

Projecthandleiding

Domotica

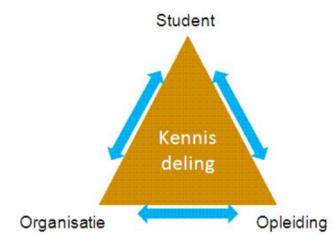
Tweedeperiodeproject
Informatica, NHL
Dec. 2016 - Jan. 2017



Samenstelling: S. Oosterhaven
Versie 6, 12/12/2016

1. Inleiding

“Eén van de uitgangspunten van de opleiding *Informatica* van de NHL is dat het onderwijs op een zodanige manier wordt aangeboden dat korte, intensieve en vakmatige benaderde cursussen, volgens een concentrisch model, worden opgevolgd door projecten waarin de geleerde kennis en vaardigheden geïntegreerd worden toegepast in een projectmatige omgeving, waarin studenten in groepen aan realistische opdrachten werken.”



Bovenstaande zin is *geen* typische openingszin voor een studentenhandleiding. Het is “onderwijstaal” zoals dat wordt gebruikt bij bijvoorbeeld de verantwoording van onderwijs. Het betekent zo iets als: leer eerst iets, oefen er dan even mee in kleine opdrachtjes en pas het vervolgens toe in een leuk project. En dat is precies wat we gaan doen!

In de huidige maatschappij, anno 2016-2017, neemt informatica, of met een meer internationale benaming “software engineering” een niet meer weg te denken plek in in onze maatschappij. Dat blijkt niet alleen uit de slimme apparaten die elkaar in snel tempo opvolgen en de ontelbare hoeveelheid websites en apps die we daarop moeten raadplegen om ons staande te houden in het digitale leven, maar ook uit de meer “verscholen software” die in steeds meer apparaten zit. Terwijl nut en noodzaak, commercie en ethiek hierbij vechten om voorrang, is de realiteit dat elke discussie op dat punt al op voorhand is ingehaald door de werkelijkheid. De kans is groot dat Sinterklaas binnen enkele jaren zijn pakjes per drone uit Spanje laat overvliegen. Of dat het romantische kinderbeeld aantast is niet zeker; dat het wellicht een oplossing biedt voor het Zwarte-Pieten-probleem is mooi meegenomen. En of die smart-tandenborstel met WiFi, waarmee Moeder vanuit de zelfsturende auto het poetsprofiel van haar kroost kan volgen, onder die drone hangt is evenmin zeker, maar dat die tandenborstel er gaat komen wel. Sterker nog: hij is er al, kijk maar naar de reclame.

De ingenieur-in-opleiding wordt enerzijds geconfronteerd met een dosis technologie (en ook technologisch optimisme) en anderzijds met de mogelijkheden die zelfstandig, m.b.v. internet (bijv. arduino.cc, stackoverflow.com en code-academy.com), te kunnen bestuderen en toepassen in nieuwe producten en diensten. Hoe moet het onderwijs hier op reageren? Wij hebben gemeend er goed aan te doen studenten niet alleen een vak te leren, maar ze ook m.b.v. van moderne middelen en modern onderwijs, op een even creatieve als kritische manier, problemen te leren oplossen waar de moderne maatschappij mee te maken krijgt. Dat doen we vooral in projecten. Het is dan ook de hoogste tijd, na het succes van de Racegame, een start te maken met project 2. Dat plaatsen we binnen de kaders van Internet Of Things en vooral Domotica.



Enkele definities

Domotica = huisautomatisering. Een samentrekking van *domus* (woning) en *informatica*.

Informatica = verwerking van informatie met de computer of (iets vollediger) de wetenschap die zich bezighoudt met de beheersing van complexiteit (o.a. op het vlak van informatie en gegevens, communicatie, softwarebouw, en technisch-wetenschappelijke rekenen).

Het combineren van deze twee definities geeft invulling aan het thema van dit project.

Thema

Het thema van het project is “Domotica”. We gaan we onze eigen woon- en leefomgeving automatiseren m.b.v. een embedded systeem en een daarmee samenwerkende app.

Projectdoel

Jullie (studenten) leren in relatief korte tijd en groepsgewijs een uitdagend, innovatief en “echt” systeem, met aanzienlijke complexiteit, te ontwerpen, te programmeren, te presenteren en te demonstreren. Het systeem, waar meerdere componenten, moderne programmeertechnieken en - talen bij betrokken zijn, is gericht op huisautomatisering, wellicht voor de doelgroep meer specifiek, op studentenslaapkamerautomatisering.

Doele, de student ...

- leert werken met en het programmeren van een embedded system
- leert werken met en het programmeren van smart devices voor diverse software platformen
- leert beide omgevingen te integreren in een domotica-toepassing en domotica-kader.
- werkt in groepen en beheert de gemeenschappelijke code met git/github
- werkt aan diverse individuele competenties ter voorbereiding op zijn beroep (zie bijlage 1)

Projectuitvoering

Binnen het domotica-project, dat start in week 6 van periode 2 en bijna 4 weken duurt, wordt door teams van 5 à 6 personen een domotica-systeem ontworpen en gebouwd. Dat systeem bestaat uit twee hoofdcomponenten:

1. Een embedded applicatie voor de Arduino Uno, uitgerust met een ethernetshield.
Meer informatie is te vinden in de studiewijzer behorend bij dit project (zie BB).
2. Een app voor een Android smartphone, ontworpen en gebouwd in Xamarin.

Technieken, tools en talen

- Arduino in combinatie met processing/C en het interfacen van externe hardware, waaronder een RF-verbinding
- Apps programmeren voor smart phones/tablets in combinatie met Xamarin (C#)/Visual Studio (eventueel Java/Android Studio)
- Communicatie op basis van HTTP en TCP/sockets
- Interactie met externe hardware, w.o. een “klik-aan-klik-uit”-apparaat en diverse sensoren.
- Scrum/Agile voor softwareontwikkeling
- Versiebeheer met Git/GitHub



Benodigde hardware

Per student

- Laptop
- Arduino Uno en losse (ethernet) network-shield Het strekt tot aanbeveling een Arduino-starterskit aan te schaffen.

Per groep

- Smartphone (Android is het meest toegankelijk, iOS mag, maar wordt slechts summier ondersteund vanuit de opleiding)
- Draadloze schakelaar-sets 433 MHz (klik-aan-klik-uit)
- RF-zender/ontvanger, 433 MHz
- Ultrasone afstandssensor en andere sensor naar keuze (licht, temperatuur, kracht, acceleratie, joy-stick, etc.)
- Een (mini) servomotor



Benodigde software

- Visual Studio (>=2013, zo mogelijk 2015 Community edition, Dreamspark)
- Aanvullingen: Xamarin voor Visual Studio
- Arduino IDE (Integrated Development Environment) met libraries en voorbeelden. Kies de laatste versie.
- Voorbeeldcode (Arduino en Xamarin-app) voor een eenvoudig domotica-systeem



2. De opdrachtomschrijvingen

a. A-Opdracht

Breid de gegeven voorbeeldcode (Arduino en Xamarin, zoals beschikbaar gesteld via BB) uit tot een domotica-systeem dat:

1. Minstens 3 RF-stopcontacten afzonderlijk kan aansturen. Van elk stopcontact is op de Arduino én op de app de actuele status te zien (aan/uit). Het interface is functioneel, logisch, responsief en gebruiksvriendelijk. Meer info bij het college “User Experience”.



