onderling met elkaar te verbinden of op het breadboard aan te sluiten. Ook zijn de mannelijk-mannelijk jumper kabels nodig om verschillende componenten onderling op het breadboard met elkaar te verbinden. In het eerste hands-on project, in hoofdstuk xxx, wordt dieper ingegaan op het gebruik van het breadboard en de kabels.

3.1.3 LED's, drukknoppen en weerstanden

Computer stellen niets voor zonder een gebruikersinterface. Laptops en desktops interageren met de gebruiker via beeldschermen, muis en toetsenbord, geluid, ... Embedded hardware kan van dezelfde mogelijkheden gebruik maken maar dit is veelal te duur en niet noodzakelijk. De IoT-kit voorziet daarom in vrijwel overal gebruikte LED's en drukknoppen zoals bijna elk elektronisch toestel deze heeft.

3.1.4 16×2 LCD scherm

LED's schieten soms te kort als er aan de gebruiker meer complexe data moet worden weergegeven. In dat geval wordt er veelal gebruik gemaakt van een scherm. Schermen zijn er in alle soorten en maten maar in simpele toestellen wordt meestal gekozen voor een 16×2 HD44780 compatibel LCD zoals op figuur 4.



Figuur 4: 16×2 HD44780 compatibel LCD

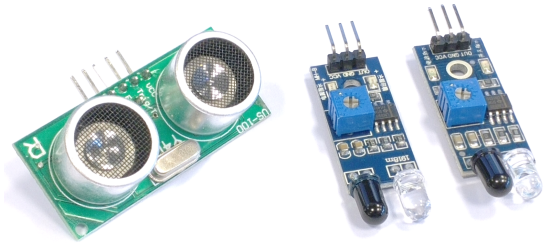
Deze schermen hebben 2 regels waarop elk 16 karakters kunnen worden weergegeven. Omdat een LCD rechtstreeks aansturen veel CPU kracht vraagt wordt meestal met een controller chip gewerkt. De HD44780 is standaard en wordt door alle grote en kleine halfgeleiderfabrikanten geproduceerd. De aansturing gebeurt over een 8-bits of 4-bits parallelle bus.

Het LCD-scherm in de IoT-kit is voorzien van deze HD44780 controller en bovendien ook van een PCF8574AT I^2C expander. Deze expander zorgt ervoor dat er geen 6 maar slechts 2 kabels naar de controller nodig zijn om het scherm aan te sturen. Hierdoor blijft er veel meer ruimte over om andere randapparatuur op het DPT-Board aan te sluiten als het scherm in gebruik is.

3.1.5 Afstandssensoren en temperatuursensor

Toestellen interageren niet enkel met mensen maar moeten ook in staat zijn om hun omgeving waar te namen. In de DPTechnics IoT-kit zijn verschillende sensoren opgenomen die dit mogelijk maken, deze zijn te zien op figuur 5:

- **US-100**: De US-100 sensor is een zeer flexibele sensor die zowel temperatuur als afstand meet. De afstand wordt gemeten door middel van ultrasoon geluid en de sensor



Figuur 5: US-100 afstand en temperatuursensor (links) en digitale nabijheidssensoren (rechts)

kan tot op 1mm nauwkeurig meten tussen 2cm en 450cm.

De temperatuursensor die ook in de US-100 zit ingebouwd is tot op 1°C nauwkeurig en wordt intern gebruikt om de afstand zo nauwkeurig mogelijk te meten. Het is namelijk zo dat de snelheid van het geluid verandert van zodra de temperatuur wijzigt.

- **Digitale nabijheidssensor**: deze sensor werkt op basis van reflectie van infrarood licht. Van zodra de ontvanger genoeg infraroodlicht ontvangt wordt de uitgang hoog, de gevoeligheid van de ontvanger kan met een instelpotmeter worden aangepast. Zo kan met twee van deze sensoren een lijnsensor worden gemaakt die een zwarte lijn op een lichte ondergrond kan detecteren.

3.1.6 9-grams servo

Een servo-motor, zie figuur 6 verschilt van een gewone motor omdat er een feedbackcircuit in zit. Dit wil zeggen dat de ingebouwde elektronica op elk moment weet wat de stand van de motoras is. Daardoor kan een servo zeer nauwkeurig worden aangestuurd wat handig is in allerlei toestellen zoals robot-armen, modelbouw, ... Het DPT-Board kan deze servo rechtstreeks aansturen via gelijk welke I/O poort omdat de motorcontroller ingebouwd zit in de servo.

3.2 Optioneel robotplatform

De IoT-kit is ideaal om voor de eerste keer kennis te maken met embedded computers en elektronica. Als de cursist echter gebeten is door deze technologische revolutie biedt DPTechnics ook een robot platform aan. De DPT-Robot is een simpel platform dat met twee motoren kan worden bestuurd. Het DPT-Board+ kan rechtstreeks op de robot worden gevezen en door een li-ion powerbank worden gevoed.
Zoals op figuur 7 is te zien is de bodemplaat van de DPT-Robot voorzien van heel wat montagegaten. Dit maakt het mogelijk om de sensoren die in de IoT-kit zitten op heel wat verschillende manieren te monteren op de robot. De nabijheidssensoren kunnen op deze manier bijvoorbeeld zowel als botsdetectie of als lijnvolgsensor worden gemonteerd.

Figuur 6: 9-grams micro servo

Het hebben van één robot platform per 2 of 3 cursisten is zeker een aanrader omdat het programmeren van een bewegend toestel als zeer leuk wordt ervaren.

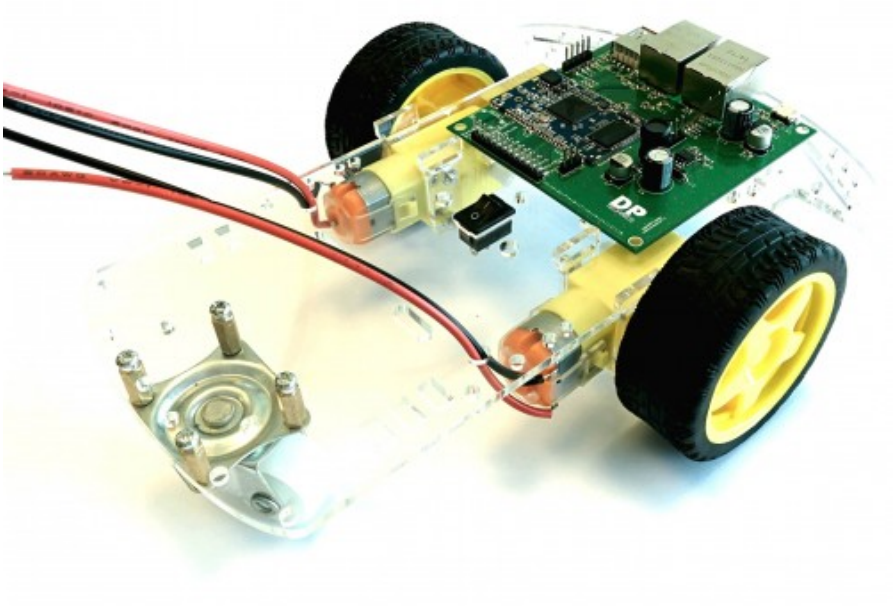
3.3 Optioneel onderzoekslabo

Indien het bedrijf of de onderwijsinstelling die de IoT-cursus inricht zich meer wil toespitsen op dit onderwerp kan het interessant zijn hiervoor een onderzoekslabo op te richten. Dergelijk labo maakt het mogelijk om meer diepgaande metingen uit te voeren op bestaande en nieuwe hardware of om zelf nieuwe hardware te ontwikkelen. Het opzetten van dergelijke onderzoeksruimte kan met een relatief beperkt budget worden gerealiseerd.

In dit hoofdstuk worden de belangrijkste onderdelen van een Internet-of-Things labo besproken. De voorgestelde opstelling is voldoende om 99% van alle IoT-toepassingen in-house te ontwerpen en dus ideaal om te starten.

3.3.1 Oscilloscoop

Op figuur 8 wordt een digitale oscilloscoop afgebeeld. Dit is een instrument waarmee de spanning op een kabel kan worden gevisualiseerd en kan worden gezien als het 'zwitsers zakmes' onder de meetapparatuur.
Een digitale oscilloscoop kan, in tegenstelling tot zijn analoge tegenganger, ook trage signalen goed weergeven. In de meeste IoT apparaten blijft de snelheid van signalen onder de 10MHz en dus is geen duur model vereist. Het is eerder aan te raden om te kiezen voor meer meetkanalen dan een hogere bandbreedte. Dit grote aantal kanalen is handig om bijvoorbeeld communicatiebussen zoals I^2C of SPI te debuggen omdat deze veelal uit 2 tot 4 draden bestaan.



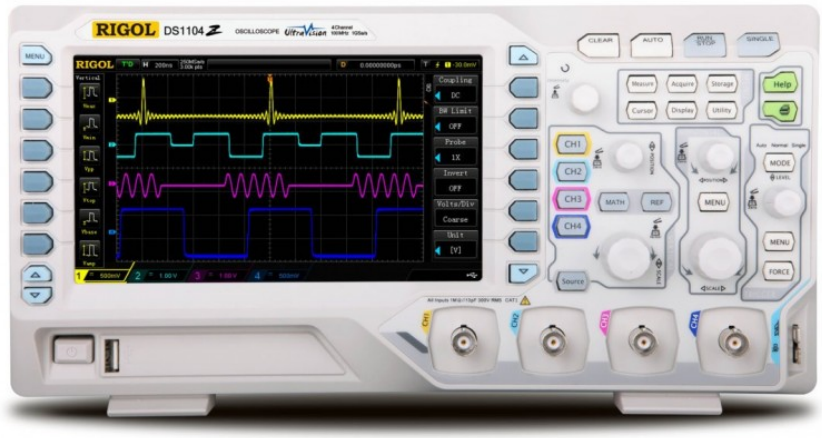
Figuur 7: Het DPT-Robot platform

De oscilloscoop die op dit moment van schrijven het meest interessant lijkt te zijn is de Rigol DS1054Z. Deze instap oscilloscoop heeft een bandbreedte van 50MHz en heeft 4 kanalen met een sample rate tot 1GS/s. De prijs/kwaliteitsverhouding van deze toestellen is zeer goed waardoor het een goede keuze is in het onderzoekslabo.

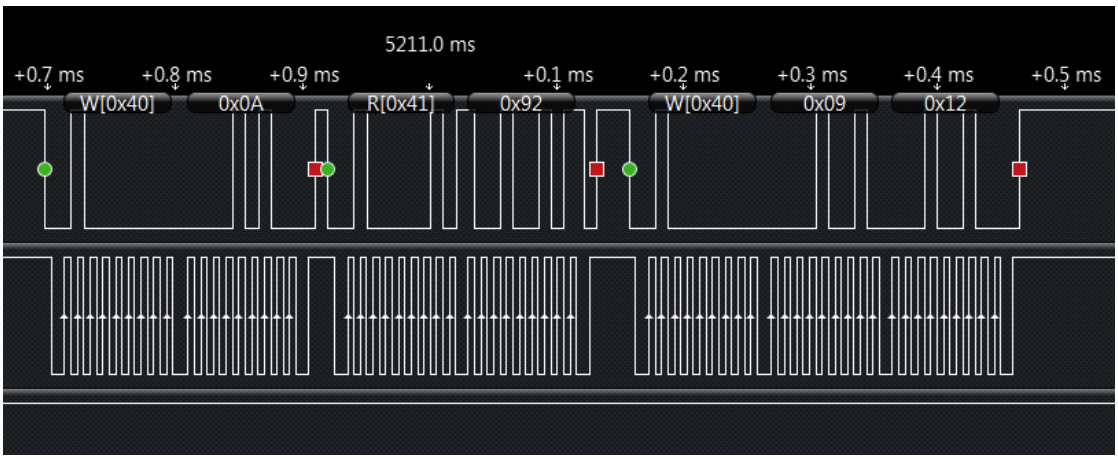
3.3.2 Logic analyzer

Een oscilloscoop is perfect om signaalkwaliteit te bekijken of om de kwaliteit van een voeding te onderzoeken. Hoewel de oscilloscoop ook kan worden gebruikt om digitale communicatie te ontcijferen is dit niet de meest gemakkelijk methode omdat de signalen niet worden geïnterpreteerd.

Een logic analyzer (LA) is hiervoor beter geschikt daar deze de signalen ook kan interpreteren. Als een LA wordt aangesloten op een SPI, I^2C of andere communicatie dan kunnen de verzonden bytes worden weergegeven zoals op figuur 9. Indien er veel met dergelijke communicatiebussen wordt gewerkt is het zeker aan te raden om naast de oscilloscoop een



Figuur 8: Rigol digitale oscilloscoop



Figuur 9: Logic analyzer software

LA aan te schaffen.

3.3.3 Labo voeding

Het DPT-Board kan rechtstreeks via een USB kabel van stroom worden voorzien. Echter vereisen veel randaparaten of meer stroom dan de 500mA die een USB poort kan leveren of een andere spanning dan de 5 volt die op een USB poort staat. Daarom is het voor elk

elektronicalab onontbeerlijk om minimaal één labovoeding te hebben. In de praktijk zijn de spanningen 3,3 volt, 5 volt en 12 volt veel voorkomend en meestal wordt niet meer dan 2 ampère afgenomen. Dit valt ruim binnen het bereik van de meeste labovoedingen.

Op figuur 10 staat een voorbeeld van een standaard dubbele labovoeding. Er kan gekozen worden tussen analoge en digitale meters, er wordt aangeraden om de digitale versie te kiezen daar de aflezing veel makkelijker is. Het is ook aan te raden een model te kiezen dat naast twee regelbare spanningen/stromen ook een vaste 5 volt aansluiting heeft. Dit kan veelal handig zijn als er bijvoorbeeld een 5V sensor samen met een 3,3 volt microcontroller en 12 volt motor moet worden gevoed.

Figuur 10: Dubbele gestabiliseerde labovoeding

3.3.4 Multimeter

Er zijn heel veel verschillende soorten multimeters van zeer goedkoop tot zeer duur. De aanschaf van een multimeter is dan ook niet zo makkelijk en wat volgt zijn slechts enkele algemene richtlijnen. Het is belangrijk te letten op volgende zaken:

- **True RMS**: bij het meten van een niet-gelijkspanning zal een niet true RMS meter een meetfout geven. De meerkost van een true RMS meter weegt tegenwoordig niet meer op tegen de verwarring die de meetfout kan veroorzaken, daarom wordt aangeraden om zowieso voor een true RMS model te gaan.
- **Aantal counts**: het aantal 'counts' zegt iets over de resolutie van de multimeter of het aantal verschillen in spanning dat de meter kan registreren onafhankelijk van de plaats van de komma. Hier geldt dus hoe hoger, hoe beter. Er wordt aangeraden om minimaal een 2000 counts meter aan te schaffen.
- **Autorange**: de analoog-digitaal omzetter in een multimeter heeft een vast interval, bijvoorbeeld van 0 volt tot 4,096 volt. Het meten van een ander interval zoals bijvoorbeeld tussen 0 volt en 200 volt moet dus met een verzwakker gebeuren. Om zo

nauwkeurig mogelijk te meten moet deze verzwakking geminimaliseerd worden. Bij een niet autoranging multimeter moet dit handmatig met een knop worden gedaan terwijl autoranging meters zelf de optimale verzwakking kiezen. Daarom wordt aangeraden steeds een autoranging model te kiezen.

- **Computer verbinding**: de iets duurdere meters bezitten vaak een functie waardoor hun data rechtstreeks naar een PC kan worden verzonden. Deze functie kan handig zijn als metingen over een lange periode moeten worden uitgevoerd zoals uren of zelfs dagen. Zo kan een spanningsgrafiek van enkele dagen nuttig zijn als er aan analoge schakelingen wordt gewerkt waarbij de nauwkeurigheid van een spanningsreferentie moet worden bepaald.

In deze cursus is gekozen voor een UNI-T UT61E multimeter met een USB datalogger. Deze meter heeft 22000 counts, een true RMS functie, autoranging en een USB datalogger functie voor een zeer betaalbare prijs. De grootheden die deze meter kan meten zijn AC/DC spanning, AC/DC stroom, weerstand, capaciteit en frequentie tot 220MHz.

3.3.5 Aansluitmateriaal

Een goed uitgerust IoT/elektronica labo heeft verschillende types aansluitkabel nodig om alles met elkaar te verbinden. De meest belangrijke types kabel zijn:

- krokodil-krokodil kabels: de bekende krokodil aansluiting is zeer handig om allerlei zaken die niet op elkaar passen snel aan te sluiten. Een minimum van 10 stuks is aan te raden in een IoT-labo daar deze kabels zeer veel worden gebruikt.

- banaan-krokodil kabels: de meeste labovoedingen hebben draaiconnectoren waar een banaanstekker in past. De banaan-krokodilstekkers maken het mogelijk om de labovoeding snel op een werkstuk aan te sluiten.

- jumper kabels: mannelijk-mannelijk, vrouwelijk-vrouwelijk en vrouwelijk-mannelijk zijn drie combinaties jumper kabels die zeker aan te raden zijn om te hebben. Deze worden op een breadboardschakeling zeer vaak gebruikt.

3.3.6 Soldeerbenodigdheden

Als een breadboard schakeling goed werkt is het vaak de volgende stap om deze schakeling daadwerkelijk op een printplaat te solderen. Allereerst is hiervoor een soldeerstation nodig met temperatuurregeling zoals te zien op figuur 11. De twee meest belangrijke parameters bij het kiezen van een soldeerstation zijn het vermogen en de soldeerbout zelf. Voor IoT-elektronica, welke relatief klein is, is een vermogen tussen de 50 en 80 Watt ideaal. Een te laag vermogen bemoeilijkt het solderen en kan leiden tot kapotte componenten omdat deze te lang moeten worden verwarmd. Een te hoog vermogen laat het soldeer verbranden wat voor slechte verbindingen kan zorgen.

Figuur 11: Weller WSD81i 80 Watt soldeerstation

De soldeerbout zelf is belangrijk omdat deze bepaalt welk type soldeerstiften er kunnen worden gebruikt. De soldeerstift is het uiteinde van de bout welke in aanraking komt met het soldeertin. Een goede stift om mee te beginnen is een ronde stift tussen 1mm en 2mm dik.

Minstens even belangrijk is om het juiste type soldeertin te kiezen. In consumentenelektronica mag sinds het invoeren van RoHS (Restriction of Hazardous Substances) geen loodhoudend soldeer meer worden gebruikt. Er wordt aangeraden om dit ook in het elektronica labo te respecteren wegens gezondheidsredenen. De loodvrije soldeerlegeringen zijn ondertussen ver genoeg doorontwikkeld om met de hand te kunnen gebruiken. De voorgestelde legering is Sn99.3Cu0.7 met een flux kern. Flux is een chemisch middel dat zorgt voor het goed uitlopen van het gesmolten soldeertin.

Figuur 12: Rookafzuiger met actieve koolfilter (links) en heteluchtstation (rechts)

Hoewel niet strikt noodzakelijk indien er slechts sporadisch wordt gesoldeerd kan een rookafzuiger een goede investering zijn. Niet alleen komt dit de gezondheid ten goede maar dit verhoogt ook het werkgemak. Een voorbeeld van een simpele rookafzuiger met aktieve

koolstoffilter, waardoor er geen luchtafvoer naar buiten moet zijn, is te zien op figuur 12.

Rechts op figuur 12 wordt een heteluchtstation afgebeeld. Indien het doel van het IoT-labo ook het onderzoeken is van hoe bestaande toestellen werken en/of men toestellen wil herstellen is een dergelijk toestel aan te raden. Van zodra er een IC met meer dan 4 pootjes moet worden gedesoldeerd, of een IC met soldeerpads onder de behuizing kan men eigenlijk niet zonder dit gereedschap. Het hetelucht station kan alle pootjes tegelijk verwarmen waardoor het IC gemakkelijk van de print kan worden gehaald.