2. Het interface van de app toont de actuele sensorwaarden afkomstig van minstens twee sensoren, beide verbonden met de Arduino. Je bent vrij in de keuze van je sensoren. Aanbevolen wordt het gebruik van eenvoudige en goedkope sensoren zoals een (analoge) lichtsensor en een (analoge of digitale) temperatuursensor. Mocht je andere sensoren willen gebruiken dan mag dat ook. Denk aan sensoren voor het meten van kracht, hoek (joy stick of potentiometer), versnelling (acceleratie) of afstand.

Er worden geen hoge kwaliteitsnormen aan de sensoren gesteld; het mogen goedkope componenten zijn, zoals de digitale “one wire” temperatuursensor “DS18B20” of een analoge NTC (=warmteweerstand) en de LDR (=fotoweerstand). Dergelijke sensoren zijn niet alleen goedkoop en geschikt voor de Arduino, ze worden vaak ook meegeleverd als onderdeel van een Arduino-ontwikkelkit.

De meet snelheid is instelbaar en ligt tussen 1 en 10 metingen per 10 seconden.

Optioneel is de mogelijkheid om de sensoren via de app aan en uit te kunnen zetten.



3. De app en het interface moeten een mode ondersteunen waarbij de volgende domotica-toepassing te realiseren is: één van de beide sensoren moet, via de app, op een zodanige manier gekoppeld kunnen worden aan één van de drie stopcontacten dat grenzen bewaakt kunnen worden. Voorbeelden: wanneer de temperatuur boven een bepaald en via de app instelbaar niveau komt gaat er een lamp (ventilator of koude fohn) aan of wanneer de hoeveel licht onder een bepaald en eveneens via de app instelbaar niveau komt, gaat er een lamp aan. De bewaking heeft betrekking op een statisch grens. Als de gemeten grootheid weer onder de grens komt gaat de fohn/lamp weer uit.



4. Het systeem ondersteunt een functie waarbij minstens één van de stopcontacten wordt aangeschakeld op grond van een, via de app instelbaar, tijd criterium (vgl. met een kookwekker, lamp aan na 10 minuten of actie op een bepaalde tijd (koffiezetterapparaat aan om 5 uur ’s middags)).
5. De app heeft een flexibel (gebruikers)interface t.a.v. de koppeling tussen Arduino (server) en app (client) voor wat betreft netwerkconfiguratie, met bijvoorbeeld een automatische connectie met de server of met een lijstje met IP-nummers.

Optioneel/Bonus voor de A-opdracht (of alvast ideeën voor B-opdracht). Toepassen van meer sensoren (bijv. luchtdruk, luchtvochtigheid, infrarood, afstand, aanraak)

- Configuratie van de sensoren via de app (bijv. koppeling aan IO-pin, bereik, spanning)
- Grafische weergave van data, met grafiekjes en trends.

- Dynamische grenzen i.p.v. statische grenzen (snelle stijging per tijdseenheid).
- Logging of eenvoudige dataopslag van sensorgegevens.
- Koppeling van tijdfuncties aan Google agenda (lamp aan: 27 jan. 9.00 uur en uit 27 jan. om 14.00 uur).

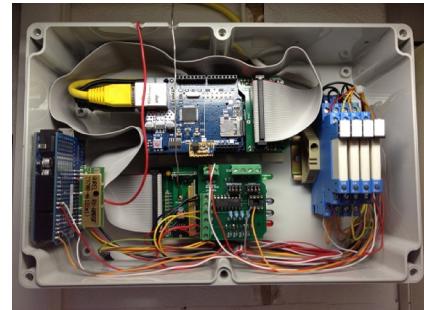
b. B-Opdracht

Bedenk en bouw een **innovatief domotica-systeem** op basis van (minimaal) een Arduino met ethernet en daaraan gekoppelde sensoren/aktuatoren en een Xamarin-app.

Het woord “innovatief” is een beetje een modewoord dat vaak gebruikt wordt in combinatie met nieuwe technische hoogstandjes. De betekenis van het woord is echter een beetje sleets en vaag geworden. Bedoeld wordt een vernieuwend of grensverleggend systeem.

Het kader is daarmee erg ruim. Het voordeel daarvan is dat er veel vrijheid is en daarmee samenhangend de mogelijkheid tot inventieve en creatieve oplossingen. Het nadeel zou kunnen zijn dat de opdracht voor sommige groepen te weinig richting heeft. Schakel in dat geval je tutor of de projectleiding in. In de veronderstelling dat het ambitieniveau hoog licht, geven we een voorzetje met enkele voorbeelden:

- een modern en app-configureerbaar inbraakalarm
- een geautomatiseerde koffiezetter met personificering en agendakoppeling
- een weerstation met geautomatiseerde acties (regenscherm neer bij regen) en lokale weervoorspelling
- een geavanceerde indoor lichtregeling met sfeerdetectie
- een bediening voor binnenhuis audio en video met smart afstandsbediening en Kodi-integratie
- een systeem voor indoor klimaatcontrole (van huiskamer, schuur of kas) met thermostaatkoppeling
- een systeem voor huisbeveiliging met integratie van bijvoorbeeld glasbreuk- en rooksensoren.



Een ander perspectief is die van de doelgroep. Voorbeelden:

- domotica voor ouderen, met registratie van beweeglijkheid of de detectie van calamiteiten, zoals een val
- domotica in de zorg, met logging en bewaking van patiëntdata en slimme alarmering op grond van correlaties van gemeten grootheden
- domotica voor studenten, met app-gestuurde (beetje cliché) bierflesdopper, chipszakopener, alcoholalarmeringssysteem, bedopmaaksysteem met geurdetector, adaptief studenten-weksysteem met schrikdraad-integratie en collegetijden-notificatie of een game-reguleringssysteem met vermoeidheidsdetector, Educator-integratie en mogelijkheden tot ouderlijk toezicht, automatisch inschrijfsysteem voor tentamens met RFID-tags.
- domotica in de mode, met interactieve en awareness kleding (kleding die reageert op de omgeving, zoals nabijheid of aanraak), bio-kleding (kleding die



lichaamsparameters, zoals temperatuur, vochtigheid, vruchtbaarheid, etc.) meet en in beeld brengt of, voor gevorderden, een DIY Harry Potter-onzichtbaarheidsmantel.

Het is een kleine moeite om deze lijst nog uit te breiden met meer thema's of doelgroepen, maar dat laat de organisatie verder aan jullie eigen inventiviteit, creativiteit en originaliteit over. Het idee zal onderhand wel duidelijk zijn. De voorbeelden lijken (en zijn soms ook) wat vergezocht en misschien ook niet te realiseren in 4 weken tijd, maar in aangepast of afgezwakte vorm kunnen ze je wellicht op haalbare ideeën brengen. De boodschap is: wees innovatief, denk "out of the box" en niet in beperkingen. Jullie geven mede vorm aan het Internet Of Things.

Tot slot van deze oriëntatie op domotica willen wij nog een seizoensgebonden thema aandragen, namelijk de **winter**, met tevens enkele suggesties onzerzijds:

- Een domotica-uitbreiding op de kerstboom, met temperatuurafhankelijke kerstballen, lichtgevoelige engeltjes en een Arduino-piek die op commando van de app "*Jingle Bells*" afspeelt.
- Een mooie versiering van je dakgoot met lampjes die via een app van kleur kunnen veranderen, zo mogelijk gepersonifieerd en geïntegreerd met locatiegegevens, zodat wanneer een NHL-Informaticadocent in een straal van 1 km van jullie woning komt de lampjes rood oplichten.
- Of een sneeuwdetectie en -waarschuwingsysteem waarbij, bij temperaturen onder de -2 graden, een sneeuwdikte van minstens 3 cm en vanaf de tijd dat het buiten begint de schemeren, op de smartphone je "*your-perfect-winter-weather playlist*" wordt afgespeeld en rondom de kerst natuurlijk "*I'm Dreaming of a White Christmas*".
- Tot slot: de domotica-oliebollenmepper, waarbij de oliebollen met een ingenieuze servo-systeem uit de olie worden geslingerd, in de vlucht van poedersuiker worden voorzien en, instelbaar via de app, op het juiste bord belanden.



Dit is maar een vrij willekeurige greep uit de eindeloze hoeveelheid mogelijkheden. Jullie oplossingen zijn ongetwijfeld nog innovatiever en creatiever ...

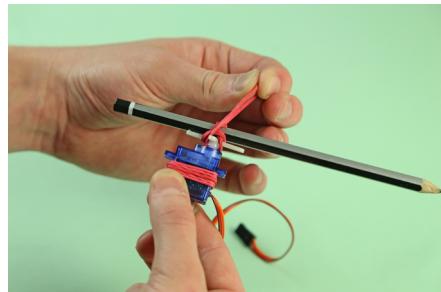
Wat verwachten we? Een demonstratie van een werkend, innovatief domotica-systeem. Leef je uit en zoek je technische en creatieve grenzen op. Imponeer ons, maar maak het niet te ingewikkeld!

De resultaten worden beoordeeld door een vakkundige jury. Een perfecte uitwerking van de hierboven genoemde suggesties geven uiteraard sowieso recht op een 10. Wel even demonstreren natuurlijk. Wat we op dat punt verwachten is omschreven in het hoofdstuk "Organisatie".

c. C-Opdracht

Bij deze opdracht werken meerdere groepen samen. Hierbij gaat het niet om gemeenschappelijke code-ontwikkeling, maar vooral om communicatie, communicatie tussen groepen en studenten en tussen systemen.

Elke app ondersteunt een “mode” waarin deze samenwerking gerealiseerd dient te worden. Eenmaal in deze mode, is elk domotica-systeem (Arduino en app), “gevoelig” voor zijn/haar buren. Elk systeem is daartoe (extra) uitgerust met een servomotor en een ultrasone afstandssensor. Aan de as van de servo is, volgens



nevenstaande foto, een potlood bevestigd. Bij aansturing van de servo dient het potlood over 90 graden te bewegen, van “Hoog” tot “Rechts”. De stand “Hoog” is de rusttoestand. In de stand “Rechts” is het de bedoeling dat dit gedetecteerd wordt door de rechter buur(Arduino), die deze verandering detecteert doordat het potlood in de meetbundel van haar afstandssensor komt. Eenmaal gedetecteerd, laat deze buur(Arduino) op haar beurt de slagboom van “Hoog” naar “Rechts” gaan. Dit herhaalt zich voor meerdere systemen. In principe zou dit moeten kunnen werken voor de systemen van alle groepen. Om e.e.a. beheersbaar en overzichtelijk te houden worden er echter clustergroepen van 6 tot 8 projectgroepen gevormd. Elke clustergroep moet deze multi-slagboomoperatie uitvoeren. Hoe gaat dat precies in zijn werk?

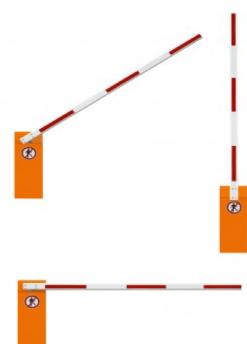
DS1 (domoticasysteem 1) krijgt van de projectleiding de opdracht om via de app haar slagboom te laten zakken, waarop DS2, m.b.v. de afstandssensor, automatisch reageert door ook haar

slagboom te laten zakken, waarna DS3 volgt, etc. Dit herhaalt zich tot en met DS n (met n = het aantal projectgroepen in een clustergroep). Wanneer DS1 de slagboom van DS n “ziet” doet deze haar slagboom weer omhoog, waarop DS2, wederom op basis van sensorinformatie, dat ook doet, waarna de rest van de keten volgt. De opdracht is voltooid wanneer de slagboom van DS n weer omhoog staat. Alle systemen hebben de slagboom dan één keer dicht en één keer open gedaan. Het is natuurlijk de bedoeling dat dit proces zich zonder verder menselijke tussenkomst voltrekt. Mocht dat lukken, dan vinden wij, van de

projectleiding, het niet vreemd dat het omhoog gaan van de laatste slagboom gepaard gaat met een zeker collectief gevoel van trots en tevredenheid, zelfs niet wanneer daarbij op gepaste wijze wordt geapplaudisseerd.

De projectororganisatie bepaalt welke groep welk DS-nummer krijgt. Ieder groep moet er dus op voorbereid zijn de rol van DS1 te moeten kunnen vervullen (aanzetten *met* de app). Tijdens het hele proces dient elke app steeds de toestand van de slagboom zichtbaar te kunnen maken, bijvoorbeeld met informatie over of er zich een slagboom (of object) op korte afstand van de sensor bevindt. De vormgeving is vrij.

Het is aan jullie (studenten, groepen/groepsleiders) om te bepalen hoe deze groepsopdracht, in termen van opstelling, sensoren, app-integratie, etc. gerealiseerd gaat worden. Communicatie met andere groepen is hiervoor noodzakelijk. Waarschijnlijk is het verstandig de projectleiding op sommige punten ook bij deze communicatie te betrekken.



Nog een opmerking van technische aard: positionering in een cirkel lijkt handig, maar is dat, om technische redenen, niet. Bedenk dat een ultrasone sensor een (ultrasone) geluidspuls uitzendt en daarna wacht op de reflectie daarvan. De gemeten tijdsduur tussen het verzenden en ontvangen is een maat voor de afstand. Omdat alle sensoren op dezelfde (ultrasone) frequentie werken, kan een sensor geen onderscheid maken tussen een reflectie van haar eigen golf of een primaire golf van een tegenoverliggende sensor. Oppassen dus! Als een cirkel niet werkt dient er zich een andere vraag aan, nl. hoe geeft systeem DS_n door dat DS₁ weer aan de beurt is (met $n > 1$). Dat is alles wat wij er over zeggen. Het is verder aan jullie. De sleutel tot succes is waarschijnlijk een goede onderlinge communicatie.

3. De organisatie

Het project kent naast ca. 120 studenten ook nog tutoren (voornamelijk docenten), experts en studentassistenten. Er is ook een projectorganisatie. Die wordt gevormd door de docenten S. Oosterhaven en J. Foppele. Zij zijn de “eigenaren” van het project en zorgen voor de organisatie, de informatie en de communicatie, tijdens alle fasen van het project, waaronder de voorbereidingsfase. Naast de aftrap en enkele flankerende colleges organiseren en bemannen zij gedurende de looptijd van het project wekelijks een vragen(half)uur (donderdag tussen 11.00 en 11.30 uur, bij D3049). Iedereen is daar welkom met vragen, graag domotica-gerelateerd.