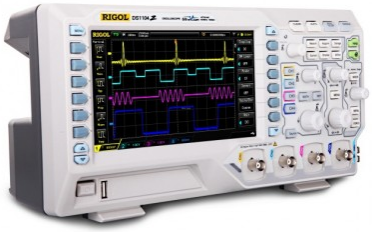
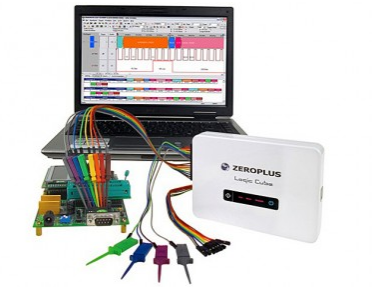
3.3.7 Netwerkinfrastructuur

In een IoT-labo zullen zeer veel zaken 'connected' zijn, wat wil zeggen dat een goede netwerkinfrastructuur een must is. Veelal zal een onderzoekslabo zich in een onderwijs en/of bedrijfswereld bevinden met een relatief restrictief netwerk. Daarom wordt aangeraden een apart afgescheiden netwerk in het labo aan te leggen. Hiervoor zijn volgende onderdelen nodig:

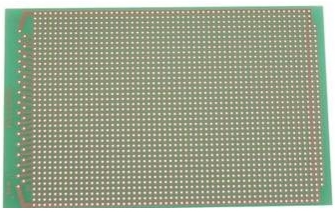
- **Router**: uiteraard moet het private netwerk van het labo aan het restrictieve bedrijfsnetwerk worden gekoppeld. Met behulp van een router wordt dit mogelijk. Om de kosten laag te houden kan men makkelijk zelf een router bouwen door op een server met minimaal 2 gigabit ethernet kaarten het pfSense software pakket te installeren. Deze opstelling biedt de meeste functionaliteit en hoogste snelheid.
- **Switch**: indien men wil werken met VLAN's is een managed switch aan te raden, zeker voor IoT toestellen die later in een corporate netwerk moeten opereren is het aan te raden VLAN functionaliteit te integreren. Power over Ethernet (PoE) waarmee toestellen gevoed worden via de netwerkkabel is ook een aan te raden functie.
- **Netwerkkabel**: als er geen bestaande kabelinfrastructuur is wordt aangeraden te kiezen voor categorie 6 kabel daar deze klaar is voor de toekomstige ethernetsnelheden.

3.3.8 Praktijkvoorbeeld

Om beter te kunnen inschatten wat de kosten zijn om een volledige uitgerust IoT onderzoekslabo in te richten is een praktijkvoorbeeld uitgewerkt. De vermelde prijzen zijn op het moment van schrijven de gemiddelde marktprijs van het product. In de tabel wordt aangegeven of een product een 'must-have' is of eerder een 'nice-to-have'. De 'nice-to-have' producten zijn niet noodzakelijk maar wel een gemak om aan te schaffen indien het budget dit toelaat. Omdat onderwijsinstellingen BTW dienen te betalen zijn alle beschreven bedragen inclusief BTW.

Oscilloscoop: Rigol DS1054Z		
	Aantal kanalen: 4 Bandbreedte: 50MHz Sample rate: 1GS/s max. Inclusief: - 4 probes - USB kabel Garantie: 3 jaar MUST-HAVE	€399,00
Logic analyzer: Zeroplus LAP-C 16000		
	Aantal kanalen: 16 Sample rate: 100MS/s max. Inclusief: - meetsnoeren - USB kabel Garantie: 2 jaar NICE-TO-HAVE	€120,00
Labovoeding: McPower Digi 302-05		
	Spanning: 0V - 30V Stroom: 0A - 5A Aansluitingen: - dubbel 0-30V - 5VDC vast Garantie: 2 jaar MUST-HAVE	€189,00

Multimeter: UNI-T UT61E		
	True RMS 20000 counts Autoranging USB datalogger Inclusief: - batterij - meetprobes - USB kabel - PC software Garantie: 2 jaar MUST-HAVE	€77,50
Aansluitmateriaal: banaan naar krokodil snoeren		
	Voor labovoeding MUST-HAVE	€1,75
Soldeerstation: Weller WSD 81i		
	Vermogen: 80W Soldeerpunt: 2.4mm Digitale temp. instelling MUST-HAVE	€299,00

Soldeertin: Loodvrij met harskern		
	Samenstelling: -97%Sn -1%Cu -2%Sb Diameter: 0.5mm MUST-HAVE	€8,00
Experimenteerprint: FR4 met eilandjes		
	Materiaal: FR4 Enkelzijdig Aantal: 3 MUST-HAVE	€3,00
Aansluitmateriaal: krokodillenklemsnoeren		
	Aantal: 10 MUST-HAVE	€2,80
Gereedschap: 6-delige schroevendraaierset		
	Kruiskop en platte kop MUST-HAVE	€24,95

Gereedschap: 6-delige preciesie schroevendraaier set		
	Kruiskop en platte kop MUST-HAVE	€27,95
Gereedschap: zijkniptang		
	Electronica kniptang MUST-HAVE	€13,95
Gereedschap: puntbektang		
	Electronica kniptang MUST-HAVE	€14,95
Soldeerstation: Aoyue 852 hete-lucht-station		
	Vermogen: 450W Temperatuur: 100°C - 480°C NICE-TO-HAVE	€100,00

Hulpgereedschap: derde hand met vergrootglas		
	NICE-TO-HAVE	€3,95
Componenten: E3-reeks weerstanden		
	MUST-HAVE	€7,95
Componenten: E12-reeks weerstanden		
	MUST-HAVE	€11,95
Componenten: verzameling transistoren		
	BC547B, BC557B, BC337, BC327, BC517, BC516, BD139, BD140 MUST-HAVE	€9,95

Componenten: verzameling LEDs		
	MUST-HAVE	€9,50
Componenten: set elektrolitische condensatoren		
	MUST-HAVE	€12,50
Componenten: set keramische condensatoren		
	MUST-HAVE	€12,50
Ontwikkeling: breadboard		
	Aantal: 1 MUST-HAVE	€20,29

Ontwikkeling: breadboard draadjes		
	Aantal: 3 MUST-HAVE	€6,95
Gereedschap: loeplamp		
	NICE-TO-HAVE	€39,95
Solderen: tinzuiger		
	MUST-HAVE	€2,50
Reinigen: isopropanol alcohol		
	Verwijdert flux NICE-TO-HAVE	€7,95

Veiligheid: soldeerdamp afzuiging		
	NICE-TO-HAVE	€31,95

Met bovenstaand materiaal kan men een basis elektronica onderzoekslokaal inrichten. In deze inventaris is geen meubilair opgenomen aangezien standaard tafels en bureaustoelen afdoende zijn. Pas als men langdurig prototypes zou beginnen solderen is het aan te raden een aangepaste soldeerwerktafel aan te schaffen omwille van ergonomische redenen.

Aangezien een IoT laboratorium wordt opgericht is een goede computer en netwerkinfrastructuur zoals eerder besproken een must. Onderstaande componenten zijn meer dan afdoende om een goed labo in te richten en veelal zijn componenten zoals een computer reeds beschikbaar waardoor deze niet moeten worden meegerekend in het budget.

Labopc: Lenovo ThinkPad L540		
	Processor: Intel core i5 4210M Scherm: 1920x1080 LED RAM: 4GB DDR3 Opslag: 500GB HDD + 8GB SSD NICE-TO-HAVE	€963,55
Monitor: BenQ LED Monitor BL2420PT		
	Display: 23,8" LED TFT IPS Paneel NICE-TO-HAVE	€399

Switch: TP-Link TL-SG1024		
	Snelheid: 1Gbit Aantal poorten: 24 MUST-HAVE	€109,90
Router: HP Server Tower ProLiant MicroServer G8 G1610T met pfSense		
	Gigabit custom pfSense router MUST-HAVE	€219,95
Kabel: Cat5e kabel 10m		
	Minimum 10 kabels MUST-HAVE	€7,99
Randapparatuur: microsoft optical mouse 200		
	Bekabelde muis NICE-TO-HAVE	€9,95

Nu bovenstaande extensieve lijst van producten is samengesteld kan een budget nodig om een IoT laboratorium in te richten worden geschat:

- Minimumbedrag met enkel 'must-have' producten: € 1523,73 incl. BTW
- Bedrag met alle 'nice-to-have' producten inbegrepen: € 3200.03 incl. BTW

Er kan dus worden geconcludeerd dat het zelfs met een beperkt budget mogelijk is een goede IoT onderzoeksruimte in te richten.

4 Hands-on cursus

4.1 Inleiding tot basiselektronica

Aan de basis van alle IoT projecten ligt de hardware, dit is net wat IoT onderscheid van puur software development. In enkele hands-on projecten zal de cursist kennis maken met de basisbeginselen van digitale elektronica. Vooraleer er met de hardware aan de slag kan worden gegaan is het noodzakelijk enkele korte maar zeer handige formules en componenten te kennen.

4.1.1 Spanning, stroom, weerstand en vermogen

Net zoals in andere vakgebieden wordt in de elektronica het een en ander op een fysische manier uitgedrukt in grootheden. Tabel 1 geeft een overzicht van de meest voorkomende grootheden in elektrische schakelingen.

Grootheid	Symbool	Eenheid	Afkorting van eenheid
Stroom	I	Ampère	A
Spanning	U	Volt	V
Weerstand	R	Ohm	Ω
Vermogen	P	Watt	W

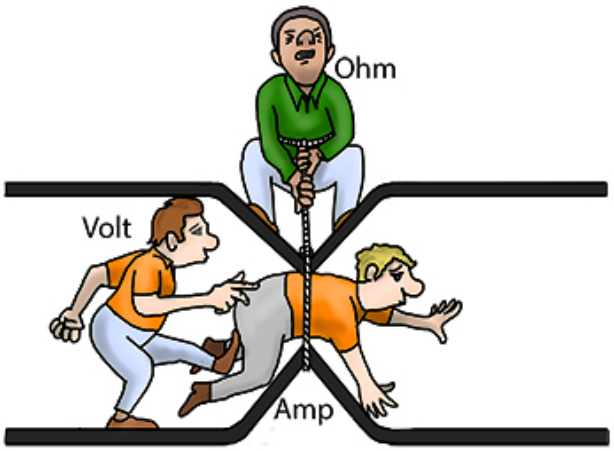
Tabel 1: Veel voorkomende grootheden in de elektronica

Als de cursist nog geen voeling heeft met de begrippen uit tabel 1 kan dit uitgelegd worden aan de hand van een waterleiding met een kraantje. De druk op het water in de leiding of dus hoe snel het water door de leiding vloeit kan men vergelijken met de spanning. De hoeveelheid water die door de leiding stroomt of dus het debiet, kan men het best vergelijken met de stroom. Het kraantje bepaalt de hoeveelheid water dat door de leiding kan stromen en dat komt overeen met de functie van een weerstand in een elektronische schakeling.

Op figuur 13 wordt de analogie met de waterleiding nog eens verduidelijkt. De spanning, de druk op het water, duwt de stroom, de hoeveelheid water, door de weerstand, het kraantje. Hoe hoger de weerstand, dus hoe dichter het kraantje wordt gedraaid, hoe meer spanning er nodig is om dezelfde stroom door de weerstand te krijgen of dus hoe meer druk er op het water moet zitten om dezelfde hoeveelheid door het kraantje te krijgen.

Het vermogen, uitgedrukt in watt, is een afgeleide grootheid van de spanning en de stroom. Het drukt uit hoeveel spanning er op een geleider staat en hoeveel stroom die spanning door de geleider kan krijgen. In hoofdstuk 4.1.4 wordt dit aan de hand van enkele formules verduidelijkt.

Al deze eenheden zijn absolute eenheden, behalve de spanning. Een spanning is namelijk



Figuur 13: Eenvoudige voorstelling van spanning, stroom en weerstand

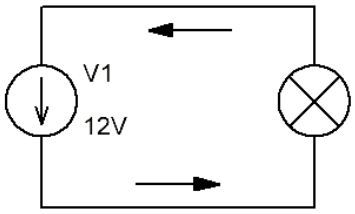
een verschil in potentiële energie of potentiaal tussen twee punten. Daarom zegt men in de praktijk dat een spanning over 2 punten staat, zoals bijvoorbeeld de twee aansluitingen van een batterij. De potentiaal op een bepaald punt kan dus stijgen of dalen en dit heeft als gevolg dat het potentiaalverschil, of dus de spanning, zowel positief als negatief kan zijn.

Aangezien een spanning echter relatief is moet men dus specifieren hoe deze moet worden gemeten, anders kan men niet weten of er een positieve of negatieve spanning wordt bedoeld. Omwille van deze reden staat er op een batterij steeds een plus en min aangeduid. De plus duidt het punt met de hoogste potentiaal aan en de min het punt met de laagste potentiaal. De spanning wordt dan berekend door de laagste potentiaal van de hoogste potentiaal af te trekken.

Als men steeds op dezelfde plaats spanning gaat meten en de spanning verandert niet van teken, ze blijft dus met andere woorden positief of negatief, dan spreekt men van gelijkspanning. Als de spanning wel van teken verandert spreekt men van wisselspanning. Een batterij geeft dus gelijkspanning af, want de positieve en negatieve pool wisselen nooit van kant. Uit een stopcontact komt echter wisselspanning, de spanning verandert daar namelijk 50 of 60 keer per seconde van teken.

4.1.2 Parallel- en serieschakeling

In het vorig hoofdstuk is uitgelegd dat er pas een stroom kan lopen als er een spanningsverschil is tussen twee punten. Een elektrische schakeling kan dus pas werken als er een verbruiker is welke stroom verbruikt waarover een spanning staat. De spanning wordt dan door een spanningsbron, zoals een batterij, veroorzaakt. Op figuur 14 wordt de spanningsbron links weergegeven en de verbruiker rechts.
De spanningsbron V1 zorgt ervoor dat er een spanning van 12V wordt geplaatst over de

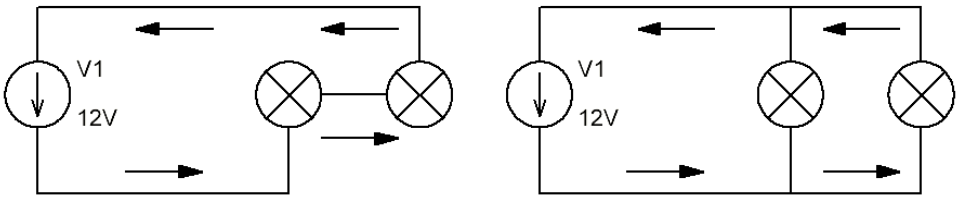


Figuur 14: Een eenvoudige stroomkring met de spanningsbron links en de verbruiker rechts

aansluitingen van de verbuiker. Daardoor gaat er een stroom lopen welke wordt aangegeven door de pijlen. Het is duidelijk dat er geen stroom meer kan lopen moest één van de twee verbindingen van de spanningsbron naar de stroombron worden doorgeknipt.

Wat op figuur 14 wordt weergegeven is de meest eenvoudige vorm van een stroomkring. Het is heel belangrijk dat de student onthoudt dat er dus twee connecties moeten zijn om de stroomkring te sluiten, enkel zo zal de verbruiker kunnen werken.

De zaak wordt complexer als er meerdere gebruikers worden aangedreven van een zelfde spanningsbron. Er zijn in dat geval 2 mogelijkheden welke beide op figuur 15 worden weergegeven.



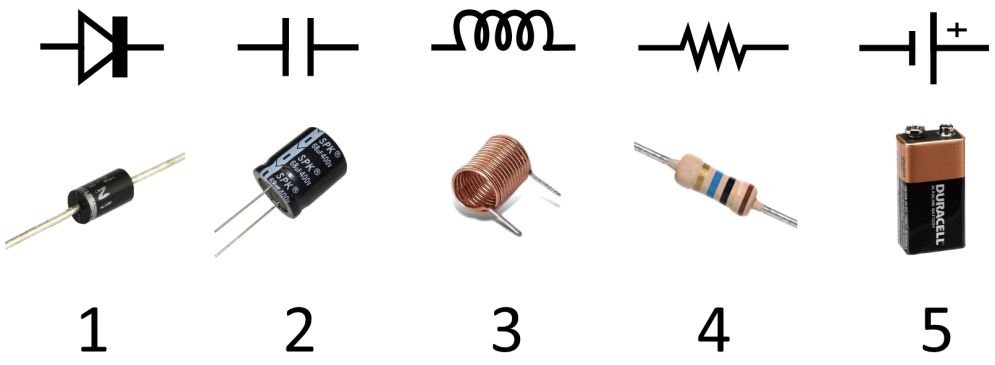
Figuur 15: Links een serieschakeling en rechts een parallelschakeling

De eerste mogelijheid is een zogenaamde serieschakeling. Dit is het geval als de spanningsbron enkel een spanning kan plaatsen over beide gebruikers tegelijk en de verbruikers onder elkaar onzichtbaar verbonden zijn voor de spanningsbron. Voor de spanningsbron lijkt het dus alsof er slechts één grote verbruiker aan hangt. Een dergelijke opstelling kan men in de praktijk terugvinden in een serie kerstlampjes. De spanning komt uit het stopcontact en het ene lampje wordt doorgelust naar het andere lampje wat op zijn beurt terug wordt doorgelust naar het volgende lampje tot het uiteinde de stroomkring sluit en terug in het stopcontact gaat.

De andere mogelijkheid wordt een parallelschakeling genoemd. De spanningsbron ziet in dit geval wel alle verbruikers en de spanning over alle gebruikers is dus dezelfde. Hoewel dit in de praktijk minder goed zichtbaar is zijn de stopcontacten in een woning hiervan een goed voorbeeld. De spanning op elk van de stopcontacten is gelijk aan 230 volt wisselstroom,

ongeacht welke verbruikers er op zijn aangesloten.

4.1.3 Elektronica componenten



Figuur 16: Veel voorkomende elektronica componenten: diode (1), condensator (2), spoel (3), weerstand (4), spanningsbron (5)

Elektronica is opgebouwd uit verschillende componenten, op figuur 16 worden enkele van de meest voorkomende weergegeven. De transistor wordt in dit document niet verder besproken omdat dit te ver zou leiden. Het is belangrijk te onthouden dat de transistor een elektronische schakelaar is waar de stroom in één welbepaalde richting door kan stromen. Transistoren zijn er in heel veel soorten maar het meest gebruikte type zijn tegenwoordig de MOSFET's (Metal-Oxide-Semiconductor Field-Effect Transistor) welke in veel IC's en alle processoren worden gebruikt.

De componenten weergegeven op figuur 16 zijn:

1. **Diode**: de diode is de meest simpele 'semiconductor' of halfgeleider. De stroom kan er slechts in één richting doorvloeien. Omwille van deze eigenschappen worden diodes onder andere gebruikt om van wisselspanning gelijkspanning te maken, om te beschermen tegen ompoling, om spanningspieken op te vangen, ...
2. **Condensator**: een condensator kan het best worden vergeleken met een batterij die zeer snel kan op- en ontladen worden. Bij een spanningsval zal de condensator hiervoor energie afgeven en bij een spanningspiek zal de condensator energie opnemen. Deze eigenschappen maken de condensator zeer geschikt om signalen af te vlakken en scherpe randen van een spanning vlakker te maken. De hoeveelheid lading die een condensator kan opslaan wordt uitgedrukt in Farad.
3. **Spoel**: een spoel is niets meer dan een gewikkelde draad. Door het wikkelen, al dan niet rond een kern, worden echter opmerkelijke eigenschappen waargenomen. Waar

een condensator steeds een zelfde spanning zal nastreven zal een spoel steeds dezelfde stroom nastreven. Ook een spoel kan dus energie opslaan en deze hangt af van de zelfinductie van de spoel die in Henri wordt uitgedrukt. Spoelen worden onder andere gebruikt in filters, transformatoren, schakelende voedingen, ...

4. **Weerstand**: dit component biedt, zoals de naam reeds zegt, een weerstand tegen stroom. Alle stroom die er niet door raakt wordt omgezet in warmte. Weerstanden worden zeer veel gebruikt om verschillende spanningen te maken, stromen te limiteren, ingangen te beschermen, ...