De uitvoering van het project

- De groepsgrootte is max. 6 studenten per groep. De groepsindeling is vrij. Vanuit het vak PPO wordt e.e.a. nog beargumenteerd (zie huiswerkopdrachten). Aanmelden via Blackboard.
- Alle groepen krijgen (of kiezen) een tutor. De groep zoekt op, zo mogelijk al op de eerste dag, contact met haar tutor. Tutor en groep spreken elkaar minstens 2 keer per week.
- Op grond van goed projectmanagement en softwareontwikkeltechnieken, wordt de rolverdeling (w.o. projectleider, voorzitter, documentalist, scrummaster, etc.) in overleg met de tutor vastgesteld aan het begin van de eerste projectweek en zo mogelijk al in de week voorafgaand aan de eerste projectweek.
- De uitvoering van het project staat uiteraard ook in het teken van het (groeps)proces en (de vakken) software engineering en PPO. De docenten die daarbij betrokken zijn zorgen voor aanvullende eisen en begeleiding, waaronder:
 - Samenwerking, groepsdynamiek en projectmanagement. Dit komt vooral tot uitdrukking in de tutorbegeleiding, het scrum-proces en de ondersteunende opdrachten, zoals een team-contract, vanuit het vak PPO.
 - Binnen elke groep wordt de code, Xamarin en Arduino, met Git of Github beheerd.
 - Testen speelt gedurende het project een belangrijke rol. Wat test je, waarom test je, hoe test en hoe verantwoord je dit allemaal?
 - Om een team, bij het proces van softwareontwikkeling, de nodige zelfsturing en effectiviteit te geven worden diverse elementen van de Scrum/Agile-methodiek toegepast. De vakdocenten en tutoren spelen hierbij een belangrijke rol.
 - In de laatste week (week 4) van het project worden alle groepen “onderworpen” (uitgenodigd klinkt vriendelijker, maar het is wel verplicht) aan een technisch assessment. Elke projectgroep wordt hierbij in ca. 15 minuten door een groep docenten, volgens een vaststaande procedure, bevraagd over minimaal de verplichte onderdelen van de opdracht. De vragen zijn technische van aard en moeten een goede indruk geven van de kwaliteit van de producten, de technische diepgang en de ontwerpkeuzes. Er wordt bij dit assessment ook gekeken naar de “schema’s op hoofdlijnen” van je domotica-systeem. Het gaat dan om bijvoorbeeld een architectuurschema, een structuurschema, een blokschema, een functioneel schema, een klassendiagram, een client-server-schema waarin de interfaces (en protocollen) opgenomen zijn, etc. Schema’s dus, op grond waarvan er op hoofdlijnen en op helder wijze over het systeem gesproken kan worden, zowel vragen als antwoorden. Een code review maakt deel uit van dit assessment. De resultaten van het technisch assessment zijn individueel en dragen bij aan de totaalbeoordeling. Uiteraard kunnen meerdere leden van een groep hetzelfde cijfer krijgen. Meer informatie over de assessments is te vinden in bijlage 2 (beoordelingsschema) en op BB (projectplanning).

- Het is de bedoeling om samen, met alle studenten, docenten en genodigden, in een competitieve en tevens feestelijke sfeer, het project af te sluiten. Dat doen we in de theaterzaal van de NHL De afsluiting van het project kent drie activiteiten:
 - 1) De demonstratie
Alle groepen demonstreren in maximaal 5 minuten per groep, volgens een vooraf bekend gemaakte rooster, de B-opdracht, het innovatieve domoticasysteem.
 - 2) De slagboomdemo
De tweede verantwoording is die waarbij in drie cluster meerdere groepen tegelijkertijd in actie komen voor de demonstratie van de opdracht waarbij met meerdere domotica-systemen een keten van slagbomen in beweging moet worden gesteld.
 - 3) Filmpjes met toelichting.
Alle groepen verantwoorden, ten overstaan van een goed gevulde zaal en een bekwame jury, hun werk waarbij, eveneens volgens een rooster, elke groep m.b.v. een filmpje van exact 2 minuten hun domotica-systeem in actie toont. Voorafgaand aan het starten van de film presenteert de groep zich in 1 minuut, door zich voor te stellen en een korte toelichting te geven op de film. De jury beoordeelt jullie werk en wijst ook een top 3 aan met een winnaar.

Alle onderdelen dragen bij aan de individuele beoordeling waarover hieronder nadere uitleg volgt.

De beoordeling

In de eindfase van het project krijgt iedere student een beoordeling. Deze beoordeling is in principe individueel, maar, bij normaal verloop, vooral gebaseerd op een groepsbeoordeling. De tutor, die een belangrijke bijdrage levert aan de beoordeling, kan, als daar voldoende aanleiding voor is, een voorstel indienen voor een gesplitste beoordeling.

Het (individuele) technische assessment levert ook een substantiële bijdrage aan de individuele beoordeling. Er gelden drie beoordelingscriteria:

1. Arduino, hard- en software
2. Phone/app programmering
3. User Experience app

Elke student wordt beoordeeld op twee van de drie criteria. Dit dient voorafgaand aan het assessment te worden aangegeven. Daarbij wordt aan elke student gevraagd aan welk deel van de code hij of zij gewerkt heeft. Hij of zij dient hierbij een zelfgemaakt onderdeel van de code uit te leggen. Aan de groep wordt gevraagd hoe groot de individuele bijdrage van elk groepslid is geweest.

Elke groep maakt, ter voorbereiding op dit assessment, een schema van de architectuur van het totale domotica-systeem (met embedded onderdeel en app), zoals hierboven uitgelegd.

- De grootste bijdrage aan de totale beoordeling is ingeruimd voor de technische uitwerking en bijbehorende demonstratie van de drie opdrachten. Dit is in principe gebaseerd op de groepsprestatie, maar daar kan van worden afgeweken. Belangrijke criteria hierbij zijn: kwaliteit, niveau, creativiteit, innovativiteit, originaliteit en hoeveelheid van het geleverde werk. Alle beoordelingsaspecten zijn gekoppeld aan onderliggende competenties en de daarvan afgeleide criteria. Deze zijn in een overzichtelijk schema opgenomen in bijlage 2.

Speciale aandacht is nog nodig voor een nieuw element in de eindverantwoording, namelijk het filmpje dat elke groep moet maken en tonen. Dat moet aan de volgende criteria voldoen:

- De filmduur is exact 2 minuten.
- Elke groep krijgt exact 1 minuut om zich voor te stellen en de film in te leiden of daar kort toelichting op te geven.
- De inhoud van de film is zodanig dat deze een goed en waarheidsgetrouw beeld geeft van werking van het domotica-systeem. Zorg dat alle belangrijke functies van het systeem duidelijk worden uitgelegd en goed in beeld komen, ook de details.
- De beelden moeten worden ondersteund door gesproken tekst (bijv. als voice-over). Zorg ervoor dat dit met helder, goed verstaanbare stem gebeurt. Uiteraard mogen de beelden ook voorzien worden van (geschreven) tekst.
- De vormgeving, stijl, genre en layout van de film is vrij. Voorbeelden: Een domotica-systeem, dat als hoofdpersonage in een “coming-of-age”-achtige stijl, volwassen wordt of een domotica-systeem dat zich in een “film noir”-achtige stijl wapent tegen de existentiële bedreigingen in een nihilistische wereld.
- Enige muzikale omlijsting strekt tot aanbeveling, net als het toepassen van (een gedoseerde hoeveelheid) humor.
- De filmpjes moeten een dag voor de afsluiting, uiterlijk 17.00 u, worden ingeleverd bij de projectleiding. Dat is o.a. nodig voor een vlotte doorloop tijdens de afsluiting.