5. **Spanningsbron**: de gelijkspanningsbron is in de IoT elektronica wellicht de meest voorkomende spanningsbron en kan als een batterij worden gezien. Een spanningsbron zal altijd een zelfde spanning, bijvoorbeeld 5 volt, afgeven. De tegenganger van de spanningsbron is de stroombron welke een constante stroom zal afgeven.

4.1.4 De wet van Ohm en vermogen

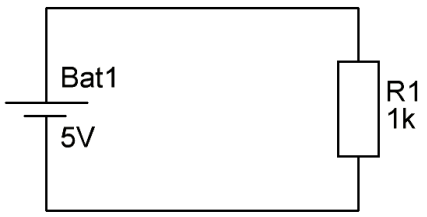
Zelfs om simpele schakelingen te ontwerpen is het vereist de wet van Ohm te kennen.

$$R \text{ (weerstand - } \Omega) = \frac{U \text{ (spanning - V)}}{I \text{ (stroom - A)}} \tag{1}$$

In vergelijking 1 wordt de wet van Ohm die stelt dat de weerstand gelijk is aan de verhouding van spanning op stroom weergegeven. Vergelijking 2 geeft weer hoe de spanning en stroom component samen het elektrisch vermogen vormen.

$$P \text{ (vermogen - W)} = U \times I \tag{2}$$

Om deze wetten in te oefenen kunnen enkele eenvoudige schakelingen worden geanalyseerd. Op figuur 17 is een eenvoudig schema te zien bestaande uit een spanningsbron van 5 volt en



Figuur 17: Elektronisch schema met een spanningsbron en een weerstand

een weerstand van 1kΩ. Met behulp van formule 1 kan gemakkelijk worden berekend wat

de stroom is die door de weerstand loopt:

$$\begin{aligned} R &= \frac{U}{I} \\ \Leftrightarrow I &= \frac{U}{R} \\ \Rightarrow I &= \frac{5V}{1000\Omega} = 0,005A \end{aligned}$$

Door weerstand R1 loopt dus een stroom van 5mA. Aangezien hier alle stroom door de weerstand loopt zal de weerstand al deze stroom omzetten in warmte. Met behulp van vergelijking 2 kan worden berekend hoeveel vermogen er aan warmte wordt gedissipeerd:

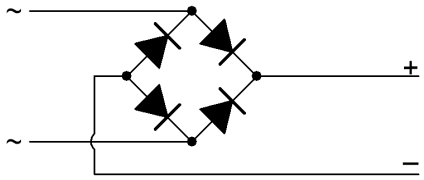
$$P = U \times I \Rightarrow P = 5V \times 0,005A = 0,025W$$

De simpele elektronische schakeling in figuur 17 verbruikt dus 25mW en geeft dit allemaal af als warmte.

Met de wet van Ohm en de formule voor het bereken van vermogen kan de beginnende elektronicus al zeer veel berekenen en daarom is het belangrijk dat de cursist deze formule goed kan toepassen.

4.1.5 Diodes en LED's

LED is de afkorting van 'Light Emitting Diode'. Een LED gedraagt zich exact zoals een gewone diode maar straalt daarnaast ook licht uit. Een diode is de meest eenvoudige halfgeleider en is een component die de stroom slechts in één richting doorlaat. Een diode zal dus stroom tegenhouden in één richting maar doorlaten in de andere richting.



Figuur 18: Een bruggelijkrichter

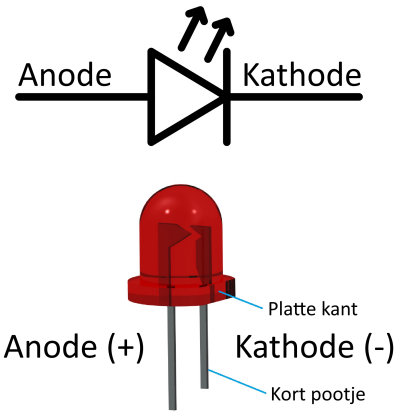
Op het eerst lijkt dit niet nuttig te zijn, maar een praktijkvoorbeeld waarin het nut hiervan meteen blijkt is een gelijkrichter. Bijna alle moderne toestellen zoals smartphones, laptops, televisies, ...werken op gelijkspanning. Toch is de spanning die uit het stopcontact komt wisselspanning, er moet dus een omvorming gebeuren van wisselspanning naar gelijkspanning en dat is waar een simpel component zoals een diode aan te pas komt.
Op figuur 18 wordt voorgesteld hoe men met 4 diodes een wisselspanning kan omvormen naar een gelijkspanning. Een diode wordt schematisch voorgesteld als een pijl die naar een streepje wijst. De pijl geeft aan hoe de stroom door de diode kan lopen van een hogere naar

een lagere potentiaal. Als er dus aan de kant van de pijl een hogere potentiaal staat dan aan de kant van het stokje kan de stroom door de diode lopen. Als de potentiaal echter hoger is aan het stokje dan aan de kant van de pijl zal er geen stroom door de diode lopen. De wisselspanning kan men nu opdelen in twee delen:

- Het deel waarbij de bovenste aansluiting een hoger potentiaal heeft dan de onderste aansluiting. In dit geval zal de hoge potentiaal dus enkel een stroom kunnen laten lopen door de diode rechts boven en de hoge potentiaal komt dus uit aan de '+' aansluiting. De laagste potentiaal zal enkel een stroom kunnen laten lopen door de diode links onderaan en dus zal de lage potentiaal uitkomen aan de '-' aansluiting.
- Het deel waarbij de onderste aanslutiing een hoger potentiaal heeft dan de bovenste aansluiting. In deze situatie wordt het omgekeerd en de hoge potentiaal zal enkel een stroom kunnen laten lopen via de diode rechts onderaan en dus uitkomen aan de '+' aansluiting. De laagste potentiaal zal een stroom kunnen laten lopen door de diode links bovenaan en zal dus uitkomen aan de '-' aansluiting.

Het maakt met andere woorden niet uit op welke van de twee linkse aansluitingen de hoogste of de laaste potentiaal staat. Aan de gelijkgerichte uiteinden van de diodebrug zal de stroom altijd in dezelfde richting lopen en zal de spanning dus altijd hetzelfde teken hebben.

Een belangrijk kenmerk van een diode en dus ook van een LED is de voorwaartse spanningsval. Ook al staat de diode zo geschakeld dat de stroom er door kan lopen, toch zal er na de diode een klein beetje spanning verdwenen zijn. Dit wordt in de diode omgezet in warmte en dus gedissipeerd. Deze voorwaartse spanningsval wordt als V_f afgekort en bedraagt bij een standaarddiode meestal 0,7 volt.



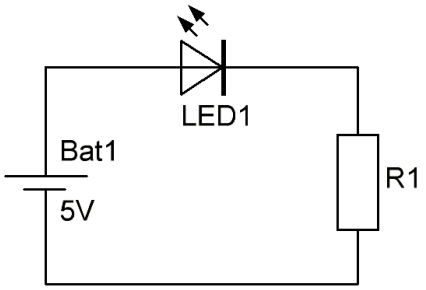
Figuur 19: Schematisch voorstelling van een LED (bovenaan) en afbeelding van een LED (onderaan)

Figuur 19 geeft weer hoe een LED wordt aangeduid in een schema. De voorwaartse spanningsval V_f van een LED is veel hoger dan deze van een normale diode en hangt sterk af van

de kleur van de LED, enkele typische waardes zijn:

- Rode LED's: tussen 1,5V en 2,2V
- Groene LED's: tussen 2,0 en 3,0V
- Blauwe LED's: tussen 2,5V en 3,5V of zelfs meer
- Gele LED's: tussen 1,5V en 2,5V
- Witte LED's: tussen 3,5 en 4V

Een LED laat net als een gewone diode zo veel mogelijk stroom door zich lopen indien dit in de juiste richting gebeurt. Als er dus een spanningsbron op de goede manier rechtstreeks op een LED zou worden aangesloten kan de stroom theoretisch gezien oplopen tot oneindig veel ampère. In de praktijk mag dit echter niet gebeuren omdat een diode kapot gaat als de stroom die erdoor loopt te hoog wordt. Er bestaan diodes die honderden ampères kunnen verdragen maar aangezien een diode gebruikt wordt om licht te geven streeft men er naar om zo veel mogelijk licht te genereren bij een zo klein mogelijke stroom. Dit zorgt ervoor dat de stroom die LED's kunnen verdragen relatief laag is en zich typisch tussen de 5mA en 70mA situeert.



Figuur 20: LED met voorschakelweerstand

Om een LED te voeden uit een spanningsbron, zoals een batterij, moet de stroom dus worden beperkt. Uit hoofdstuk 4.1.1 wordt duidelijk dat er een weerstand moet worden geplaatst tussen de LED en de spanningsbron, dit is te zien op figuur 20. Een weerstand die gebruikt wordt om de stroom door een ander component te beperken noemt men een voorschakelweerstand.

De voorschakelweerstand R1 kan met de wet van Ohm berekend worden indien volgende twee gegevens van de LED bekend zijn:

- De voorwaartse spanningsval V_f
- De gewenste stroom door de LED, afgekort door I_f (Forward current)

Stel dat LED1 de parameters $V_f = 2V$ en I_f=15mA heeft dan kan de voorschakel weerstand R1 in volgende stappen worden berekend:

1. Bepaal de spanning die over de weerstand R1 staat door rekening te houden met de spanning die over LED1 valt:

$$V_{R1} = V_{Bat1} - V_f = 5V - 2V = 3V$$

2. Bereken met behulp van de wet van Ohm hoe groot de weerstand moet zijn om de gewenste stroom I_f door het circuit te laten lopen:

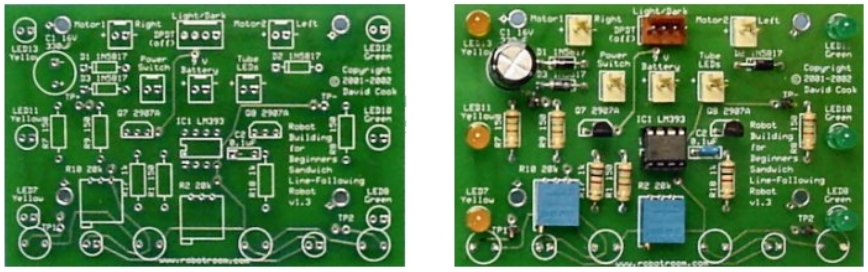
$$R = \frac{U}{I} = \frac{V_{R1}}{I_f} = \frac{3V}{0.015A} = 200\Omega$$

De voorschakel weerstand zal dus veranderen als de voedingsspanning van een circuit verandert. Het niet limiteren van de LED kan ernstige gevolgen hebben. Zo kan de spanningsbron die de LED voedt kapot gaan omdat deze een kortsluiting ervaart. In het geval dat deze spanningsbron een batterij is valt de schade mee maar als de spanningsbron een embedded computer of microcontroller is kan dit grote problemen veroorzaken.

4.2 Hands-on 1: LED schakelen met drukknop

4.2.1 Opbouwen van een circuit

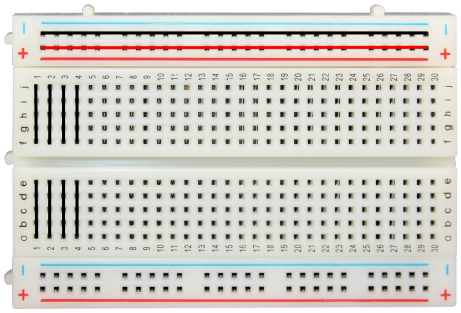
In professionele producten, maar tegenwoordig ook steeds meer in de hobby- en prototype sfeer, worden elektronische schakelingen met behulp van een 'PCB' of 'Printed Circuit Board' gebouwd. Dit zijn platen waarop kopersporen de verbindingen tussen de componenten vormen. Op figuur 21 staat een voorbeeld van een simpele printplaat. De groene kleur die



Figuur 21: PCB zonder componenten (links) en met componenten (rechts).

veelal met een PCB wordt geassocieerd is niet meer dan een coating op de printplaat welke ervoor zorgt dat het solderen van de contacten makkelijker gaat, deze coating wordt dan ook het soldeermasker genoemd. Het masker kan echter om het even welke kleur hebben. Een printplaat kan zeer complex worden en meerdere lagen koperbaantjes bevatten. De meest complexe printplaten bevatten tot wel 20 verschillende koperlagen.

Een printplaat moet ontworpen worden op de computer en het duurt enkele dagen om deze in een fabriek te laten maken. Als de schakeling eenvoudig is kiest men daarom vaak voor een andere bouwmethode om het allereerste prototype te bouwen, namelijk met behulp van een breadboard. Een breadboard maakt het mogelijk om zonder solderen componenten met pootjes, de zogenaamde through-hole componenten, met elkaar te verbinden.



Figuur 22: Een breadboard met aanduiding hoe de verbindingen intern worden gemaakt.

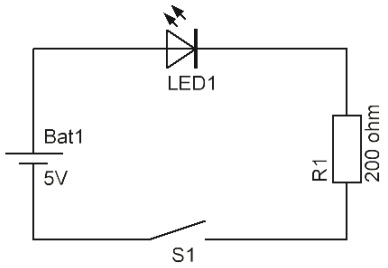
Op figuur 22 wordt een breadboard afgebeeld. De buitenzijden van het breadboard worden gebruikt om de schakeling te voeden en daar wordt dan ook de voedingsspanning op aangesloten. De voedingsrails lopen over de gehele buitenzijde van het breadboard maar twee zijden zijn niet met elkaar verbonden. De reden dat de beide zijden niet verbonden zijn is om met twee verschillende spanningen te kunnen werken, bijvoorbeeld 5V en 3.3V.

De componenten worden midden op het breadboard geplaatst. In het midden lopen de verbindingen steeds parallel naast elkaar, in de praktijk zal blijken dat deze opstelling het heel gemakkelijk maakt om verschillende schakelingen op te bouwen. Als verschillende kolommen van het breadboard met elkaar moeten worden verbonden wordt dit met 'jumper wires' gedaan, dit zijn kleine korte kabeltjes met harde puntjes die gemakkelijk op het breadboard passen.

4.2.2 De schakeling

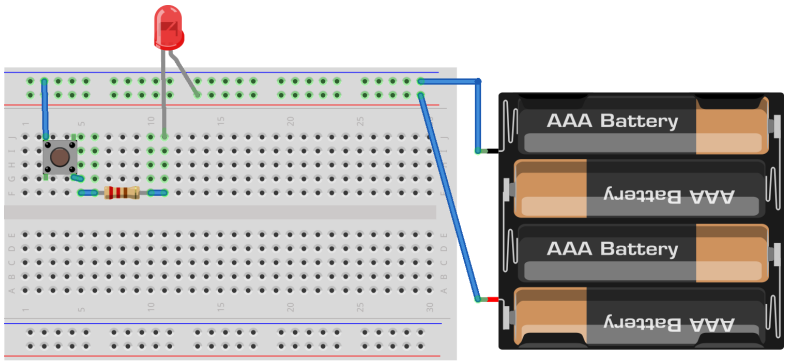
Om het breadboard te leren kennen wordt als eerste schakeling een LED geschakeld met behulp van een schakelaar. De student zou met de kennis die tot nu toe in de cursus is gegeven in staat moeten zijn een schema te tekenen zoals op figuur 23.

Als drukknop S1 wordt ingedrukt wordt de stroomkring gesloten en kan er stroom door de LED lopen waardoor deze oplicht. Weerstand R1 limiteert de stroom die door de LED loopt, indien deze niet zou worden geplaatst zou er oneindig veel stroom gaan lopen waardoor de LED kapot zou branden.



Figuur 23: Schema om een LED met een drukknop te schakelen

Op figuur 24 wordt weergegeven hoe deze schakeling op een breadboard kan worden geplaatst. Dit is slechts één van de vele mogelijkheden, belangrijk is dat men de voedingsrails langs de zijkant van het breadboard gebruikt om de spanning aan te sluiten. In deze schakeling luisterd dat niet zo nauw, maar in meer complexe schakelingen is dat zeer belangrijk om het overzicht te bewaren.



Figuur 24: Opbouw van de LED met drukknop schakeling op een breadboard

Op het eerste zicht kan deze hands-on zeer simpel lijken, maar voor studenten die voor de eerste keer een breadboard schakeling maken is dit een ideale kennismaking. Zo is het niet steeds duidelijk hoe de stroom loopt en waarom er een weerstand nodig is.

4.3 Hands-on 2: Simon Says

4.3.1 Overzicht

Leerlingen worden altijd enthousiast als het over spelletjes gaat. In deze hands-on wordt dan ook met behulp van de IoT-kit het spel 'Simon Says' gebouwd. Dit spel bestaat uit 4 gekleurde LED lampjes met per lampje een drukknop.

Als het spel wordt gestart laat de software een willekeurige sequentie lampjes oplichten, bijvoorbeeld volgende sequentie:

```
[LED1, LED2, LED3]
```

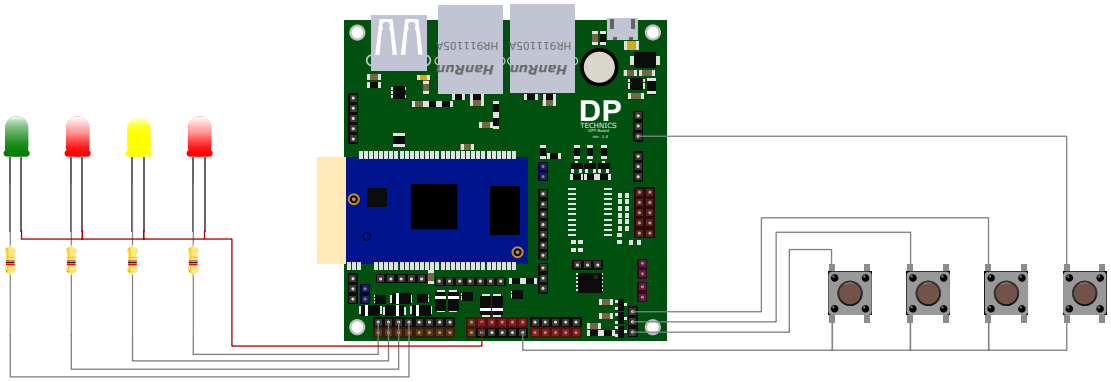
Het is de bedoeling dat de gebruiker nu dezelfde sequentie met de drukknoppen herhaalt. De speler wint dus het spel als hij de combinatie

```
[KNOP1, KNOP2, KNOP3]
```

ingeeft en de speler verliest indien hij een andere combinatie indrukt. Als de speler een spelletje wint is het de bedoeling dat hij/zij naar het volgende level gaat. Het volgende level wordt een stapje moeilijker omdat de sequentie dan 1 stap langer wordt. Stel dat bovenstaande sequentie level 1 was dan zouden er in level 2 niet drie maar 4 lampjes moeten oplichten.

4.3.2 Schema

Het schema voor het Simon Says spelletje wordt weergegeven op figuur 25. De LEDs worden



Figuur 25: De schematische opbouw van het Simon Says spelletje

in deze schakeling aangestuurd door mosFET's die op het DPT-Board geplaatst zijn. De mosfets schakelen de 'ground'. Dit wil zeggen dat de positieve +5V steeds aangesloten is, maar het DPT-Board de negatieve pool (0V) schakelt.

Om een schakelaar en/of drukknop in te lezen moet er normaal gezien met een pull-up of pull-down weerstand worden gewerkt. Deze weerstand zorgt ervoor dat de ingang van het DPT-Board niet gaat zweven als de schakelaar open is. Het handige aan het DPT-Board is dat deze al ingebouwde pull-up weerstanden heeft. Het enige wat een schakelaar dus moet doen is de ingang van het DPT-Board verbinden met de ground.

4.3.3 Het DPT-Board opstarten

Het hardware gedeelte is nu volledig afgewerkt. Uiteraard doet de hardware niets zonder dat er softwarecode wordt ingeladen. Het schrijven van software voor het DPT-Board gebeurt in Javascript via de DPT-WEB-IDE. Om deze IDE op te starten moet het DPT-Board opgestart worden door een micro-USB kabel aan te sluiten tussen een GSM-lader of USB poort op de computer en het DPT-Board. Het groende LED-lampje op het board zal beginnen met knipperen en van zodra het stopt met knipperen is het ontwikkelbord volledig opgestart.

Van zodra het DPT-Board is opgestart zal het een WiFi netwerk uitzenden met de naam 'DPT-Board'. De student kan nu met een laptop of tablet verbinden met dit netwerk en surfen naar 192.168.1.1 waarna de DPTechnics webinterface zichtbaar wordt. Vanuit dit programma moet men navigeren naar

`Develop > Code editor`

Om te beginnen met programmeren moet er een nieuw bestand worden aangemaakt. Daarvoor moet de student kiezen voor `File > New File`, hij/zij krijgt dan het venster te zien zoals op figuur 26. Hier moet voor het type javascript worden gekozen en een willekeurige naam mag worden gekozen.

Figuur 26: Pop-up venster waarin een nieuwe file wordt aangemaakt

4.3.4 De programmacode

In bijlage 8.1 wordt de volledige programmacode voor het spel weergegeven. Deze code is standaard JavaScript zoals men het in een moderne browser schrijft, maar er zijn enkele standaard beschikbare functies die door DPTechnics zijn toegevoegd aanwezig. De meest belangrijke functies die zijn toegevoegd zijn:

- `print(object)`: in een browser wordt de functie `console.log(object)` gebruikt om output te schrijven naar de debug console. In de JavaScript variant op het DPT-Board is die functie vervangen door de functie `print`.
- `digitalWrite(io_number, true/false)`: met de `digitalWrite` functie kan een uitgang van het DPT-Board worden in- of uitgeschakeld. De eerste parameter is het poortnummer dat de functie moet bedienen en de tweede parameter is een booleaanse waarde die aangeeft of de uitgang ingeschakeld (`true`) of uitgeschakeld (`false`) moet zijn.

- `digitalRead(io_number)`: deze functie is de tegenganger van de `digitalRead` functie en aanvaardt als enige parameter het poortnummer waarvan de programmeur de status wil uitlezen. De functie geeft `true` terug als de ingang hoog is, of dus als er spanning op staat, en `false` als de ingang laag is, of dus als er geen spanning op staat.

Nadat de programmacode in de WEB-IDE is geschreven kan men op de play knop duwen om deze uit te voeren. Als alles goed gegaan is heeft de student nu een volledig werkend Simon Says game gebouwd.

Indien er toch foutmeldingen zouden optreden dan worden deze onder aan de IDE weergegeven. Om te lezen wat er geprint is met de `print` functie kan er op de 'console' knop worden gedrukt. De berichten die men heeft uitgeprint verschijnen dan onderaan op het scherm.

4.4 Hands-on 4: Een LCD aansturen

4.4.1 Inleiding

Een embedded toestel moet ook vaak gegevens aan de gebruiker presenteren. De goedkoopste manier om dit in een toestel te verwezenlijken is door gebruik te maken van LED lampjes. De hoeveelheid informatie die daarmee kan worden voorgesteld is echter vrij gelimiteerd. Daarom worden veel toestellen voorzien van een zogenaamd 16×2 LCD scherm.

Dit scherm, die ook in de IoT-kit zit, heeft een ingebouwde controller-chip waarnaar karakters kunnen worden gezonden. De aanduiding 16×2 duidt op het feit dat het scherm 16 kolommen heeft en 2 rijen, dus 32 posities in totaal. Op iedere positie past juist één karakter. In deze hands-on zal de student leren hoe hij/zij het LCD scherm moet aansluiten en hoe er tekst op het scherm kan worden weergegeven.

Dergelijke LCD schermen worden meestal door een 4-draads of 8-draads interface aangestuurd. Dit zou er echter voor zorgen dat bijna alle I/O pinnen in gebruik zouden zijn waardoor het board niets anders meer zou kunnen bedienen. Daarom wordt er gebruik gemaakt van een I^2C port expander, namelijk de PCF8574 van Phillips/NXP. Deze chip heeft 8 I/O pinnen die worden gecontroleerd door middel van digitale communicatie over de I^2C bus. Daardoor moeten er slechts 2 aansluitingen, namelijk serial data (SDA) en serial clock (SCL), worden aangesloten op het DPT-Board.

4.4.2 De I^2C-bus en seriele communicatie

In IoT en andere elektronische toestellen zitten er tegenwoordig veel verschillende systemen en geïntegreerde schakelingen. Deze systemen moeten met elkaar kunnen communiceren en hiervoor wordt zeer vaak seriële communicatie gebruikt. Dit type communicatie zendt alle bits achter elkaar op het verzendkanaal. De manier waarop dit gebeurt wordt beschreven in het communicatie protocol. Er zijn drie protocollen om tussen verschillende chips op de zelfde printplaat te communiceren die vaak voorkomen:

- **SPI** of **Serial Peripheral Bus**: dit is een synchroon serieel protocol waarbij de zendontvangers via 3 kabels met elkaar worden verbonden, namelijk:

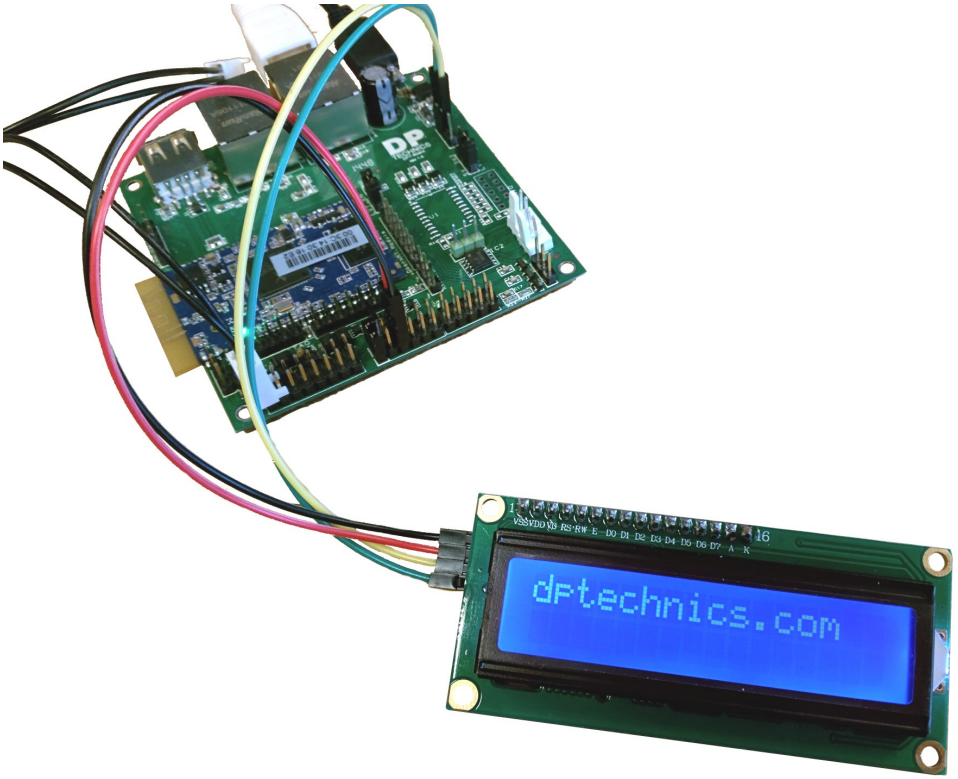
  1. MOSI (Master Out Slave In): over deze kabel loopt data van de master naar een slave.
  2. MISO (Master In Slave Out): over deze kabel loopt de data van de slave naar de master.
  3. Clock: via deze kabel wordt het kloksignaal dat de master genereert naar de slaves gezonden.

  De begrippen 'master' en 'slave' zijn veel voorkomende begrippen bij communicatie in een bus-topologie. Een bus is een systeem waarbij er meerdere zendontvangers verbonden zijn via dezelfde kabels, dit in tegenstelling tot een point-to-point verbinding waar er slechts 2 zendontvangers rechtstreeks met elkaar verbonden zijn. De master selecteert met welke slave hij wil communiceren door middel van een 'chip select' verbinding. Naast de MISO, MOSI en CLOCK lijnen moet er bij SPI dus nog een aparte CHIP SELECT lijn worden verbonden tussen de master en de slave. Als er heel veel slaves zijn zal SPI dus nog altijd vrij veel verbindingen nodig hebben.

- **I^2C**: ook dit is een synchroon serieel protocol waarbij er 2 fysieke verbindingen nodig zijn tussen de zendontvangers. De eerste datalijn wordt 'Serial CLock' (SCL) genoemd en de tweede datalijn wordt 'Serial DAta' SDA genoemd. Er kunnen tot 127 verschillende zend-ontvangers op dezelfde 2 fysieke kabels worden aangesloten, het is dus ook een bus-topologie. Er wordt één specifieke zendontvanger tot master gepromoveerd, enkel deze master mag de SCL-lijn bedienen. Waar er bij SPI wordt gewerkt met een apparte verbinding om de slave te selecteren wordt dit bij I^2C opgelost door elke slave een 7-bit adres toe te kennen. Ieder bericht op de I^2C-bus wordt door dat adres vooraf gegaan waardoor de slave weet of hij het bericht moet verwerken of niet.

- **RS-232 (TTL)**: RS-232 is de meest voorkomende asynchrone seriële verbinding. Het feit dat deze verbinding asynchroon is duidt op het feit dat de zendontvangers zelf hun kloksignaal opwekken. Er zijn slechts twee fysieke verbindingen en dat zijn de datasignalen RX (Receive) en TX (Transmit). De snelheid van de communicatie moet vooraf worden afgesproken, in de context van seriële verbindingen wordt deze snelheid in 'baud' uitgedrukt. Deze manier van communiceren wordt ook vaak gebruikt voor communicatie tussen verschillende printplaten en op RS-232 signaalniveau kan de kabel tussen de zendontvangers tot wel 10 meter lang worden.

Er kan nog veel meer verteld worden over seriële communicatiebussen maar dit valt buiten de scope van een inleidende cursus. De softwarebibliotheken van DPTechnics zorgen ervoor dat de communicatie met bijvoorbeeld I^2C slaves zeer eenvoudig kan worden opgezet. Ook voor het aansturen van het LCD-scherm via de PCF8574 port expander zijn alle bibliotheken voor handen.



Figuur 27: Het aansluitschema voor het LCD scherm

4.4.3 Het schema

De aansluitingen voor deze hands-on zijn zeer eenvoudig en er dienen slechs 2 datalijnen te worden aangesloten. De SCL datalijn van het LCD moet aan IO20 worden gekoppeld en de SDA datalijn van het LCD moet aan IO23 worden gekoppeld. Het is ook zeer belangrijk om de LVLS_ON en LVLS_5V jumpers te plaatsen. De I^2C bus heeft enkel deze twee aansluitingen nodig om te functioneren.

4.4.4 De programmacode

In bijlage 8.2 wordt de code beschreven die nodig is om het LCD-scherm aan te sturen. De code begint met het initialiseren van de I^2C-bus, daarvoor heeft de DPTechnics bibliotheek een `I2C` prototype in de bibliotheek zitten. Bij het aanmaken van het `I2C` object moet worden opgegeven welke I/O poorten van het DPT-Board de SCL en SDA lijn moeten worden.

Als de I^2C-bus goed is geïnitialiseerd moet ook het LCD scherm worden geïnitialiseerd zoals in het codevoorbeeld. Vanaf dan kan er op het LCD worden geschreven met alle beschikbare LCD functies, een overzicht van deze functies wordt weergegeven in bijlage 8.3. Het meest

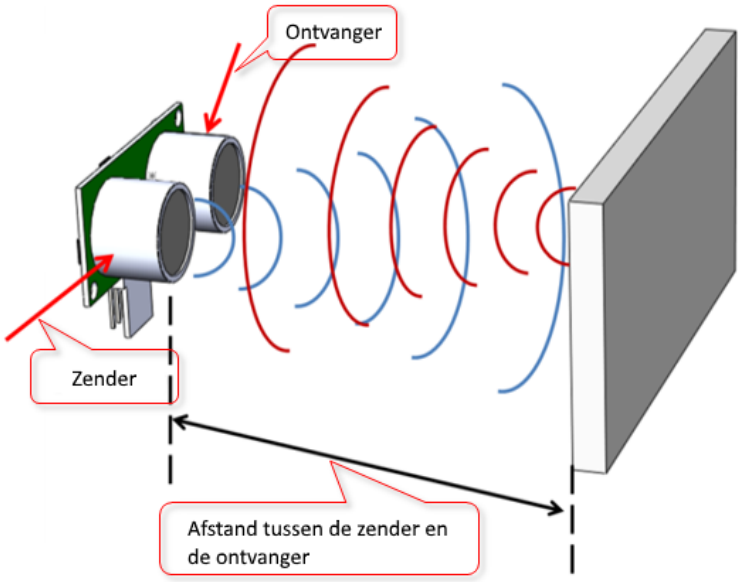
belangrijke om te weten is dat dergelijke LCD's verwachten dat de cursor op een bepaalde positie wordt geplaatst vooraleer er tekst op het scherm wordt geprint. Door de cursor na het printen opnieuw op dezelfde positie te plaatsen zal de tekst op het LCD-scherm overschreven worden.

4.5 Hands-on 3: Parkeersensoren

4.5.1 Overzicht

Een belangrijk onderdeel van IoT-toestellen bestaat uit het kunnen waarnemen van de omgeving. Elektronica die het mogelijk maakt voor computers om de buitenwereld te kunnen voelen worden sensoren genoemd. Een drukknop is de meest eenvoudige sensor die er bestaat en de student heeft dit type sensor reeds gebruikt in hands-on 2 om input te 'voelen' van de personen die het spel spelen.

In deze hands-on wordt er een andere sensor gecombineerd met het LCD scherm om een parkeersensor te maken, namelijk een ultrasone afstandssensor. De software zal de sensor uitlezen een waarschuwing op het scherm plaatsen als de afstand kleiner wordt dan 10 centimeter. Als de gebruiker op een knop drukt gaat de parkeersensor uit en komt er op het scherm 'Inactive' te staan.



Figuur 28: Werking van een ultrasone afstandssensor

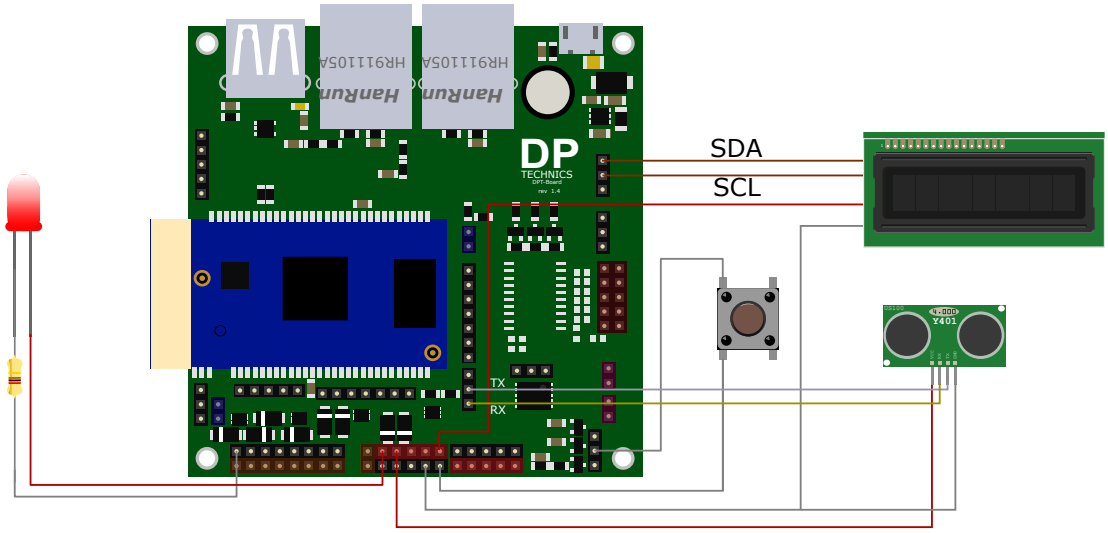
De ultrasone sensor werkt met een zender en een ontvanger die vlak naast elkaar gemonteerd zitten op een printplaat. Van zodra er aan de sensor een afstandsmeting wordt gevraagd zal

de sensor een ultrasone puls uitzenden. Deze puls zal terugkaatsen op het object tot waar men de afstand wil meten en opgevangen worden door de ontvanger. Aangezien de snelheid van het geluid gekend is kan men de afstand nu gemakkelijk berekenen aan de hand van de tijd die er zat tussen het verzenden en ontvangen van de puls.

In de US-100 afstandsensor die bij de IoT-kit zit is ook een temperatuursensor geïntegreerd. Het is namelijk zo dat de snelheid van het geluid afhangt van de temperatuur, de sensor in de kit houdt hier rekening mee waardoor veel betere resultaten worden bekomen dan met een sensor zonder thermometer.

4.5.2 Het schema

De US-100 afstandssensor wordt aangesloten op de UART (Universal Asynchronous Receiver/Transmitter) van het DPT-Board en het LCD scherm op de I^2C-bus. In het schema op figuur 29 staat aangegeven hoe alles verbonden moet worden, hierop staan ook de drukknop en het LED waarschuwingslampje afgebeeld.



Figuur 29: Het schema om een parkeersensor te bouwen

4.5.3 De programmacode

In bijlage 8.4 wordt de code voor het bouwen van de parkeersensor overlopen. Het is een samenvatting van de code die in de vorige hands-on voorbeelden is gegeven. Als de student de bovenstaande projecten kon bouwen en programmeren dan is dit project een perfecte oefening om te zien of het integreren van de verschillende componenten goed lukt.

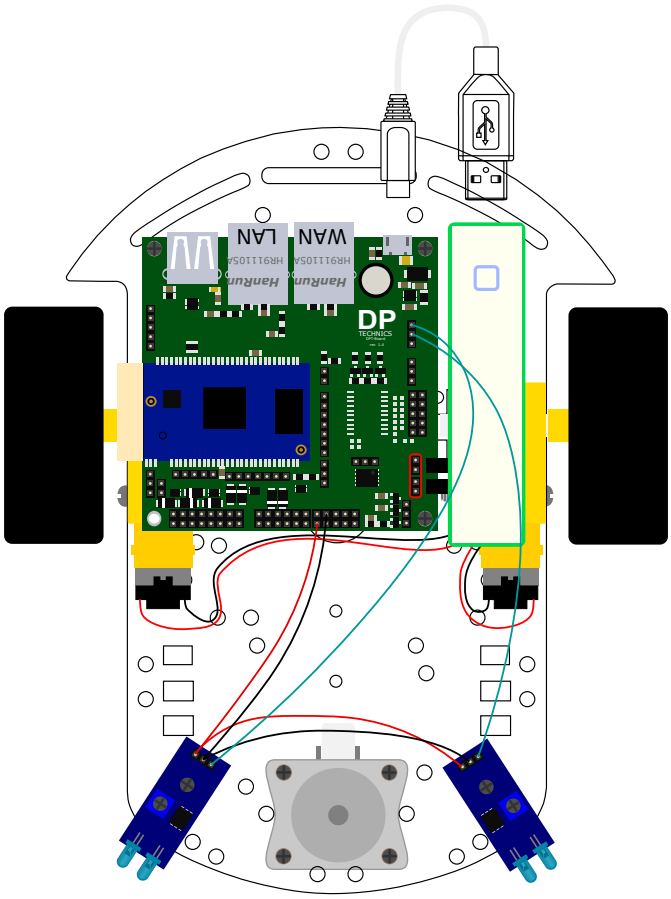
Het gebruik van meerdere programmalussen door middel van de functie `setInterval` valt op in deze code en is eigen aan de programmeertaal JavaScript. Onderliggend worden deze

lussen in slechts één 'event-loop' uitgevoerd. Als de student code voor bijvoorbeeld een microcontroller zou bestuderen die in C is geschreven dan zou men zien dat er slechts één grote programmaloop wordt gebruikt.