Ook nieuw is het onderdeel “User Experience”. Dit wordt getoetst en beoordeeld bij de diverse demonstraties van de eindproducten en, zoals hierboven vermeld, bij het technische assessment. Specifieke criteria voor het onderdeel User Experience:

- is de gedurende het project geven feedback terechtgekomen in het eindresultaat
- gebruiksgemak van app, interactie, doelgroepgericht
- klopt de interface met het platform (telefoon/tablet)?
- Bonus: Is het goed te bedienen, ook met 1 duim als je net wakker bent?
- Bonus: Ziet het er “gelikt” uit?

Randvoorwaarden bij het project

Algemeen

- Studenten krijgen voldoende voorbeeldcode om een snelle start te maken, zowel voor de Arduino als voor de app.
- De Arduino kan, via een ethernetkabel, en een adhoc-netwerk, eenvoudig op een laptop worden aangesloten. Handiger is het echter om een (eigen, oude, goedkope en instelbare) router te gebruiken. Dat hoeft beslist geen Gigabit-router te zijn. Meer informatie over dit onderwerp is te vinden in de laatste opdracht van het vak embedded systems.
- Alle project- en vakinformatie is te vinden op Blackboard (Organization Informatica -> Courses -> Jaar 1 -> Periode 2 -> Project Domotica en tevens bij het vak Embedded systems 1).
- Tijdens de uitvoering van het project is er ondersteuning vanuit de tutor en zijn er meerdere experts beschikbaar voor technische vragen. Ook de studentassistenten, die ondersteuning bij de practica boden en nog bieden, zijn tijdens het project beschikbaar voor vragen.
- De projectteam kunnen gedurende het hele project werken in de bekende practicumlokalen, waar de studentassistenten te vinden zijn, of elders in de school, in studielandschappen of



break out-rooms.

Per student

- Iedere deelnemende student heeft het practicum van programmeren 1 (PRG1) met goed gevolg afgerond en volgt, voorafgaand aan en deels parallel met het project, Programmeren 2 (PRG2) en de cursus Embedded systems (EMS1)
- Iedere student zet zich, in redelijkheid en naar beste vermogen, in om het project voor hem/haar-zelf én voor de groep tot een succes te maken.

Meer informatie op BB: studiewijzer, jaarprogramma.

Fasering en planning (met weeknummers gerelateerd aan de start van de periode).

[week 3 en 4]

- Indeling van de groepen en toedeling tutoren. Dit gaat via BB en onder regie van het vak PPO.

[week 5]

- Alle studenten organiseren een eigen ontwikkelomgeving, in aanvulling op hun C#-practicum, voor Xamarin en, voor zover dat nog niet gebeurd is, voor hun embedded-omgeving. Xamarin dient als plugin van VS te worden geïnstalleerd. Afzonderlijke aanmelding bij de Xamarin-site is niet (meer) nodig. Voor Arduino moeten enkele libraries worden ontsloten, o.a. voor het Ethernet-shield, de RF-interfaces en klik-aan-klik-uit-devices en voor de servomotor en de afstandssensor.

[week 6]

- [ma. 19/12/2016] Aftrap van het project voor alle studenten en andere betrokkenen. Aspecten: uitleg doel, opdrachten en organisatie; tevens toelichting code en informatie op BB; laatste controle op indeling groepen en toedeling tutoren.
- Alle studenten bestuderen de gegeven voorbeeldcode. Met deze code kan in principe een werkend domotica-systeem worden gemaakt.
- Alle studenten proberen deze code “aan de praat te krijgen” op hun eigen (of groeps-) Arduino en in Xamarin/Visual Studio. Mogelijke problemen: Xamarin-versies, MS Visual Studio registratie, Android SDK-manager en Android versie, koppeling en development voor (eigen) telefoon, GUI-designer, hardware-aspecten bij Arduino (sensoren, draadjes, solderen, etc.). Samen lossen we deze problemen op.
- Het project wordt met enkele flankerende colleges ondersteund, waaronder app-ontwikkeling, embedded systems en user experience.
- Studenten werken volgens de Scrum-methodiek met minimaal (en onder toezicht van de tutor): zelfsturing binnen het team, gericht op werkende software, “user stories”, “daily scrum” met “stand-ups”, “product backlog” en “testing”. Een klassieke planning is niet nodig.

[week 6, 7, 8, 9]

- Werken aan de opdrachten en zo nu en dan, volgens het rooster, een flankerend college.

[week 8]

- Verantwoording van enkele PPO-gerelateerde zaken onder de regie van de docent.

[week 9]

- Technische assessment
- Afsluiting (vrijdag 27/1/2017)
Tijdens een even competitieve als feestelijke afsluiting wordt alle groepen geacht hun werk te verantwoorden middels demonstraties en filmpjes met toelichting.

4. Tot slot

Het domotica-project kent veel vrijheden, vrijheden in ontwerpen, hardware, elektronica en software. Dat wordt door bijna iedereen als prettig ervaren. Het is geen “dichtgetimmerd” project, maar er zijn marges en er is veel ruimte voor eigen inbreng en creativiteit. Het is een natuurlijke houding van een ingenieur(-in-opleiding) om de rek in deze marges en grenzen op te zoeken. Wij, als organisatie en leiding van het project, proberen de onzekerheden en de discussies die dat met zich meebrengt te voeren in een sfeer van openheid, redelijkheid en fatsoen, zodat iedereen voor zichzelf en de groep het maximale uit dit project kan halen. Wij hebben er geen enkele twijfel aan dan dat alle andere betrokkenen dezelfde instelling hebben, waardoor het “als vanzelf” en op natuurlijke manier een leuk, leerzaam en uitdagend project wordt.

Risico's

Elk project kent risico's. Die liggen niet in datgene wat hierboven is aangegeven, maar eerder in het werken met embedded systemen, elektronica en andere, voor velen nieuwe, hardware. Bij het practicum Embedded is duidelijk geworden dat embedded software sterk lijkt op “gewone” software. Het grote verschil is echter de context: embedded software heeft ten doel om de omgeving te leren kennen en deze te beïnvloeden, doorgaans niet via muis en beeldscherm, maar via fysische grootheden. Aan de inputkant: sensoren, aan de outputkant: actuatoren. Aan beide kanten zitten overduidelijk, maar gelukkig beheersbare, risico's:

Een bit in software is abstract (false of true, nul of één), maar een bit aan de uitgang is doorgaans een signaalpin waar “echte” spanning op staat, die bij het aansluiten van een (elektronische) component leidt tot een “echte” stroom. Zowel de spanning als de stroom kunnen, bij onjuist gebruik, dermate hoge waarden aannemen dat een component letterlijk in rook op gaat. Het is dan te hopen en afhankelijk van de exact details welke component dat is: een LED, je Arduino of je eigen laptop. Enkele regels:

- Wees voorzichtig met het aansluiten van externe hardware. De populaire strategie bij softwareontwikkeling (eerst proberen, dan begrijpen) werkt hier niet goed.
- Controleer de pinout van externe schakelingen, zoals sensoren en transistoren. Teken vooraf een schema. Controleer de werking van de componenten of laat dat controleren. Zijn ze analoog of digitaal? Welke spanning hebben ze nodig? Hoe moeten ze worden aangesloten? Hoe werken ze?
- Controleer de polariteit van componenten. Begrijp wat je doet, hoe je het aansluit en wat er mis kan gaan. Vraag in geval van twijfel hulp bij iemand die het wel zeker weet. Maak geen kortsluitingen. Een draadje tussen de +5V (of een signaalpin) en de GND vormt een kortsluiting, een weerstand niet. Google nog even naar de wet van ohm ($U = I * R$ en vooral $I = U/R$).
- Het is een conventie om voedingsdraden die spanningsvoerend zijn (de +) met rode draadjes uit te voeren en de voedingsdraden die de nul-potentiaal hebben (de - of de GND) met zwarte. Gebruik voor de signaaldraden andere kleuren, zo mogelijk gedifferentieerd naar functie. Dus bijvoorbeeld: wit voor de servo, geel voor de afstandssensor, blauw voor de RF-zender, etc. Houd je aan deze conventie. Schakelingen zijn (door deskundigen) veel eenvoudiger te “lezen”.
- Sluit, als je niet zeker weet wat je doet, geen externe spanningen (zoals 12 V) op je Arduino. Een dergelijke spanning kan via de USB-poort in direct contact komen te staan met je laptop, die daarvan stuk kan gaan.
- Sluit geen motoren rechtstreeks aan op signaalpinnen of voedingspinnen van de Arduino. Motoren, lineaire, stappen- of servomotoren, nemen veel stroom. Die moet geleverd worden door een externe bron. Pas in dat verband ook op met servo-motoren. Er zijn types bij die je niet kunt voeden vanuit de USB-poort van je laptop en ook niet via een eenvoudige

voedingsadapter. De voorgeschreven mini-servo kan *wel* gevoed worden vanuit de Arduino (en de daarvan gekoppelde laptop).

- Gebruik een externe voedingsadapter zodra je componenten gaan aansluiten die meer dan ca. 100 mA aan stroom vragen.
- Mocht er een branderige lucht in je omgeving ontstaan trek dan in ieder geval de USB-poort van je laptop los en zet de Arduino uit. Geniet pas daarna van de geur.
- **Pas op:** het werken met **230 Volt** is een risico van een nog andere orde: daarmee kunnen niet alleen apparaten stuk gaan, maar ook mensen. En dat is niet de bedoeling. Met relatief eenvoudige schakelingen, zoals relais, kunnen externe 230V-apparaten, zoals lampen, fôhns, koffiezetter en andere huishoudelijke apparaten worden geschakeld. Pas op dat een relais naast een "zwakstroom"-kant (die je aansluit op de Arduino) ook een "sterkstroom"-kant heeft via welke je de 230V(apparaten) schakelt. Het is een kleine moeite via deze doorgaans weinig afgeschermd componenten een flinke schok te krijgen. En dat is niet alleen "niet prettig", het kan ook **dodelijk** zijn.
- Om deze en nog andere redenen gebruiken we de klik-aan-klik-uit-hardware. Deze stopcontacten zijn geïsoleerd en afgeschermd en via een RF-zender (laagspanning) te schakelen. Let op dat een klik-aan-klik-uit-stopcontact, geplaatst in een gewoon stopcontact, uiteraard dezelfde risico's met zich meebrengt als zo'n gewoon stopcontact. Oppassen dus! Meer informatie over de klik-aan-klik-uit-schakelaars vind je in bijlage 3.



Tja, wie durft er na zoveel waarschuwingen nog aan het project te beginnen? Als je, echter, je gezonde verstand gebruikt en weet wat je doet is er niets aan de hand en kun je naar hartenlust experimenteren en je kennis en vaardigheden binnen dit project enorm vergroten. Als je echt niet meer durft, schrijf dan een klik-aan-klik-uit-simulator, bijvoorbeeld in PHP of zo. Dat is ook leuk en voordat er rook uit PHP komt...

De projectorganisatie wenst alle studenten en andere betrokkenen een leuk, leerzaam en uitdagend project toe. Wij zijn erg nieuwsgierig naar de uitkomsten en natuurlijk vooral naar de filmpjes.

Dec. 2016, De Domotica-projectleiding

Bijlage 1, Leerdoelen en competenties

Het domotica-project is, zoals veel eerstejaarsonderwijs, onderwijs op niveau 1. Dit heeft betrekking op de onderliggende competenties, waarbij geldt: niveau 1: beginners, niveau 2: gevorderden, niveau 3: expert. Het project is, conform de kaders van de NHL, praktijkgericht en daarnaast tevens gericht op integratie van kennis, vaardigheid en houding en, in het verlengde daarvan tevens op de concrete beroepspraktijk voor een software engineer. De inhoud van veel, aan het project voorafgaande, vakken en onderwijsactiviteiten komen terug of worden toegepast binnen het project: programmeren, embedded systems, analyse, software engineering. Dat geldt ook voor de persoonlijke ontwikkeling (PPO), met oriëntatie op (eigen) talenten, motivatie, groepssamenstelling, het werken aan zelfstandigheid en zelfvertrouwen en de opbouw van het persoonlijk portfolio.

Toelichting competenties op niveau 1: alle aspecten sluiten aan bij beginfase van de body of knowledge: het project is overzichtelijk en van korte duur (4 weken). Het kent een heldere structuur met duidelijke fases en mijlpalen. De opdrachten zijn concreet, opgedeeld in kleine onderdelen en kennen een opbouw in toenemende complexiteit. De basisuitwerking is gebaseerd op relatief veel gegeven voorbeeldcode. Eén van de eindproducten biedt daarnaast de mogelijkheid tot substantiële verdieping en complexiteit, ook als extra uitdaging voor de "goeie student" en tevens de mogelijkheid tot inzetten van innovativiteit en creativiteit, zowel in oplossingen, ontwerp, code als vormgeving. Binnen duidelijke kaders is er voldoende ruimte voor individuele- en groepsvrijheid. Tegelijkertijd is er tevens een duidelijke individuele en groepsverantwoordelijkheid, ook en vooral voor het eindproduct. De begeleiding is systematische, met wekelijkste begeleidingsmomenten (2x/week, volgens rooster) en relatief veel docentsturing. De beoordeling is volgens heldere criteria en vooraf bekend (zie schema in bijlage 2). Er zijn wekelijkse mijlpalen, met o.a. een PPO-verantwoording en een technische assessment. Bij beide onderdelen wordt direct feedback gegeven. Tot slot is nog het tutoraat. Elke groep vergadert minimaal 2 keer per week met de tutor. Zijn/Haar oordeel blijft niet beperkt tot na het project.