4.6 Hands-on 4: Object-avoiding robot

4.6.1 Overzicht

In deze laatste hands-on wordt naast de IoT-kit ook de DPT-Robot gebruikt in combinatie met de twee digitale nabijheidssensoren. De bedoeling van dit project is om de student kennis te laten maken met differentiële sturing. Bij dat type sturing wordt de robot aangedreven en bestuurd door twee afzonderlijk bedienbare motoren. Als de twee motoren tegelijk draaien rijdt de robot rechtdoor, als echter slechts één van de motoren draait neemt de robot een bocht.



Figuur 30: Illustratie van de object-avoiding robot met de nabijheidssensoren

Om ervoor te zorgen dat de robot nergens tegen rijdt moet deze worden voorzien van een aantal sensoren, de IoT-kit voorziet hiervoor 2 nabijheidsschakelaars.

Dit type detector werkt met een zender en ontvanger net zoals de ultrasone afstandssensor maar in plaats van geluidsgolven wordt er licht uitgezonden. De nabijheidssensor meet geen absolute afstand maar geeft een hoog signaal van zodra er een object zich dicht genoeg bij de sensor bevindt. De afstand waarop de sensor een hoog signaal afgeeft moet worden afgesteld door met een schroevendraaier aan het lichtblauwe stelknopje te draaien.

Op figuur 30 staat afgebeeld hoe de object-avoiding robot in elkaar zit. Het DPT-Board wordt geplaatst op de DPT-Robot, daarvoor zijn afstandshouders op het robot frame voorzien. Voor het plaatsen van de sensoren zijn heel wat gaten geboord waardoor de sensoren gemakkelijk kunnen worden bevestigd.

4.6.2 Schema

Het schema om de robot te bouwen is zeer eenvoudig. De nabijheidsschakelaars worden als volgt aangesloten:

- Op IO20 zit de linker afstandssensor aangesloten.
- Op IO23 zit de rechter afstandssensor aangesloten.

Op de daarvoor voorziene witte connectoren worden als laatste de motoren aangesloten.

4.6.3 De programmacode

In bijlage 8.5 staat de programmacode om de robot autonoom te laten rijden. Elke motor kan worden gecontroleerd door het hoog of laag maken van 2 datalijnen, er is één datalijn om de motor linksom te laten draaien en een andere om de motor rechtsom te laten draaien. Als beide datalijnen tegelijkertijd aangezet zouden worden, wordt er een elektrische rem geactiveerd, maar deze functionaliteit wordt niet toegepast in dit project.

Het tweede deel van de code bevat het algoritme die de robot gebruikt om objecten te vermijden. Op basis van de input van de nabijheidsschakelaars zal de robot bepaalde acties ondernemen:

- Als zowel de linker als rechter sensor een signaal detecteren zal de robot achteruit rijden.
- Als slechts één sensor een signaal detecteert zal de robot in de andere richting draaien. Dus als bijvoorbeeld de linkersensor een signaal detecteert zal de robot rechts draaien en omgekeerd.
- Als het draaien in een bepaalde richting te lang duurt zal de robot eerst achteruit rijden tot er niet langer een obstakel zichtbaar is en dan opnieuw draaien.

Dit simpel algoritme zorgt ervoor dat de robot uit de meeste situaties kan wegrijden zonder menselijke interventie. Het is aan de studenten om het basisalgoritme te verbeteren en de robot vloeiender te laten rijden, er zijn voor dit softwarprobleem namelijk zeer veer verschillende correcte oplossingen te bedenken.

5 Integratieproject

5.1 Doel en omschrijving

Het integratieproject is een zeer grote opdracht welke nog niet volledig door een student die deze cursus gevolgd heeft zal kunnen worden gebouwd. Het doel van deze opdracht is dan ook eerder om een praktijkvoorbeeld van een Internet-of-Things product aan te reiken en het volledige ontwerpproces van a tot z voor te stellen aan de student. Met de kennis die in de cursus werd gegeven zou de student in staat moeten zijn alle stappen in de bouw van het project te begrijpen.

In dit project wordt de volledige implementatie van een digitaal leervolgsysteem beschreven. Lectoren in hogescholen en universiteiten moeten vaak werkcolleges begeleiden waarin veel studenten aanwezig zijn. Momenteel is er geen makkelijke manier voor deze begeleiders om snel de leerstatus van de student bij te houden en te verwerken. Het leervolgsysteem die in dit integratieproject wordt voorgesteld lost deze problemen op door het opnemen van de leerstatus zeer eenvoudig te maken en de verwerking van de gegevens volledig te automatiseren.

5.2 Werking

Elke student is uitgerust met een NFC studentenkaart. De gebruikte NFC chip werken volgens het NFC-A principe en bevatten dus een zogenaamd 'tag-ID'. Aan de hand van deze NFC-kaart zullen de lectoren de student uniek identificeren. Verder moeten de lectoren volgende functies kunnen uitvoeren:

- De aanwezigheid van een student opnemen.
- Een score geven tussen 1 en 5 toewijzen aan een student.

Er moet dus een toestel worden ontwikkeld dat deze functies op een zeer eenvoudige manier mogelijk maakt en bovendien goedkoop te produceren is.

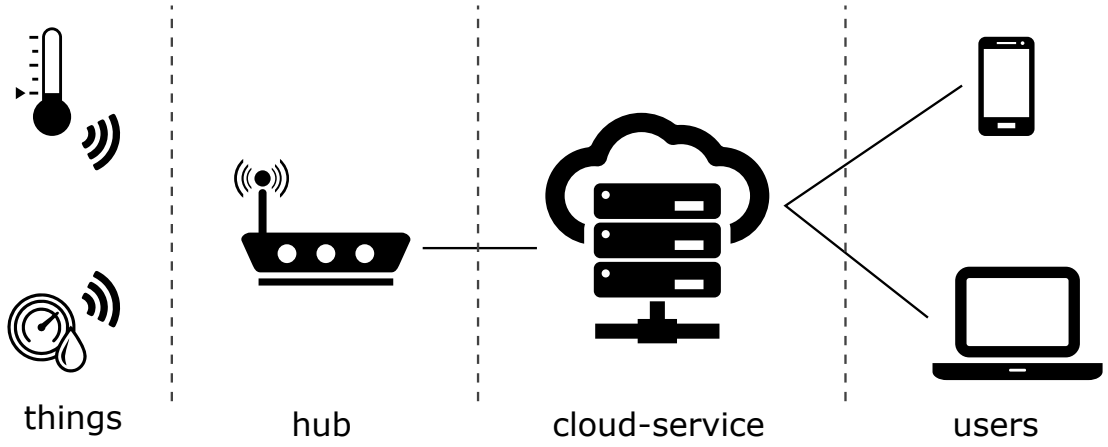
Dit toestel zal worden bijgestaan door een cloud-systeem. Deze cloud-applicatie zal NFC-kaart scans van het handtoestel ontvangen en verwerken tot de nodige rapportages.

5.3 Systeem overzicht

Het leervolgysteem is een zeer mooie illustratie van een klasiek Internet-of-Things systeem bestaande uit een cloud-service en een simpel edge-device. De cloud service zal alle gegevens van de edge toestellen verwerken wat wil zeggen dat deze laag alle business-logic bevat.

In meer complexere systemen zit soms nog een extra laag ingebouwd, deze wordt meestal de 'hub' genoemd. Deze 'hub' is een edge device die genoeg rekenkracht heeft om om een veilige manier te verbinden met de cloud service en tegelijkertijd ook over alle interfaces

beschikt die nodig zijn om met de 'things' zelf te communiceren. Een voorbeeld is een IoT-weerstation waarbij er een basis weerstation via WiFi of ethernet met internet is verbonden. Het weerstation bevat op zijn beurt dan weer een 433MHz transceivers om te verbinden met de sensoren zoals temperatuur, relatieve luchtvochtigheid, ...



Figuur 31: Klassiek gelaagd Internet-of-Things systeem

Op figuur 31 wordt een dergelijk gelaagde IoT-applicatie voorgesteld. De verbinding tussen de 'hub' en de 'things' kan tegenwoordig met heel wat types verbindingen worden verzorgd. Voorbeelden zijn BLE (Bluetooth Low Energy), 433MHz, WiFi, ... De sensoren worden niet rechtstreeks met het netwerk/internet verbonden omdat dit de sensoren onnodig duur en stroomverslindend zou maken. Dit komt omdat er voor een veilige verbinding cryptografische hardware of een krachtige processor aanwezig moet zijn.

Het komt echter ook vaak voor dat alle functionaliteit reeds in de 'hub' zelf vervat zit. De meeste slimme thermostaten zoals de 'Nest Learning Thermostat' of 'Eneco Toon' zijn voorbeelden waarbij de 'things' laag en de 'hub' laag in één toestel vervat zit.

Het leervolgsysteem zal net zoals de slimme thermostaten de 'hub' laag en de 'things' laag integreren in één toestel. De verbinding van het toestel naar de cloud-service zal via WiFi gebeuren daar ongeveer alle publieke gebouwen waar dit system kan worden ingezet volledig van WiFi zijn voorzien. Deze handscanner vormt dus de basis van het IoT-leervolgsysteem en zal beschikbaar moeten worden gesteld aan elke lector die met het systeem aan de slag gaat.

Minstens even belangrijk als de handscanner is het cloud-systeem. Dit zal drie functies vervullen:

- **Implementatie business-logic**: de handscanners zelf zullen geen gegevens verwerken, het zijn als het ware domme devices die de data die ze verzamelen meteen doorsturen

naar de cloud. Het is de cloud die op basis van de ontvangen data beslissingen zal nemen zoals het versturen van emails, het uitrekenen van gemiddelde waardes en het algemene verwerken van de data. Als er heel veel handscannergegevens worden verzameld en als er op deze data verschillende algoritmes worden losgelaten dan zit men in het domein van de 'big data' analyse.

- **Device portal API**: de handscanners moeten hun gegevens op een gestandaardiseerde manier naar de cloud versturen. Om dit te verwezenlijken moet de cloud een device API aanbieden. Tegenwoordig wordt dit veelal geïmplementeerd via een JSON RESTful API. Het protocol moet voorzien zijn van authenticatie zodat er kan worden verzekerd dat de API-requests wel degelijk van geauthenticeerde handscanners komen.
- **Gebruikers presentatie laag**: uiteraard moeten de gegevens die het systeem verzamelt toegankelijk worden gemaakt voor de gebruikers. Dit wordt door middel van een HTML5 webapplicatie gerealiseerd welke achterliggend gebruik maakt van een user API. Het loskoppelen van de gebruikersinterface en de API zorgt ervoor dat er in de toekomst ook een app kan worden geschreven om de gegevens te presenteren.

5.4 De handscanner

5.4.1 Onderdelen

Het prototype van de handscanner die in dit document wordt beschreven is volledig custom ontworpen. Het toestel zal uit volgende componenten worden opgebouwd:

- **Cijferklavier**: een klein cijferklavier met knoppen 1 tot en met 5 zal worden geplaatst. Via deze knoppen zal een score tussen 1 en 5 kunnen worden toegekend.
- **Bedieningsknoppen**: naast het kunnen toekennen van een score moet de lector ook nog kunnen schakelen tussen de twee toestelfuncties, namelijk aanwezigheiden opnemen en een score toewijzen. Dit schakelen tussen de twee modes kan met één knop en een mode-LED worden verwezenlijkt. Door één keer de drukken op de mode-knop kan de gebruiker schakelen tussen de verschillende modi van het toestel, de LED geeft op zijn beurt aan welke gebruikersmodus is gekozen.

  De laatste knop die op het toestel moet zitten is een knop om het toestel verder in te kunnen stellen. Als deze knop voor minimum 2-seconden wordt ingedrukt zal het toestel in installatiemodus gaan. In deze speciale modus zal de scanner zijn eigen WiFi netwerk uitzenden. De gebruiker kan dan met een laptop, smartphone of tablet via de webinterface dat het toestel zal aanbieden de verbindingsinstellingen van het toestel wijzigen.
- **Micro-USB aansluiting en batterij**: om de draagbaarheid en het gebruiksgemak van het toestel te verhogen zal het ook voorzien worden van een oplaadbare batterij welke via een micro-USB aansluiting kan worden opgeladen. De batterijtechniek met de hoogste energiedensiteit zijn de op lithium gebaseerde batterijen. Een lithium-polymeer

accu heeft de hoogste energiedichtheid maar is niet zo stabiel als een lithium-ion batterij. Omwille van de veiligheid is er dus gekozen om de handscanner te voeden met een lithium-ion batterij en dit in de vormfactor 18650. De 18650 is een single-cell lithium-ion batterij die in de meeste consument apparatuur wordt gebruikt. Zo zijn bijvoorbeeld ook laptopbatterijen intern uit zes of negen 18650 lithium cellen opgebouwd.

- **NFC-scanner**: NFC is een techniek waarbij de zender, de NFC-kaart in dit geval, zijn energie haalt uit de ontvanger. Om de ontwikkeling te versnellen zal gebruik worden gemaakt van een NFC-module die gebaseerd is op de PN5321.
- **DPT-Module**: Om al deze functionaliteiten op een simpele en goedkope manier te kunnen implementeren is voor de DPT-Module gekozen. Deze volledige geïntegreerde Linux WiFi oplossing op een module is een zogenaamde System-on-Module welke meteen op een printplaat kan worden gesoldeerd. Het is ook deze module die op het DPT-Board, welke doorheen de cursus werd gebruikt, is gemonteerd.

5.4.2 Het schema

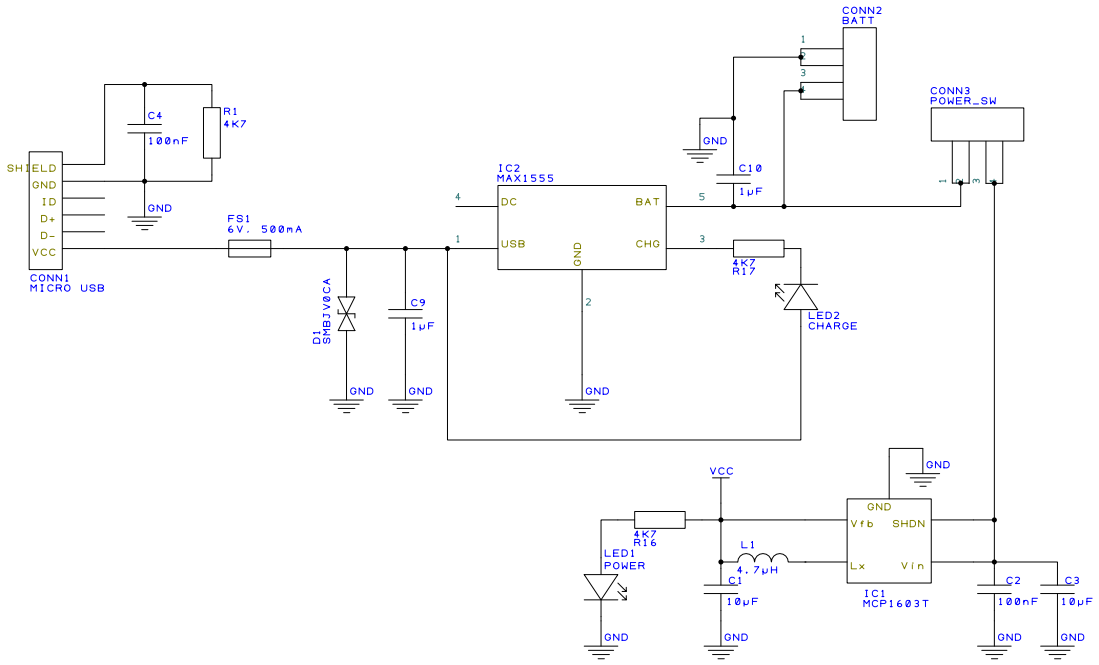
Het elektronische schema is de basis van elk IoT-toestel. Voor het integratieproject splitsen we het schema op in twee delen:

1. De voeding en laadelektronica bevat de lithium batterij lader en de spanningsregelaar voor de DPT-Module.
2. De bedieningselektronica welke de knoppen en LED indicators bevat.

Op figuur 32 wordt het eerste deel van het elektronisch schema weergegeven. Dit schema is opgebouwd rond twee verschillende IC's (Integrated Circuits):

- **IC1, MCP1603T**: het volledige toestel werkt intern op 3.3V. De batterij heeft echter een spanning van 4.2V als hij vol is en 3.4 volt als deze leeg is. Deze spanning loopt nog hoger op als de batterij wordt opgeladen. Het is duidelijk dat er een regeling nodig is die deze fluctuerende spanning omzet naar de gewenste spanning van 3.3V. De MCP1603T is een schakelende spanningsregelaar, zonder diep op de theorie in te gaan kan men stellen dat dit IC de spanning met een gemiddeld efficiëntie van 80
- **IC2, MAX1555**: het opladen van een lithium accu is een complexe zaak en het kan ook potentieel gevaarlijk zijn indien een lithium cel verkeerd wordt opgeladen. Daarom is er in het ontwerp voor gekozen om hiervoor een specifiek IC te gebruiken. Dit IC is ontworpen om een USB-voeding van 5V als ingang te nemen en een batterij te laden op de uitgang. Ook kan de chip gemakkelijk schakelen tussen het voeden via de USB-poort en het voeden via de batterij waardoor continu-operatie mogelijk wordt.

Verder zijn er aansluitingen voorzien om via pin-headers een batterij en een hoofdschakelaar aan te sluiten. LED2 duidt aan of het toestel aan het laden is of niet, de LED brandt zolang



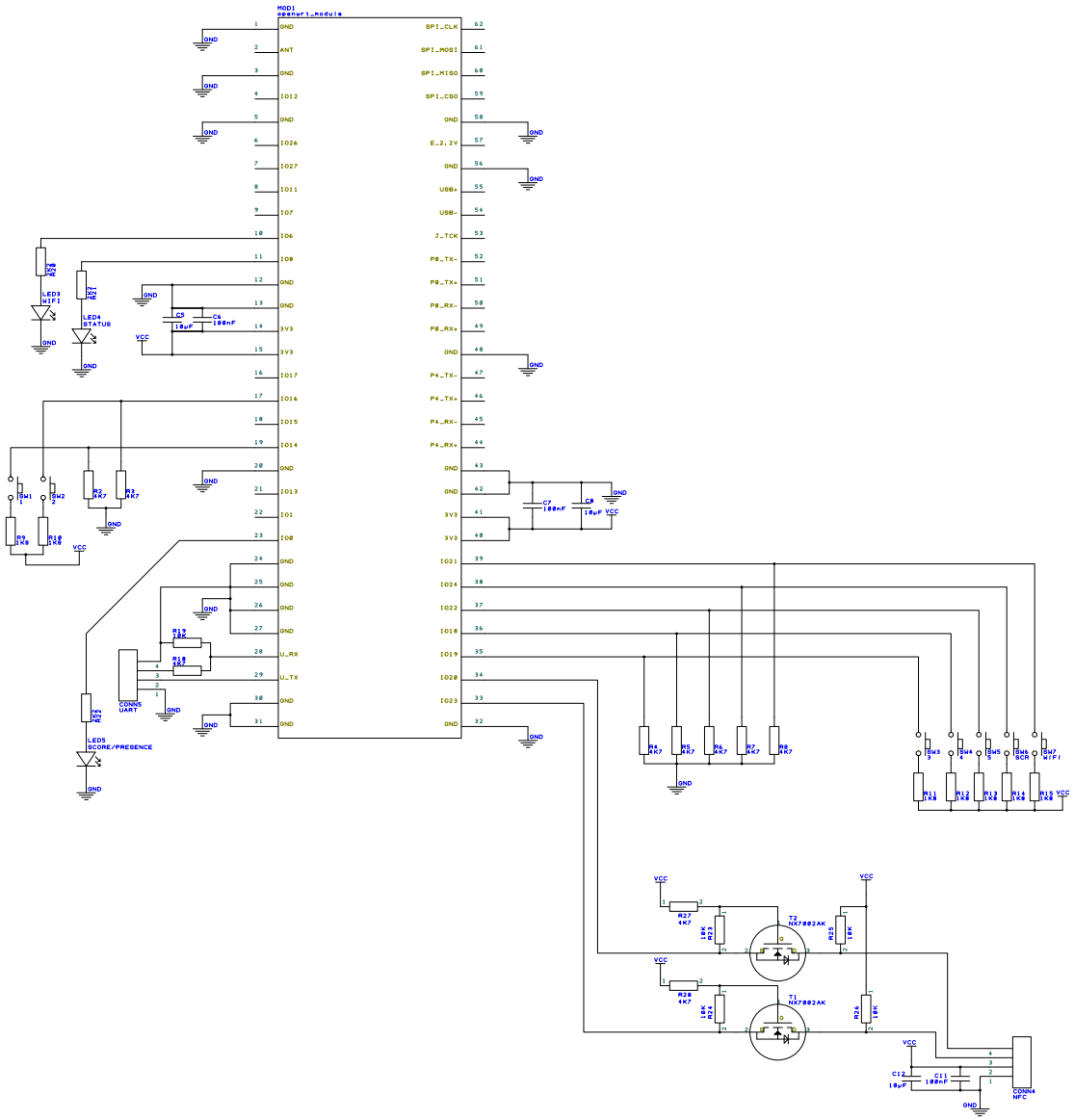
Figuur 32: De voeding en laadelektronica van de leervolgsysteemhandscanner

de batterij wordt bijgeladen en gaat uit als de batterij volledig opgeladen is.

LED1 geeft aan of de spanning is ingeschakeld en de 3.3V spanningsregelaar werkt. Het uitrusten van voedingslijnen met een LED indicator is een veelvoorkomende en handige praktijk. Zowel gebruikers als technici kunnen meteen belangrijke fouten in het toestel zoals een lege batterij of kapotte voeding herkennen zonder meetaparatuur nodig te hebben.
Figuur 33 is veel simpeler dan het voedingsschema. Het grootste deel van het schema bestaat uit de knoppen en aansturing van de status LED's. De I/O (Input/Output) poorten van de DPT-Module werken niet op de voedingsspanning van 3.3 volt maar wel op 2.5 volt, dit is een veel voorkomende spanning in de nieuwere computerchips. Dit is dan ook meteen de reden van de vele weerstanden in het schema, er wordt namelijk gebruik gemaakt van spanningsdelers om ervoor te zorgen dat de I/O-poorten niet een te hoge spanning te zien krijgen.

Het spanningsdeler principe is goedkoop en doeltreffend voor het inlezen van de knoppen en het koppelen van de seriele poort (UART). De NFC-Module wordt echter aangestuurd via het I^2C protocol. Dit protocol werkt met zogenaamde 'open-drain' uitgangen, een term welke afgeleid is uit de transistortechnologie en veel voorkomt in digitale elektronica. Een open drain uitgang heeft 2 toestanden:

- **Hard naar ground**: in deze toestand is de open drain uitgang rechtstreeks gekoppeld met de 'ground' of 0V rail.



Figuur 33: De bedieningselektronica van de leervolgsysteemhandscanner

- **Zwevend**: in deze toestand is de open drain uitgang met niets verbonden. Dit wil zeggen dat de stroomkring niet gesloten is en de rail om het even welke spanning kan aannemen, de kleinste storing in de omgeving zal zichtbaar zijn op deze uitgang. Om het spanningsniveau in de zwevende toegang te definiëren moet er dus steeds met een externe pull-up weerstand worden gewerkt.

Verder werkt I^2C met een bidirectionele datalijn, SDA (Serial DAta) en een seriele klok SCL (Serial CLock). Het is de combinatie van open drain uitgangen met pull-up weerstanden en

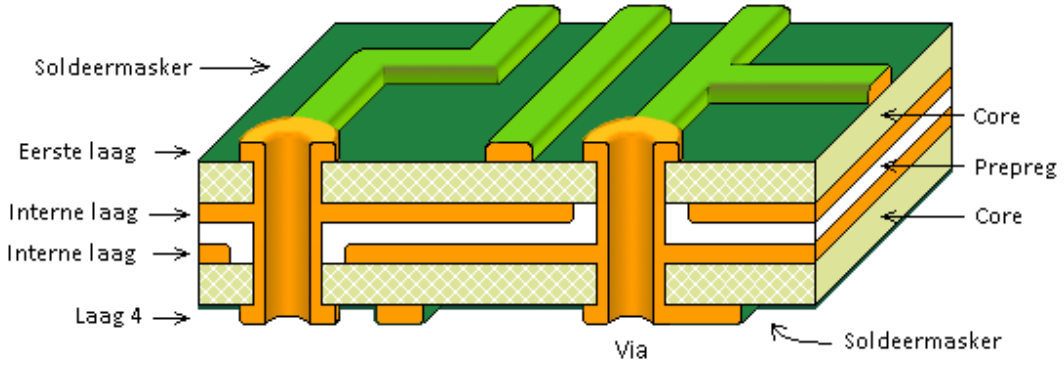
het bidirectionele karakter van deze SDA lijn dat ervoor zorgt dat de level-translation niet met een simpele spanningsdeler kan worden uitgevoerd. Een spaningsdeler werkt namelijk maar in één richting en niet in twee.

Om dit probleem op te lossen wordt er in het schema gebruik gemaakt van een standaard mosFET level-shifter. Zonder op de details in te gaan zorgt de mosfet dat een 2.5V spanning aan de source ervoor zorgt dat er 3.3V op de drain komt de staan en omgekeerd. Hierdoor kan de DPT-Module die op 2.5V werkt perfect via I^2 werken met een 3.3V NFC module.

5.4.3 Printontwerp

Nadat het elektrisch schema volledig is getekend en gecontroleerd moet er een ontwerper op bais van dit schema een printplaatlayout maken. Dit gebeurt met speciale software waarin het schema wordt opgeladen. De software weet van elk elektronisch component dat is opgenomen in het schema de 'footprint' of voetafdruk.

Printplaten bestaan uit een laag hard materiaal met één of meerdere lagen koper er op of er tussen. Vroeger werden printplaten in goedkopere toestellen vaak gemaakt van een soort samengeperst hard karton. De duurdere printplaten werden uit een glasvezelverbinding, FR4 genaamd, opgebouwd. Tegenwoordig kan men FR4 echter zo goedkoop fabriceren wat ertoe geleid heeft dat ongeveer elke moderne printplaat uit dat materiaal gemaakt is.



Figuur 34: Een voorbeel van een 4-laags PCB stackb

Een simpele printplaat bevat slechts één koperlaag waarmee alle verbindingen worden gevormd. Het aantal verbindingen dat op een printplaat aanwezig is, is de laatste jaren echter zeer sterk gestegen. Om dit op te lossen is men meerlaags-printplaten beginnen te ontwerpen. Bij dergelijk type printplaten zitten er op zowel de bovenzijde als onderzijde een zichtbare koperlaag en het is ook mogelijk dat er interne lagen worden toegevoegd.

Het komt vaak voor dat verbindingen van de ene koperlaag naar de andere koperlaag moeten lopen. Dit wordt door middel van via's zoals op figuur 34 bewerkstelligd. Een via wordt

gemaakt door een gat te boren door de printplaat en de binnenkant van dat gat te bekleden met koper. De meeste simpele consumenten-elektronica bevat heden ten dage een 4-lagen printplaat, als we echter een computermoederbord van de nieuwste generatie zouden analyseren dan zal men al snel aan 12 of meer koperlagen raken.

Op de meeste commerciële printplaten zal men op de koperlagen ook een soldeermasker aanbrengen. Het soldeermasker geeft de printplaat zijn karakteristieke groene kleur, maar dit kan desgewenst ook in andere kleuren worden aangebracht.. Enkel op de soldeerpads wordt er geen soldeermasker aangebracht, deze blijven rauw koper. Waar wel masker is aangebracht zal het solder niet blijven kleven. Het soldeermasker zorgt er dus voor dat het soldeer niet te ver uitloopt en dat soldeer van nabijgelegen pads niet aan elkaar plakt.

Als laatste laag kan nog een silkscreen worden aangebracht. Het silkscreen is niet meer dan een laag geprinte inkt waarmee de plaatsen en de namen van de componenten worden aangeduid. Vaak worden ook de logo's van de fabrikant en/of eventuele keurmerken meteen op het silkscreen geplaatst.

Er zijn heel wat soorten verpakkingen van elektronische componenten, omdat er echter duizenden verschillende soorten elektronische componenten bestaan heeft men heel wat standaardverpakkingen ontworpen in de loop der jaren die door meerder fabrikanten worden gebruikt. Het zijn enkel de wat specialere componenten die non-standaard voetafdrukken gebruiken, maar goede PCB ontwerp software laat de printplaatdesigner toe deze footprints zelf te specifieren zodat die ook kunnen worden gebruikt in het printplaatdesign. Op basis van hun footprint kunnen elektronische componenten in twee verschillende groepen worden ingedeeld:

- **Through-hole componenten**: alle componeten die tot deze groep behoren hebben verpakkigen met pootjes. Dit zijn de metalen pinnetjes aan de onderzijde van het component in kwestie. Het through hole component krijgt zijn naam van de manier waarop het gemonteeerd is op de printplaat. Voor elk pinnetje wordt er in de printplaat een gat geboord, dat gat wordt net zoals een via langs binnen bekleed met koper.

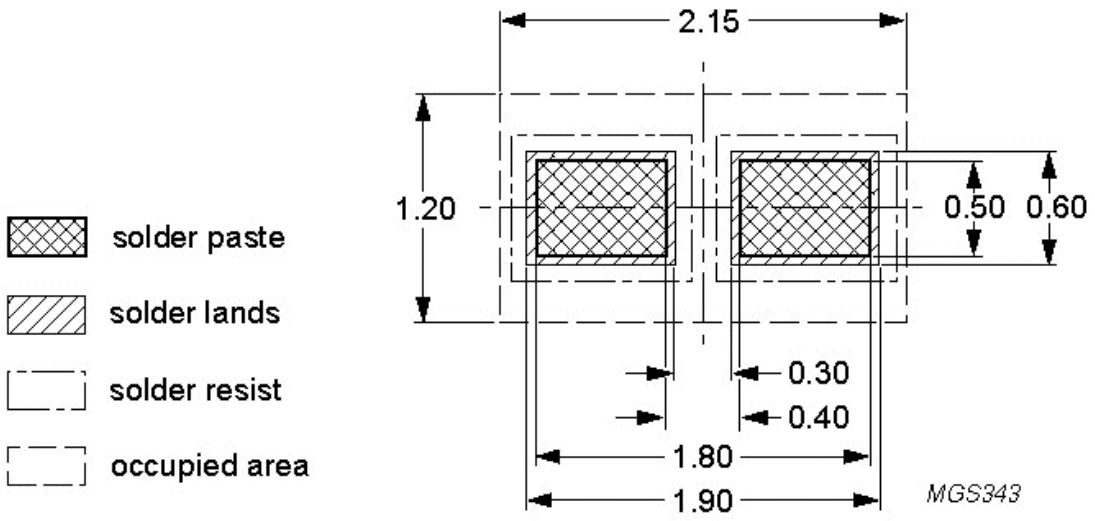
  Hoewel dit type component nog steeds vaak wordt gebruikt worden ze in nieuwere producten vaak vervangen door de SMD technologie.

- **SMD of Surface Mounted Devices**:

  SMD voetafdrukken zorgen ervoor dat er niet langer gaten moeten worden geboord om het component te bevestigen op de printplaat. Figuur 35 is een voorbeeld van een footprint waarbij geen boordwerk nodig is, het is een simpele weerstand of condenator met twee aansluitingen die op de zijkanten.

  Deze manier van werken geniet momenteel de voorkeur omwille van verschillende redenen:

  - Goedkoper dan through-hole componenten wegens de kleinere afmetingen.



Figuur 35: De PCB voetafdruk van een SMD weerstand

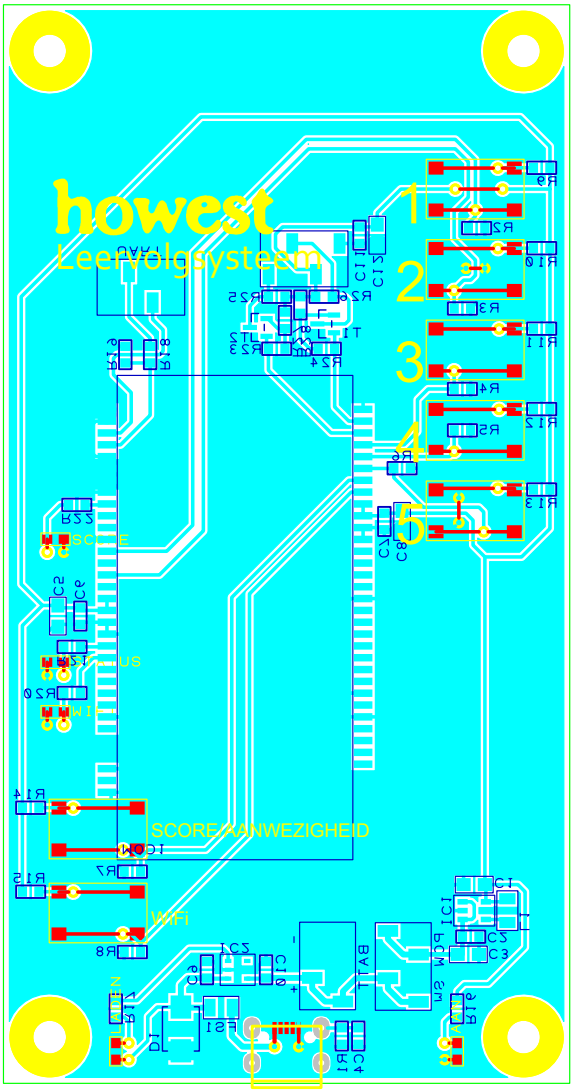
- Wegens de kleine afmeten is het goedkoper om ze te transporteren
- SMD componenten kunnen makkelijk in een oven wordt gesoldeerd.
- Componenten kunnen door middel van een pick-and-place robot worden geïnstalleerd op de printplaat.

Voor de leervolgscanner zal de printplaat 2 functionaliteiten hebben. Enerzijds wordt het de frontplaat van de witte behuizing en anderzijds zal het alle elektronische componenten met elkaar verbinden. Omdat de printplaat ook als afdekplaat wordt gebruikt zal een wit soldeermasker worden gebruikt en worden de meeste componenten, behalve de bedieningscomponenten, langs de onderkant van de print gemonteerd. Door deze constructie wordt een egale witte bovenkant bekomen wat de gebruiksvriendelijkheid van het toestel ten goede komt.

Het uiteindelijke printplaat ontwerp voor de handscanner wordt weergegeven op figuur 36. De kleuraanduidingen op de figuur hebben volgende betekenis:

- Geel: de silkscreenlaag aan de bovenzijde van de print.
- Donkerblauw: de silkscreenlaag aan de onderzijde van de print.
- Rood: het koper aan de bovenzijde van de print.
- Appelblauwzeegroen: het koper aan de onderzijde van de print.

Het soldeermasker wordt nooit weergegeven in de designsoftware omdat dit wordt berekend op basis van de footprint van de componenten.



Figuur 36: Het printplaatontwerp voor van de leervolgsysteemhandscanner

5.4.4 Het afgewerkt product

Als behuizing voor het prototype product is gekozen voor een standaard witte ABS behuizing. Normaalgezien zit er op deze behuizing ook een wit ABS deksel, maar dit wordt dus vervangen door de printplaat die precies op maat gemaakt is. Op deze manier wordt het een mooi effen productbehuizing.

De NFC scanner wordt onderaan aan de bovenkant van de behuizing gemonteerd waardoor

de studentenkaarten zowel onderaan als aan de bovenste zijkant van de behuizing kunnen worden gescanned. Naast de NFC-module wordt de behuizing ook voorzien van een aan/uit-knop aan de onderste zijkant, deze zit op een gemakkelijk te bereiken plaats waardoor er weinig gevaar is dat deze per ongeluk wordt ingedrukt waardoor het apparaat uitvalt.

Het laatste losse component dat ook in de behuizing wordt geplaatst is de 18650 lithium-ion cel. Deze wordt in een batterijhouder geplaatst die met warme lijm permanent wordt bevestigd. De cel kan dus makkelijk worden vervangen door de bovenkant van de behuizing te verwijderen, er moet niet worden gesoldeerd om de batterij te vernieuwen.

Figuur 37: De handscanner van het leervolgsysteem

Het uiteindelijke resultaat van de handscanner is te zien op figuur 37. Op de foto is de micro-USB laadkabel geplaatst en wordt de batterij opgeladen aangezien de laad LED-indicator oplicht.

5.4.5 De firmware

Minstens even belangrijk als de hardware is de firmware van een toestel. De ontwikkelingskosten van de firmware zijn vaak even duur, of zelfs duurder dan de kosten om de hardware te ontwikkelen en dus zeker niet te onderschatten. Voor zeer kleine en educatieve projecten zoals deze welke zijn voorgesteld in de bovenstaande cursus is Javascript een goede en ge-

makkelijke taal. Vanaf dat er echter meer functies zijn vereist of als men het onderste uit de kan wil halen qua performance is de taal bij uitstek in de embedded wereld nog steed ANSI C.

De taal C is een gecompileerde systeemprogrammeertaal welke in de jaren 70 is uitgevonden in het bekende Bell Labs met de bedoeling om op een portable manier een OS te ontwikkelen. Voor dat C bestond werd het OS van een computer namelijk in veel gevallen in CPU specifieke assembler taal geschreven waardoor het besturingssysteem volledig opnieuw moest worden ontwikkeld van zodra men van processorarchitectuur veranderde.

De Javascript interpreter die wordt gebruikt in deze cursus is op zijn beurt in C geïmplementeerd. Het is gecompileerde C-code die de emacsscript specificatie implementeert en de hardware-specifieke functies uitvoert. Aangezien de JS-engine dus gewoon een extra laag is tussen de hardware en de C-code wordt een commercieel project meteen in C ontwikkeld.

Deze keuze werd ook gemaakt voor de handscanner van het leervolgsysteem. Door de vele C-libraries die met de DPT-Module worden geleverd is het ontwikkelen van de firmware voor de handscanner relatief eenvoudig. Het enige wat nog moest worden voorzien om de firmware af te werken waren:

- De driver om met de I^2C NFC chip te communiceren.
- Het implementeren van de cloud-service API.

Om een driver te ontwikkelen voor een chip volstaat het de datasheet van de chip naukeurig door te lezen. In de datasheet van een digitale computerchip staat volledig beschreven welke signalen en commando's een chip verwacht en ook hoe de chip zal reageren op deze commando's. In de C-code moet dit gedrag dan worden geïmplementeerd.

Het consumeren van de cloud-service API wordt in C-code in vele gevallen gedaan met behulp van de cURL library. Ook voor het bouwen van de firmware voor de handscanner is hiervan gebruik gemaakt.

5.5 Het cloudsysteem

5.5.1 Het gekozen platform

Het ontwikkelen van een Internet-of-Things cloud is in wezen niet anders dan het ontwikkelen van een dynamische website. Een DBMS (DataBase Management System) wordt gebruikt om alle gegevens gestructureerd in op te slaan en er is een presentatielaag om deze gegevens weer te geven en te manipuleren. Het enige bijkomende is dat er ook een device-API moet worden ontwikkeld.

De ontwikkelplatforms waaruit men kan kiezen zijn, wegens deze grote overeenkomsten, dan ook nagenoeg gelijk aan deze waaruit men kan kiezen voor het ontwikkelen van dynamische websites. Zo zijn er traditioneel PHP, ASP.NET en Java. Toch zal men vaak zien

dat IoT-services geen van deze drie traditionele platforms gebruiken. Deels vanwege de lage performantie en deels vanwege het minder vernieuwende eco-systeem.