Bijdrage aan de competenties (niveau 1)

Onderzoeken:	systematisch en methodisch werken, volgens moderne technieken (Scrum/Agile)
Analyseren:	voorbeeldcode analyseren; onderdelen en functies benoemen. aanvullende eisen opstellen
Adviseren:	communiceren binnen eigen groep en daarbuiten (bij superopdracht) presenteren (voorstellen, film maken, toelichting geven)
Ontwerpen:	ontwerpmethodes toepassen; prototype ontwerpen; keuzes onderbouwen; eenvoudig architectuurplan opstellen (client-server-model met socket-communicatie); oriëntatie op testen: voorbereiden en meedoen aan testsessie.
Realiseren:	prototype maken; opdelen in fasen en modules, client: app; (embedded) server: Arduino met extra hardware; concreet resultaat opleveren en verantwoorden
Beheren:	beheren eigen en groepsoftware-omgeving (tools, project, versiebeheer, Github)
Managen:	projectplan opzetten; plannen, samenwerken
Professionaliseren:	flexibele opstelling; inlevingsvermogen; dilemma's onderkennen; constructief samenwerken; reflecteren

Vertaald naar concrete methoden, technieken, tools en talen:

- Algemeen software engineering en software-ontwerp (w.o. specificeren)
- Ontwerpen van systeemdelen (client/server). Toepassen van standaard methoden en technieken (w.o. Scrum/Testing/Projectmanagement)
- Programmeren (C#, C/Processing, facultatief: Java)

- Smartphone-programmering (Xamarin, programmeermodel voor smart devices/tablets), inclusief ontwerpen, programmeren en user experience.
- Netwerktechnologie, met communicatie van systeemdelen, ethernet met sockets en TCP/IP en serial (over USB) verbinding
- Interfacing van hardware en aansturing van I/O (w.o. sensoren, aktuatoren en een RF-zender)
- Testen (testmethoden en testrapportage) en code review
- Beheren van eigen ontwikkelomgeving met Github

Bijlage 2 Beoordelingscriteria en beoordelingsschema. De definitieve versie is als afzonderlijk pdf-document te vinden op BB.

Domeincompetenties, Informatica, niveau 1										Beoordelaar			Student 1 Student 2 Student 3																	
<small>Groeps- en individuele beoordeling P2P-project ("Domotica-project") 2016/2017, versie 1</small>																														
<small>T,E,P T = Tutor, E = Expert / Opleiding, P = Projectleiding / Opdrachtgever</small>																														
Analysen	Ontwerpen	Realiseren	Beheren	Managen	Adviseren	Onderzoeken	Professionaliseren																							
			X	X	X		X																							
X	X	X			X	X	X																							
			X	X																										
X	X	X						X	X																					
<small>ligt vast door de beoordelaar in te vullen wordt door de spreadsheet uitgerekend</small>																														
<small>Student</small>																														
T	Functioneren in projectgroep			op basis van			20%			7 10 8																				
E	Verantwoording			op basis van			30%			7 10 8																				
T	Groepswerk			op basis van			10% 7,5																							
E,P	Resultaat			Technische demonstratie en toelichting			40% 6,5			6 8																				
<small>Individuele cijfers</small>																														
<small>Beoordeling</small>																														
<small>Voor het technische assessments wordt een afzonderlijke beoordelingsformulier gebruikt</small>																														
<small>Jury-criteria: innovativiteit, originaliteit en creativiteit</small>																														
<small>De kwaliteit van het eindproduct komt tot uitdrukking in een combinatie van Resultaat en Verantwoording</small>																														

Bijlage 3, Klik-Aan-Klik-Uit-schakelaars (KaKu).

klik-aan-klik-uit, Action

Van de klik-aan-klik-uit-schakelaar (model Action), die aanvankelijk in de studiewijzer werd aangelezen, is ondertussen een nieuw model op de markt verschenen. Zie foto hiernaast. Gelukkig zijn beide modellen bruikbaar. Op te voorkomen dat je het koffiezetterapparaat bij de buren aanzet (wat onder sommige omstandigheden wellicht gewenst is, maar voor ons nu even niet), hebben de schakelaars naast een apparaat-code (A,B,C,D,E of 1,2,3,4) een systeem-code. De oude schakelaars maken gebruik van een 10 bits code, waarvan 5 voor de systeem-code, die handmatig, met dipswitches, in te stellen is. De nieuwe apparaten zijn zelflerend en kunnen niet meer handmatig geconfigureerd worden. Bovendien is de totale code nu 24 bits, wat, zeker met veel buren, prettier en veiliger is. Een bijkomend voordeel is dat het nieuwe model ook met een aan/uit-



schakelaar op het schakelapparaat zelf te bedienen is. Na het voor de eerste keer aanzetten van een apparaat (= in stopcontact plaatsen) knippert het rode lampje tot dat je met de afstandsbediening contact maak, door herhaald te drukken, op "On". Het lampje gaat daarna uit. Je kunt dit herhalen voor de "off"-toestand en daarna voor de andere schakelapparaten. De eenmaal ingesteld codes zijn bij normaal gebruik natuurlijk onzichtbaar. De 24-bits code wordt gebruikt om een 433 MHz draaggolf te moduleren (vgl. met WiFi). Met de RF-zender, die wordt aangestuurd via een signaalpin van je Arduino kun je eveneens 24-bits codes versturen. De twee vragen die zich daarbij natuurlijk aandienen zijn: "Wat zijn de (geleerde/ingestelde) codes?" en "hoe verstuur ik de code vanuit de Arduino?". Gelukkig zijn er genereuze mensen die hun code voor andere beschikbaar stellen.

klik-aan-klik-uit, Action, Nieuw (EuroDomest 972080)

Op GitHub is een project te vinden waar je Arduino-code kunt downloaden waarmee je de nieuwe schakelaars kunt bedienen. RC staat hierbij voor Remote Control.

<https://github.com/sui77/rc-switch>

Download de software en plaats, volgens de aanwijzingen, de directory "RCSwitch-2" in zijn geheel in de libraries-directory van de Arduino-omgeving (hier tref je ook de Ethernet-modules aan). Na het opnieuw opstarten van je Arduino-software tref je onder Voorbeelden -> RCSwitch-2 negen programma's aan. Twee daarvan kun je gebruiken om je schakelaars aan de praat te krijgen en om daarna de code te integreren in je eigen applicatie. Met ReceiveDemo_Simple kun je de 24-bits codes van jouw afstandsbediening achterhalen. Sluit daartoe het receiver-printje aan met de data-ingang op pin 2 (dat is om technische redenen essentieel). Uiteraard moet het printje ook gevoed worden met een +5Volt-signal en een GND-signal. Controleer de aansluiting zorgvuldig. Het omgekeerd aansluiten van + en - is fataal voor de zender en ontvanger. Het ReceiveDemo_Simple-programmaatje toont, via de Serial Monitor, de door jou ingestelde codes. Noteer of registreer deze codes en gebruik ze in het tweede programmaatje: SendDemo. Sluit het Send- of Transmitter-printje aan met de data-uitgang op een vrije Arduino-pin, bijvoorbeeld pin 3. Uiteraard moet ook dit transmitter-printje gevoed worden met +5 en GND. De bruikbare onderdelen van de code zijn de volgende:

```
#include <RCSwitch.h>      // RCSwitch wordt oranje na juiste installatie

RCSwitch mySwitch = RCSwitch(); // declaratie van mySwitch
mySwitch.enableTransmit(3);    // activeer de sender, op pin 3

mySwitch.send(2210415, 24);   // zet de lampen bij fam. Oosterhaven aan
delay(100);
mySwitch.send(2210414, 24);   // zet de lampen bij fam. Oosterhaven uit
delay(100);
```