Nieuwere platformen zoals GoLang en Node.js worden vaker gebruikt om IoT-services op te zetten. Het is als bedrijf echter ook vaak de keuze van de klant welk platform er zal worden gebruikt. Het is namelijk zo dat naast de technische motieven om voor een bepaald platform te kiezen er ook andere zijn zoals:

- De licentiekosten die gepaard gaan met een bepaald platform. Zo zal een Windows-only platform zoals ASP.NET een hogere licentiekost hebben als een platform die ook op open en graties besturingssystemen zoals Linux draaien.
- Het ontwikkelaarsteam is een zeer belangrijke factor in de keuze van het platform. Een ontwikkelteam zal meeste één bepaald platform veel beter kennen dan andere platformen. Het is zeer belangrijk hiermee rekening te houden omdat dit de ontwikkelingstijd van een applicatie sterk kan beïnvloeden.
- De onderhoudbaarheid van een platform en dus het aantal te vinden developers die het platform kennen is ook zeer belangrijk. Een platform waarvoor veel minder ontwikkelaars te vinden zijn zal meestal tot een hogere onderhoudskost leiden.

Op basis van bovenstaande criteria is voor de ontwikkeling van het leervolgsysteem gekozen voor Node.JS als platform. Dit is een modern platform die matuur genoeg is om grotere systemen mee te implementeren. Er zijn zeer veel softwarebibliotheken beschikbaar via het NPM pakketmanagement systeem en ook zijn er legio ontwikkelaars voor dit platform beschikbaar. Dit wil zeggen dat het systeem in de toekomst makkelijk uitbreidbaar zal zijn en de onderhoudskost relatief laag.

5.5.2 Structuur

De structuur van het cloudsysteem is onder te verdelen in 4 verschillende onderdelen:

- **De device API**: deze API wordt uitsluitend geconsumeerd door de handscanners en voorziet alle mogelijkheden om de aanwezigheden en scores van studenten te ontvangen alsook om de status van de handscanners uit te lezen.
- **De presentation API**: in het leervolgsysteem is de user interface voor 100
- **De user interface**: de interface is gebouwd in het moderne HTML5 en consumeert de presentation API in de vorm van een webapp.
- **De database laag**: de database laag vormt een abstractie naar de database voor de rest van de applicatie. Dit is de enige laag waar SQL code te vinden is.

5.5.3 De device API

Deze API hoeft slechts een paar specifieke functies te vervullen:

- Het ontvangen van heartbeats van handscanners.
- Het ontvangen van aanwezigheden van handscanners.
- Het ontvangen van score's van handscanners.

Al deze functies moeten van authenticatie voorzien zijn. Dit wordt verwezenlijkt door voor elke handscanner een geheime sleutel aan te maken die wordt opgeslaan in de database laag van de cloud service. Elke request van een handscanner moet vooerzien zijn van de juiste key vooraleer de request wordt aanvaard.

Het is zeer belangrijk dat alle communicatie met de device API via een versleutelde TLS verbinding met een geldig SSL certificaat verloopt omdat het authenticatie schema anders niet langer geldig is. Het zou anders namelijk mogelijk zijn om een zogenaamde 'Man In The Middle' of MITM aanval uit te voeren op het systeem.

Het is namelijk triviaal om zelf een hotspot uit te zenden waarmee de handscanner zou kunnen verbinden. Als dat gebeurt kan de aanvaller alle verkeer tussen de handscanner en de cloud API uitlezen. Moest dit verkeer niet over een TLS verbinding lopen zou de aanvaller meteen de geheime sleutel van het toestel kunnen uitlezen. Vanaf dat de aanvaller deze sleutel heeft is het volledige systeem gehackt, want dan kan de aanvaller de volledige device API raadplegen.

Ook al is het met een TLS verbinding normaalgezien onmogelijk om een dergelijke aanval uit te voeren, dan nog is het aan te raden enkele voorzorgsmaatregelen te nemen voor het geval dit toch gebeurt. Een eerste zeer belangrijke maatregel is om de API niet meer mogelijkheden te geven dan strikt noodzakelijk. Zo is het voor de device API niet nodig dat er gegevens kunnen opgevraagd worden uit het systeem, het is dan ook aan te raden deze functionaliteit niet alsnog te implementeren. Een andere maatregel zou kunnen zijn om het aantal request per minuut te limiteren. Als er een beveiligingslek is kan men dan tenminste niet in zeer korte tijd de voledige database volschrijven, ...

5.5.4 De presentation API

De presentatie API is net zoals de device API een JSON REST API maar deze biedt alle functies aan die nodig zijn om de volledige UI van data te voorzien. In deze context kan UI zowel als een HTML5 webapplicatie als een native Android/iOS/...app zijn en dit is ook net de reden waarom er bij het ontwerp voor dergelijke opzet is gekozen. Klassiek gezien hangt de UI veel directer gekoppeld aan de data omdat er met zogenaamde templating engines wordt gewerkt.

Een template engine zoals te vinden is in php, ASP.NET maar ook in nieuwere pakketen zoals Node.js genereert een HTML pagina met reeds ingevulde gegevens. Als deze gegevens enkel door mensen moeten worden gelezen is dat een goede en snelle oplossing. Tegenwoordig moeten echter niet alleen mensen maar ook machines toegang hebben tot de databank en dan is HTML geen goede voorstelling meer. Formaten zoals JSON en XML zijn veel makkelijker te lezen voor computers en dan vooral het JSON formaat zorgt voor een zeer compacte notatie.

Net zoals de device API moet de presentation API van goede beveiliging worden voorzien, zeker omdat deze toegang geeft tot de gegevens van alle studenten. Er zijn zeer veel verschillende technieken waarmee een dergelijke API kan worden beveiligd. Bij het kiezen van een correcte oplossing is het hierbij belangrijk tussen volgende twee zaken onderscheid te maken:

- **Authenticatie**: het authenticeren is het controleren of een entiteit, een machiene of persoon, echt is wie hij bewijst. De meest bekende vorm van authenticeren is het ingeven van een gebruikersnaam en wachtwoord. Er is normaalgezien maar één persoon die deze goede combinatie van beide kan kennen en dus is bewezen dat hij is wie hij beweert te zijn.
- **Authorisatie**: naast authenticeren moet het systeem, eenmaal de entiteit gekend is, weten tot welke gegevens en acties een entitiiteit toegang heeft. Vooraleer een entitiiteit een actie kan uitvoeren moet deze eerst door het authorisatiesysteem worden goedgekeurd.

Voor dit integratieproject is er gekozen om de authenticatie te laten gebeuren met een simpele wachtwoord/gebruikersnaam combinatie. Als de gebruiker inlogt met de goede combinatie wordt er een cookie met een zogenaamd 'session token' meegestuurd met het antwoord. De client moet bij elke volgende request de cookie met ditzelfde session token terugsturen.

Het token in de cookie heeft 2 verschillende functies. De eerste functie is uiteraard om de gebruiker bij elke request te authenticeren. De tweede functie van het token is het bijhouden van een zogenaamde 'session'. Een session is een techniek die wordt gebruikt om 'state' te introduceren in HTML applicaties door gegevens te koppelen aan het token die met elke request wordt meegestuurd in de cookie. De session bevat in deze cloud applicatie alle gegevens over de ingelogde entiteit zodat de requests kunnen worden geauthoriseerd.

Een ontwikkelaar heeft heden ten dage een grote keuze aan authorisatiesystemen. De meest simpele kunnen met een paar `if` en `else` structuren worden geïmplementeerd, maar in de praktijk zijn er twee volwassen systemen die vaak worden gebruikt:

- **Access control list** of **ACL**: in deze strategie wordt een lijst van permissies gekoppeld aan een object. Elke gebruiker en/of machine in het systeem heeft dus een grote lijst van permissies of toegangsrechten. Dit systeem laat een heel fijne controle en instelling toe. Het grootste nadeel aan dit systeem is dat de beheerder voor elke gebruiker een grote lijst aan toegangsrechten moet beheren, in grotere systemen leidt dit al snel tot fouten.

- **Role Based Access Control** of **RBAC**: dit systeem probeert de grote tekortkoming van het ACL-systeem, namelijk het niet gegroepeerde management, op te lossen. In een RBAC systeem moet men volgende zaken onderscheiden:
  - **Permissies**: een permissie heeft in een RBAC context exact dezelfde betekenis als in een ACL systeem. Het is een bepaald recht zoals het mogen opvragen van een lijst van gebruikers. In tegenstelling tot een ACL wordt een permissie in een RBAC systeem niet aan een entiteit maar aan een bepaalde rol gekoppeld.
  - **Rollen**: een rol kan men het best vergelijken met een functie, bijvoorbeeld boekhouder, verkoper, lector, student. Permissies worden aan een rol en dus aan een bepaalde functie toegewezen.

  In een RBAC systeem worden aan een entiteit één of meerdere rollen toegekend. Als er bijvoorbeeld een rol 'boekhouder' in het systeem aanwezig is en er zijn 20 boekhouders in het bedrijf dan zullen deze alle 20 exact dezelfde permissies hebben. De systeembeheerder moet slechts de rechten van de rol 'boekhouder' aanpassen om de permissies van alle boekhouders te wijzigen.

Wegens de voordelen die een RBAC systeem biedt en de vele gebruikers die het leervolgsysteem zal hebben is ook in dit systeem voor RBAC communicatie gekozen. Hierbij zullen dan zeker 3 rollen worden toegevoegd zijnde 'lector', 'student' en 'systeembeheerder', ook al bevat het systeem dan 10 000 gebruikers, toch zullen de rechten van deze gebruikers met slechts 3 lijsten van permissies te beheren zijn.

5.5.5 De user interface

Voor dit integratieproject zal de volledig gebruikersinterface in HTML5 worden geschreven. De auteur is ook overtuigd dat deze technologie de native Android/iOS applicaties overbodig zal maken. Steeds meer functies van deze native applicaties worden in de moderne browsers beschikbaar gesteld. Ik denk dan aan de camera, microfoon, GPS, lokale opslag en op Android zijn zelfs native notifications mogelijk.

Het belangrijkste sleutelwoord bij de user-interface is 'responsive', dit wil zeggen dat dezelfde HTML pagina zich kan aanpassen aan gelijk welke schermgrootte. De website kan dus zonder herladen worden weergegeven op zowel een 24 inch 4K desktopscherm als op een low-end 4 inch smartphone scherm met een resolutie van 720×1280 pixels.

Om snel te schakelen wordt er voor het integratieproject gebruik gemaakt van het Bootstrap systeem om de responsive interface te bouwen. Bootstrap is een framework van javascript en CSS dat zeer veel componenten biedt waarmee een website kan worden opgebouwd. Het enorme voordeel is dat het responsive worden van de website door het framework wordt verzorgd, hierdoor kan men als developer focussen op functionaliteit in plaats van uiterlijk.

5.5.6 De database laag

De database laag bestaat uit het DBMS (DataBase Management System) en een DAO (Database Access Object). Het werken met een DAO is een programmeerstijl die meer controle overlaat aan de programmeur dan een ORM (Object Relational Mapping) systeem. Deze grotere vrijheid betekent iets meer en verbose programmeerwerk maar meestal ook een iets betere performance als het gaat over grote queries.

De keuze tussen een ORM is vaak een vrij persoonlijke keuze en gestoeld op de ervaring van het ontwikkelaarsteam. Beide systemen bieden out-of-the-box alle functionaliteit die nodig is om een complexe database-gestuurde cloud-toepassing te ontwerpen.

Er zijn zeer veel DBMS systemen zoals `MySQL`, `MSSQL`, `PostgreSQL` , `SQLite`, ... met elk een eigen SQL dialect. Het is zeer belangrijk dat de database laag de enige laag is die weet met welk DBMS er wordt gewerkt, nergens anders in de cloud applicatie mag er iets afhankelijk zijn van het DMBS systeem. Dit is nodig voor de goede onderhoudbaarheid van het systeem, als men ooit van DBMS moet wijzigen is er dan slechts één plaats in de code waar aanpassingen moeten worden doorgevoerd.

5.6 Ingebruikname

Omdat dit het doel van dit document voorbijgaat en de source te groot zou worden is er geen code bijgevoegd. Het is belangrijker voor de student om te begrijpen waarom bepaalde ontwerpkeuzes gemaakt zijn. Om dit uitgebreide integratieproject af te sluiten worden een paar scenario's voorgesteld die de werking van het systeem voor de gebruikers aantonen:

5.6.1 Opnemen van aanwezigheden en beoordelen van studenten

Het opnemen van aanwezigheiden wordt voor lectoren zeer eenvoudig met het leervolgsysteem en kan in volgende stappen worden voltooid:

1. De lector logt in het systeem in en moet het vak waarvoor hij/zij de aanwezigheiden wil opnemen aanmaken of selecteren.
2. Op dat moment kan de handscanner worden ingeschakeld, deze zal automatisch verbinden met het WiFi netwerk en aanmelden op het systeem.
3. De lector ziet een lijst van aangemelde handscanners en selecteert de scanner die hij/zij zal gebruiken om de aanwezigheden op te nemen.
4. Als de score LED niet brandt staat de handscanner in aanwezigheidsmodus, de lector moet controleren of dit effectief zo is. Als de score LED wel brandt staat het toestel in de 'score' modus en moet men op de 'score/aanwezigheid' knop drukken om terug naar de 'aanwezigheid' modus te gaan.

5. Vanaf dat er een studentenkaart tegen de onderkant van het toestel wordt gehouden knippert de status LED één maal om aan te geven dat de kaart goed gelezen is. Op dat moment wordt de ID van de kaart naar de cloud-service gezonden die deze verwerkt en de student als aanwezig markeert. Studenten die niet aanwezig zijn zullen hun kaart niet laten scannen en zullen dus ook niet als aanwezig worden gemarkeerd.

6. Als de lector alle kaarten gescanned heeft moet hij via de webinterface de aanwezigheiden afsluiten. Het systeem weet dan dat er geen extra kaarten meer zullen worden gescand en de aanwezigheiden zijn dan finaal.

Eenzelfde werkwijze zal kunnen worden gebruikt voor het geven van een score aan de studenten in een bepaald vak. In dat geval zal men de handscanner wel in in 'score' modus moeten plaatsen door op de 'score/aanwezigheid' knop te drukken. De lector moet dan de kaart scannen, de status LED knippert één keer bij het correct lezen van de kaart. Meteen daarna moet de lector een keuze maken tussen score 1 tot en met 5 en de status LED zal 2 keer knipperen om aan te geven dat de score goed is doorgestuurd naar het platform.

5.6.2 Opvragen van rapporten

Naast het opnemen van gegevens moeten deze natuurlijk ook worden verwerkt tot nuttige rapporten en aan die rapporten kunnen acties zoals het verzenden van een email worden gekoppeld. Voorbeelden van rapporten die kunnen worden gemaakt zijn:

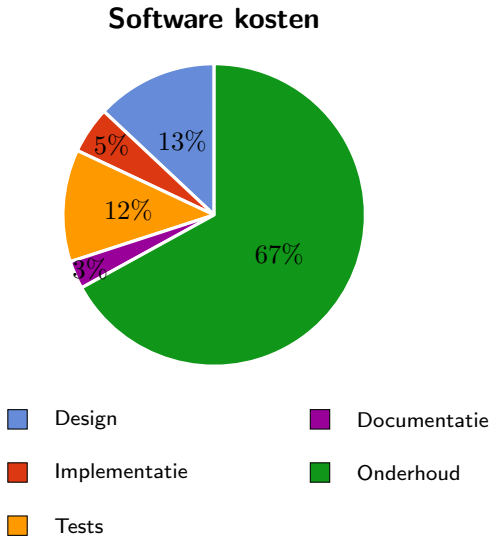
- Een overzicht van het aantal afwezigheden van een bepaalde student.
- De gemiddelde score van alle studenten voor een bepaald vak.
- Een overzicht van de scores voor elk vak van een specifieke student.
- ...

Naast het opvragen van rapporten via de gebruikersinterface is het ook de bedoeling dat het systeem op vaste tijdstippen of door bepaalde triggers emails stuurt naar o.a. studenten, lectoren en eventueel zelfs ouders. Dit zorgt ervoor dat men niet verplicht moet inloggen om bepaalde resultaten te verkrijgen.

5.6.3 Toekomstige uitbreidingen en onderhoud

Alle softwaresystemen, dus ook Internet-of-Things systemen, vergen continu onderhoud en uitbreidingen. Het is zelfs zo dat de onderhoudskosten van een softwarepakket in verhouding vele malen hoger liggen dan de ontwikkelingskosten. Op figuur 38 wordt een typische verdeling weergegeven van de kosten van een IoT-project. Het onderhoud van de software kan ingedeeld worden in twee onderdelen:

1. Het up-to-date houden van de toepassing.
2. Het toevoegen van nieuwe features.



Figuur 38: Kosten die voortvloeien uit het ontwikkelen en onderhouden van software

Het up-to-date houden van een IoT-systeem is uitermate belangrijk omdat een verouderd netwerkcomponent een zeer groot beveiligingsprobleem is. Bij veel IoT-toepassingen is het update mechanisme niet voldoende of zelfs helemaal niet aanwezig waardoor men bijvoorbeeld via een koffiezet het bedrijfsnetwerk zou kunnen binnendringen. Omwille van die reden is het belangrijk om ervoor te zorgen dat bijvoorbeeld de SSL-bibliotheek en certificatenlijst up-to-date is. Maar ook kernel-updates, serversoftware updates, . . . zijn allemaal noodzakelijk voor de goede en vooral veilige werking van het IoT-systeem.

Veelal zullen de gebruikers van een systeem veranderingen vragen en nieuwe features willen zien. Hoewel niet strikt noodzakelijk zorgt het toevoegen van deze features voor een groter gebruiksgemak en meer tevreden gebruikers van het systeem. Enkele mogelijke uitbreidingen van het leervolgsysteem zouden bijvoorbeeld de volgende kunnen zijn:

- Koppelen van het systeem met andere IT-systemen binnen de onderwijsinstelling.
- Het inzetten van de handscanners om ook toegangscontrole te voorzien op deuren.
- Andere types badges dan NFC-A ondersteunen.

Een goede documentatie van de softwarecode is voor dit onderhoud cruciaal. Het komt veel voor dat de ontwikkelaars die het onderhoud van een systeem uitvoeren niet dezelfde zijn als diegene die het systeem hebben gebouwd. Als de documentie van de systeembouwers niet voldoende is kan dit voor een enorme verhoging van de onderhoudskost leiden. Ook het toevoegen van nieuwe features is dan bijna niet meer mogelijk.

6 Kritische reflectie

Het ontwikkelen van een IoT-cursus is niet gemakkelijk en het is moeilijk om bepaalde zaken op een gemakkelijke manier toegangkelijk te maken. Het document zoals dit hier is voorgesteld is vooral geschikt om te gebruiken als basis voor een cursus in het hoger onderwijs zoals universiteiten en hogescholen.

Tijdens het maken van deze bachelorproef zijn de hands-on voorbeelden in de IoT-cursus van HoWest uitgetest. Zoals steeds met nieuwe technieken zijn hier een paar kinderziektes door ontdekt. Het grootste probleem was dat updates moeilijk op de DPT-Boards konden worden geïnstalleerd en dit had twee oorzaken:

- De DPT-Boards konden niet goed overweg met de WPA2-Enterprise verbindingen die op de hogeschool beschikbaar zijn.
- Het updatemechanisme werkte niet automatisch.

Aan beide problemen is tijdens het maken van de bachelorproef gewerkt en ook allebei de problemen zijn opgelost. Het DPT-Board kan nu met alle soorten draadloze netwerken en beveiligingstypes overweg. Ook het update mechanisme is zeer sterk verbeterd en de software op de DPT-Boards kan zichzelf nu volledig automatisch updaten.

Het integratieproject is vooral op hardware- en firmwarevlak volledig uitgewerkt met een zeer mooi resultaat. Het cloud-platform is volledig ontworpen en uitgetekend, maar de implementatie ervan is niet volledig afgewerkt. Alles is echter goed gedocumenteerd en de basisfunctionaliteit van een IoT-platform is dan ook volledig zichtbaar. Het platform is met andere woorden voorbereid om goed te worden onderhouden door het volgende ontwikkelteam.

7 Conclusies

Aan de hand van dit document kunnen hogescholen en middelbare scholen een IoT-cursus inrichten. De basis van digitale elektronica en het programmeren van IoT-toepassingen is slechts het topje van de ijsberg bij het beginnen aan een job in deze sector. Het integratieproject was een uitbreiding om te laten zien welk breed spectrum aan specialiteiten er nodig zijn om een volledige IoT-toepassing uit te bouwen zonder van prefab-systemen gebruik te maken. Er zijn softwaredevelopers, hardwaredevelopers, UI designers, infrastructuur installateurs, ... nodig om dit te verwezenlijken.

Deze evolutie naar een geconnecteerde en geautomatiseerde wereld zal op korte termijn een grote hoeveelheid aan technische uitdagende jobs creëren. Deze bachelorproef heeft als doel om onderwijsinstellingen te helpen cursussen in te richten die beter aansluiten bij deze nieuwe jobs. Het is belangrijk om deze cursussen zo in te richten dat jongeren geënthousiasmeerd worden door techniek.

Met het DPT-Board ontwikkelplatform kunnen studenten op korte termijn compleet werkende projecten bouwen zoals in de hands-on hoofdstukken van dit document. Deze korte leercurve zonder al te veel theorie zorgt ervoor dat studenten techniek niet als iets onbegrijpelijk en moeilijk gaan ervaren. Dit heeft de auteur van deze bachelor reeds uitgetest in samenwerking met enkele middelbare scholen in Vlaanderen. Zelfs studenten die nog niet hadden geprogrammeerd waren in staat om de robot een bepaald traject te laten afleggen.

Ook na het schrijven en uitwerken van deze bachelorproef zal de auteur verdergaan met het promoten van IoT-techniek in het onderwijs. Daarvoor zijn nu reeds samenwerkingsverbanden met HoWest vastgelegd. In samenwerking met de onderwijsrichting van de hogeschool zal DPTechnics nog meer hands-on voorbeelden uitwerken en zullen cursussen worden ontwikkeld die ook voor het middelbaar en zelfs lager onderwijs geschikt zijn.

Door het ontwikkelen van deze cursussen zullen leerkrachten uit de middelbare en lagere graad in staat zijn om ook zonder voorkennis techniek te integreren in hun lessenpakket. Het is enkel op die manier dat leerkrachten technologie als een evenwaardige pijler in het lessenpakket kunnen gaan opnemen. Dit is nodig omdat een onderwijzer zonder goed ontwikkeld ondersteunend materiaal niet de mogelijkheden heeft om een volwaardige en interessante cursus aan te bieden.

8 Overzicht van de bijlagen

8.1 Simon Says broncode

```
/**
 * Simon Says for the DPT-Board
 */

// translating to hardware pins. ex: [23, 22, 1, ...]
var output = [6, 7, 16, 8];
var input = [12, 13, 15, 19];

// length of current run
var currLength = 2;

// generated series
var serie = [];

// read series
var read = [];

function setLength(nr) {
        currLength = nr;
}

function isCorrect() {
        print("is " + serie + " === " + read);
        for (var i=0; i < serie.length; i++) {
                if (serie[i] !== read[i]) {
                        return false;
                }
        }
        return true;
}

function doWon() {
        setLength(currLength+1);
        print("-- won -- next level " + currLength);
        digitalWrite(output[0], 1);
        digitalWrite(output[1], 1);
        digitalWrite(output[2], 1);
        digitalWrite(output[3], 1);
        setTimeout(restart, 2000);
}

function doLost() {
        setLength(2);
        print("-- lost -- going back to level " + currLength);
```

```
        digitalWrite(output[0], 1);
        digitalWrite(output[1], 0);
        digitalWrite(output[2], 0);
        digitalWrite(output[3], 1);
        setTimeout(restart, 1500);
}

function restart() {
        digitalWrite(output[0], 0);
        digitalWrite(output[1], 0);
        digitalWrite(output[2], 0);
        digitalWrite(output[3], 0);
        setTimeout(playRound, 1500);
}

function nextClick(nr) {
        print("button:␣" + nr);
        read.push(nr);
        if (serie.length === read.length) {
                print("finished␣run␣of␣" + read.length);

                if (isCorrect()) {
                        doWon();
                } else {
                        doLost();
                }
        }
}

function randPin() {
        var x = Math.random();
        if (x < 0.25) return 0;
        if (x < 0.50) return 1;
        if (x < 0.75) return 2;
        if (x < 1.00) return 3;
}

// call function "button" with button-nr and down/up only once.
// default state = 1
var states = [1,1,1,1];

setInterval(function buttonReader() {
        for(var i=0; i < input.length; i++) {
        var curr = digitalRead(input[i]);
                if (curr !== states[i]) {
                        if ((curr === 0)) {
                                nextClick(i);
                        }
                        states[i] = curr;
```

```
                }
        }
}, 20);

function playRound() {
        serie = [];
        var nr = 0;

        function doOn() {
                digitalWrite(output[serie[nr]], 1);
                setTimeout(doOff, 600);
        }

        function doOff() {
                digitalWrite(output[serie[nr]], 0);
                nr += 1;
                if (nr < currLength) {
                        setTimeout(doOn, 333);
                } else {
                        read = [];
                }
        }

        // make random serie
        for (var i=0; i<currLength; i++) {
                serie.push(randPin());
        }
        print("to␣guess:␣" + serie);
        doOn();
}

setLength(2);
playRound();
```

8.2 LCD weergave broncode

```
/**
 * LCD Hands-on code
 */

// Initialize an I2C bus with IO23 as SDA and IO20 as SCL
var i2c = new I2C(23,20);
i2c.begin();

// The LCD screen has I2C slave addres 0x3F
i2c.setSlaveAddress(0x3F);
```

```
// Initialize an LCD with 2 rows and 16 characters
var lcd = new I2CLCD(i2c, 16,2);

// Write 'dptechnics.com' onto the LCD screen
lcd.begin(function(){
        lcd.setCursor(0,0, function() {
            lcd.print("dptechnics.com");
        });
    });
```

8.3 DPTechnics I2C LCD bibliotheek

```
/*
 * Copyright (c) 2015, Daan Pape
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met
    :
 *
 *     1. Redistributions of source code must retain the above copyright
 *        notice, this list of conditions and the following disclaimer.
 *
 *     2. Redistributions in binary form must reproduce the above copyright
 *        notice, this list of conditions and the following disclaimer in the
 *        documentation and/or other materials provided with the distribution
    .
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS
    "
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */

/**
 * The I2CLCD prototype class to control a standard PCF8574
 * I2C LCD.
 * @param {I2C} i2c the I2C object with the LCD address already set.
```

```
 * @param {int} cols the number of columns on the screen.
 * @param {int} rows the number of rows on the screen.
 * @returns {I2CLCD} the I2CLCD object.
 */
function I2CLCD(i2c, cols, rows) {
    /* Constants */
    var PCF8574_RS = 0x01;
    var PCF8574_EN = 0x04;
    var PCF8574_BL = 0x08;

    var CMD_CLEAR = 0x01;
    var CMD_OFF = 0x08;
    var CMD_ON_NO_CURSOR = 0x0C;
    var CMD_ON_BLINK_CURSOR = 0x0F;
    var CMD_ON_SOLID_CURSOR = 0x0E;

    var CMD_SHIFT_LEFT = 0x18;
    var CMD_SHIFT_RIGHT = 0x1C;

    var CMD_FIRST_LINE = 0x80;
    var CMD_SECOND_LINE = 0xC0;
    var CMD_THIRD_LINE = 0x94;
    var CMD_FOURTH_LINE = 0xD4;

    var CMD_LINE_OFFSETS = [CMD_FIRST_LINE, CMD_SECOND_LINE, CMD_THIRD_LINE,
    CMD_FOURTH_LINE];

    /* Public prototype variables */
    this.n_cols = cols;
    this.n_rows = rows;

    /* Private variables */
    var i2cdev = i2c;
    var lcdBacklight = PCF8574_BL;
    var lcdCursor = CMD_ON_NO_CURSOR;
    var lcdOn = true;

    /* The type of LCD commands */
    var CommandType = {DATA: 0, COMMAND: 1};

    /**
     * Write data to the LCD screen.
     * @param {uint8} data the data to write to the LCD.
     * @param {CommandType} type the type of data to write to the LCD.
     * @param {function} callback the callback function when writing is
    complete.
     */
    var _lcd_write = function (data, type, callback) {
        var low_bits = (data & 0x0f) << 4;
```

```
        var high_bits = data & 0xf0;

        var control = lcdBacklight;
        var delay = 0;
        if (type === CommandType.DATA) {
            control = control | PCF8574_RS;
            delay = 1;  /* Wait 1 ms for data writes */
        } else {
            delay = 5;  /* Wait 5 ms for command execution */
        }

        /* Send higher part of the byte */
        i2cdev.writeByte(high_bits | control);
        setTimeout(function () {
            i2cdev.writeByte(high_bits | control | PCF8574_EN);
            setTimeout(function () {
                i2cdev.writeByte(high_bits | control);
                setTimeout(function () {
                    /* Send lower part of the byte */
                    i2cdev.writeByte(low_bits | control);
                    setTimeout(function () {
                        i2cdev.writeByte(low_bits | control | PCF8574_EN);
                        setTimeout(function () {
                            i2cdev.writeByte(low_bits | control);
                            setTimeout(function () {
                                if(typeof(callback) === "function") {
                                    callback();
                                }
                            }, delay);
                        }, 1);
                    }, 1);
                }, delay);
            }, 1);
        }, 1);
    };

    /**
     * Initialize the LCD screen.
     * @param {function} callback the callback function when the LCD is
    initialized.
     */
    this.begin = function (callback) {
        /* Initialize the LCD display to use 4-bit mode */
        i2cdev.writeByte(0x30);
        i2cdev.writeByte(0x30 | PCF8574_EN);
        setTimeout(function () {
            i2cdev.writeByte(0x30);
            i2cdev.writeByte(0x30 | PCF8574_EN);
            setTimeout(function () {
```

```
                i2cdev.writeByte(0x30);
                i2cdev.writeByte(0x30 | PCF8574_EN);
                setTimeout(function () {
                    i2cdev.writeByte(0x20);
                    i2cdev.writeByte(0x20 | PCF8574_EN);
                    setTimeout(function () {
                        /* 4-bit 2-line */
                        _lcd_write(0x28, CommandType.COMMAND, function () {
                            /* Clear display and put cursor at home position
    */
                            _lcd_write(0x01, CommandType.COMMAND, function ()
     {
                                /* Turn on display and hide cursor */
                                _lcd_write(0x0C, CommandType.COMMAND,
    callback);
                            });
                        });
                    }, 5);
                }, 1);
            }, 1);
        }, 5);
    };

    /**
     * Clears the LCD screen and positions the cursor in the upper-left
    corner.
     * @param {function} callback the callback function when the operation is
     done.
     */
    this.clear = function (callback) {
        _lcd_write(CMD_CLEAR, CommandType.COMMAND, callback);
    };

    /**
     * Positions the cursor in the upper-left corner of the LCD.
     * @param {function} callback the callback function when the operation is
     done.
     */
    this.home = function (callback) {
        this.setCursor(0, 0, callback);
    };

    /**
     * Set the location of the cursor to a specific position.
     * @param {int} col the column at which to position the cursor with 0
    being the first column.
     * @param {int} row the row at which to position the cursor with 0 being
    the first row.
```

```
     * @param {function} callback the callback function when the operation is
     done.
     */
    this.setCursor = function (col, row, callback) {
        var r = row % this.n_rows;
        var c = col % this.n_cols;

        _lcd_write(CMD_LINE_OFFSETS[r] + c, CommandType.COMMAND, callback);
    };

    /**
     * Helper function to print text to the LCD.
     * @param {string} text the to print.
     * @param {int} pos the current position in the string.
     * @param {function} callback the callback function when the operation is
     done.
     */
    var _print_helper = function (text, pos, callback) {
        if (text.length > pos) {
            _lcd_write(text[pos].charCodeAt(0), CommandType.DATA, function ()
     {
                _print_helper(text, pos + 1, callback);
            });
        } else {
            /* All text is written */
            if(typeof(callback) === "function") {
                callback();
            }
        }
    };

    /**
     * Prints text to the LCD.
     * @param {string} text the text to print to the LCD.
     * @param {function} callback the callback function when the operation is
     done.
     */
    this.print = function (text, callback) {
        _print_helper(text, 0, callback);
    };

    /**
     * Turns on the LCD display, after it's been turned off with noDisplay().
     This will restore
     * the text (and cursor) that was on the display.
     * @param {function} callback the callback function when the operation is
     done.
     */
    this.display = function(callback) {
```

```
        lcdOn = true;
        _lcd_write(lcdCursor, CommandType.COMMAND, callback);
    };

    /**
     * Turns off the LCD display, without losing the text currently shown on
    it.
     * @param {function} callback the callback function when the operation is
     done.
     */
    this.noDisplay = function(callback) {
        lcdOn = false;
        _lcd_write(CMD_OFF, CommandType.COMMAND, callback);
    };

    /**
     * Display the LCD cursor at the position to which the next character
    will be written.
     * @param {function} callback the callback function when the operation is
     done.
     */
    this.cursor = function (callback) {
        lcdCursor = CMD_ON_SOLID_CURSOR;
        if(lcdOn) {
            _lcd_write(lcdCursor, CommandType.COMMAND, callback);
        }
    };

    /**
     * Hide the LCD cursor.
     * @param {function} callback the callback function when the operation is
     done.
     */
    this.noCursor = function(callback) {
        lcdCursor = CMD_ON_NO_CURSOR;
        if(lcdOn) {
            _lcd_write(lcdCursor, CommandType.COMMAND, callback);
        }
    };

    /**
     * Display the blinking LCD cursor. If used in combination with the
    cursor() function, the result will depend
     * on the particular display.
     * @param {function} callback the callback function when te operation is
    done.
     */
    this.blink = function(callback) {
        lcdCursor = CMD_ON_BLINK_CURSOR;
```

```
        if(lcdOn) {
            _lcd_write(lcdCursor, CommandType.COMMAND, callback);
        }
    };

    /**
     * Turn on the LCD backlight.
     * @param {function} callback the callback function when te operation is
    done.
     */
    this.backlight = function(callback) {
        lcdBacklight = PCF8574_BL;
        i2cdev.writeByte(lcdBacklight);
        if(typeof(callback) === "function") {
            callback();
        }
    };

    /**
     * Turn off the LCD backlight.
     * @param {function} callback the callback function when te operation is
    done.
     */
    this.noBacklight = function(callback) {
        lcdBacklight = 0;
        i2cdev.writeByte(lcdBacklight);
        if(typeof(callback) === "function") {
            callback();
        }
    };

    /**
     * Shift the screen contents to the right.
     * @param {function} callback the callback function when te operation is
    done.
     */
    this.shiftRight = function(callback) {
        _lcd_write(CMD_SHIFT_RIGHT, CommandType.COMMAND, callback);
    };

    /**
     * Shift the screen contents to the left.
     * @param {function} callback the callback function when te operation is
    done.
     */
    this.shiftLeft = function(callback) {
        _lcd_write(CMD_SHIFT_LEFT, CommandType.COMMAND, callback);
    };
};
```

8.4 Parking sensor broncode

```
/*
 * Parking sensor based on the DPT-Board
 */

// This flag is true when the parking sensor is active
var systemActive = true;

// The last measured distance
var distance = 10;

// UART receive data buffer
var data = [];

/**
 * This function converts 2 input bytes to
 * a distance in centimeters for the US-100
 * sensor response
 */
function UltrasonToDistance(input1, input2){
    return (input1 * 256 + input2)/10;
};

/**
 * This function checks if the sensor distance
 * is less than 10cm when the system is activated.
 * Only when the systme is active and the distance
 * is less than 10cm, the function returns true.
 */
function isToClose(){
    if(systemActive){
        uart.writeByte(0x55);
        return Math.floor(distance) < 10;
    } else {
        displayWrite("No measure");
        return false;
    }
};

/**
 * This function checks if the last measured
 * distance is to close and activates the
 * correct warnings.
```

```
 */
function warningCheck(){
        if(isToClose()){
                displayWrite("Attention");
                digitalWrite(6,1);
        } else {
                displayWrite("Inactive");
                digitalWrite(6,0);
        }
};

/**
 * This funtion toggles the system active state
 * when the button is pressed.
 */
function buttonPressed(){
    if(!digitalRead(13)){
        systemActive = !systemActive;
    }
};

/**
 * Write something on the LCD at position 0,0
 */
function displayWrite(input){
    lcd.begin(function(){
        lcd.setCursor(0,0, function() {
            lcd.print(input);
        });
    });
};

// Initialize the I2C bus with SDA = IO23 and SCL = IO20
var i2c = new I2C(23,20);
i2c.begin();

// Create an I2C LCD on address 0x3F
i2c.setSlaveAddress(0x3F);
var lcd = new I2CLCD(i2c, 16,2);

// Open the UART port with 9600 baud
var uart = new Serial();
uart.begin(9600, function(success){
   if(success) {
       print("UART␣connected");
   } else {
       print("no␣UART␣connection");
   }
});
```

```
// Register a UART receive handler
uart.onReceive(function(err, b){
        // Append a received byte to the data array
    data.push(b);
    if(data.length > 1){
        distance = UltrasonToDistance(data[0], data[1]);
        print("Distance:␣" + distance + "cm");
        data = [];
    }
});

// Turn of the warning LED
digitalWrite(6,0);

// Start all the program loops
setInterval(buttonPressed, 300);
setInterval(isToClose, 100);
setInterval(blink, 1000);
```

8.5 Broncode voor de object-avoiding robot

```
/*
 * Parking sensor based on the DPT-Board
 */

var leftSensor = 20;    // The IO port for the left sensor
var rightSensor = 23;   // The IO port for the right sensor

// Stop all motors
function stopDriving()
{
        digitalWrite(18,0);
    digitalWrite(21,0);
    digitalWrite(22,0);
    digitalWrite(24,0);
}

// Drive forward with the robot
function driveForward()
{
        digitalWrite(18,1);
    digitalWrite(21,0);
    digitalWrite(22,0);
    digitalWrite(24,1);
}
```

```
// Drive in reverse
function driveBackward()
{
        digitalWrite(18,0);
    digitalWrite(21,1);
    digitalWrite(22,1);
    digitalWrite(24,0);
}

// Turn the robot to the left
function driveLeft()
{
        driveBackward();
        setTimeout(function() {
                stopDriving();
                digitalWrite(18,1);
                digitalWrite(21,0);
                digitalWrite(22,0);
                digitalWrite(24,0);
        }, 50);
}

// Turn the robot to the right
function driveRight()
{
        driveBackward();
        setTimeout(function() {
                stopDriving();
                digitalWrite(18,0);
                digitalWrite(21,0);
                digitalWrite(22,0);
                digitalWrite(24,1);
        }, 50);
}

// Increase the turncount variable to make the robot more sensitive.
var turnCount = 0;

// This variable remembers the last turning direction
var turnDir = driveRight;

var drive = function(){
        // Stop the previous driving action
        stopDriving();

        // Detect obstacles
        var obstacleLeft = !digitalRead(leftSensor);
        var obstacleRight = !digitalRead(rightSensor);
```

```
        if(turnCount >= 5) {
                // We tried to long turning, go backward until no obstacle
    visible and turn
                if(obstacleLeft || obstacleRight) {
                        driveBackward();
                } else {
                        // Now take the turn
                        turnDir();
                        --turnCount;

                        if(turnCount <= 5) {
                                turnCount = 0;
                        }
                }

        } else {
                // Take appropriate action
                if(!(obstacleLeft || obstacleRight)) {
                        print("No␣obstacle␣detected")
                        driveForward();
                } else if(obstacleLeft) {
                                if(obstacleRight) {
                                        driveBackward();
                                } else {
                                        driveRight();
                                        ++turnCount;
                                        turnDir = driveRight;
                                }
                } else if (obstacleRight) {
                        if(obstacleLeft) {
                                driveBackward();
                        } else {
                                driveLeft();
                                ++turnCount;
                                turnDir = driveLeft;
                        }
                }
        }

        // Change this for speed adjust
        setTimeout(stopDriving, 50);
};

// Start the control loop
setInterval(drive, 100);
```