Uiteraard gebruik je je eigen codes, die je hebt uitgelezen m.b.v. ReceiveDemo_Simple.

klik-aan-klik-uit, Action, Oud (met opschrift "Impuls")

Download, eveneens via GitHub, de code van Randy Simons:

<https://github.com/hjgode/homewatch/tree/master/arduino/libraries/RemoteSwitch>

Deze code werkt met de oude schakelaars. Installeer (kopieer) de gedownloade directory in zijn geheel naar cd libraries directory van de Arduino. Na het opnieuw starten van de Arduino verschijnen onder Voorbeelden -> RemoteSwitch vier programma's. Laad "LightShow" in de Arduino. Toelichting op de bruikbare onderdelen van de code:

```
#include <RemoteTransmitter.h>

ActionTransmitter actionTransmitter(4); // New Transmitter on pin 4

actionTransmitter.sendSignal(31,'A',true); // Switch on Action-device A
delay(1000);
actionTransmitter.sendSignal(31,'A',false); // Switch off Action-device A
delay(1000);

// The system-code (dipswitches) of the Action-device is 31 (alle bitjes 1).
```

Randy Simons heeft ook een library gemaakt voor het nieuwe schakelapparaat (NewRemoteSwitch) maar deze krijgen wij niet aan de praat. Volg dus liever de hierboven beschreven methode.

klik-aan-klik-uit, Action, Nog Nieuwer (ProMax 75.006.14 / Elro Flamingo F500S)

Recentelijk (dec. 2017) heeft de Action al weer afscheid genomen van het device dat hierboven nog als "Nieuw" werd aangemerkt. Deze derde telg in de reeks, de Promax, die in de winkel nog steeds 10 euro kost, is helaas niet bruikbaar voor het Domotica-project. De encryptie tussen afstandsbediening (of Arduino) en het draadloze stopcontact is ingewikkelder, maar daarmee natuurlijk wel veiliger, gemaakt. Nadere research heeft geleerd dat de verstuurde bitpatronen niet altijd gelijk zijn, analoog aan hoe dat bij wifi met het gebruik van meerder kanalen is geregeld. Daarmee is de kans dat je de lampen van de buren aanzet verkleind, maar tegelijkertijd is de kans ook verkleind dat je de functie van de afstandsbediening door de Arduino kunt laten overnemen. Kortom dit type bevelen we niet aan.



klik-aan-klik-uit, APA3-1500R (o.a. Gamma)

Het setje dat door de projectleiding wordt aanbevolen voor het project van leerlaar 2016-2017 draagt de naam APA3-1500R en is bij meerdere aanbieders te verkrijgen, o.a. bij de Gamma. Na enig speurwerk moet dit voor onder de 20 euro wel te vinden zijn. Waar "APA3" voor staat is niet duidelijk. De 1500 verwijst naar de max. 1500 watt die met het apparaat te schakelen zijn. Koffiezetter met een vermogen <= 1500 Watt zijn hiermee aan te sturen.

Het device kan worden aangestuurd met de code uit de onderstaande library:

<https://github.com/hjgode/homewatch/blob/master/arduino/libraries/NewRemoteSwitch>



Installeer de library in je Arduino-omgeving (zie op de Arduino-site hoe dat moet). Onder Voorbeelden vind je nu NewRemoteSwitch, met daaronder vijf voorbeeldprogrammaatjes. Met ShowReceivedCode kun je achterhalen wat de unitcode van je device is. Let op deze kaku devices zijn zelflerende. Dat heeft als consequentie dat je m.b.v. je afstandsbediening, vóór regulier gebruik, contact moet maken met je devices om de codes aan te leren. De via de afstandsbediening geleerde codes worden, via de Serial Monitor, zichtbaar gemaakt. Deze kun je vervolgens gebruiken in je eigen besturingsprogramma. Let op: uiteraard moet de je RF-receiver aansluiten om deze codes te kunnen ontvangen. Let ook hier goed op: de data pin van de receiver moet aan Arduino pin 2 (=interrupt-ingang) worden aangesloten. Bij zowel de transmitter



als de receiver is het uiteraard goed oppassen met het aansluiting van de voeding (+ en -). Weet wat je doet. Check en dubbel-check. Het verkeerd aansluiten van de voeding is fataal voor beide modules. Het is te hopen dat je dan voor 2 euro plezier beleefd aan de rook en de geur. Code-voorbeelden voor de APA3.

```
#define RFPin      3           // pin to control the RF-transmitter
#define unitCodeA 21177114     // replace with your own code

#include <NewRemoteTransmitter.h> // Remote Control, APA3, Gamma

//declaration of device
NewRemoteTransmitter apa3Transmitter(unitCodeApa3, RFPin, 260, 3); // APA3

// in setup code
pinMode(RFPin, OUTPUT); // RFPin is output

// in loop code
apa3Transmitter.sendUnit(0, true); // switch on unit 0
delay(100);
apa3Transmitter.sendUnit(0, false); // switch off unit 0
delay(100);
```

Handige link, met een overzicht van veel gangbare kaku-devices en bijbehorende technische details:

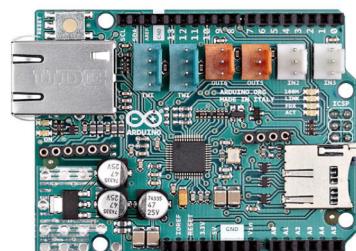


- <https://arduinodiy.wordpress.com/tag/promax/>
- Meer over Klik-aan-Klik-uit, met soorten, types, zender, ontvanger, software, etc.:
<http://www.klikaanklikuit.nl/home/>

Als je de werkzame afstand tussen de zender en de ontvanger op de Arduino enerzijds en de kaku-devices anderzijds wilt vergroten dan kunt je de beide modules van een antenne voorzien. Je kunt dat relatief eenvoudig zelf doen. Op internet is te vinden hoe dat moet.

Bijlage 4, Aandachtspunten en links

- Wat kun je doen als je via je laptop/browser/app geen contact krijgt met je Arduino-server en je IP-configuratie gecontroleerd hebt en ook begrijpt? Onderzoek welke variant van het ethernet-shield je hebt. Er zijn namelijk, naast enkele incurante types, twee versies in omloop met de naam "Ethernet shield" en "Ethernet shield 2", wat meestal wel op het Arduino-bordje staat of op het doosje. Toen "Ethernet Shield" op de markt kwam was het nog niet bekend dat er ook een "Ethernet Shield 2" zou komen, vandaar de naamgeving. Bij gebruik van "Ethernet shield" (versie 1 dus) moet het contact tussen laptop en Arduino geen probleem zijn. Bij "Ethernet shield 2", is het aan te bevelen de bijbehorende library te installeren. Deze draagt ook de naam "Ethernet2" (wat wel goed gekozen is;). Meer informatie over dit nieuwe shield, waaronder ook de manier waarop je libraries installeert (eigenlijk gewoon kopieert) tref je aan door de onderstaande links te bekijken.
<http://www.arduino.org/products/shields/5-arduino-shields/arduino-ethernet-shield-2>
<http://labs.arduino.org/Arduino%20Ethernet%20Shield%202>
Nog enkele technische details
 - het nieuwe shield is gebaseerd op een nieuwe ethernet controller, de W5500-chip. Deze is



in de plaats gekomen van de W5100.

- In je (voorbeeld)code moet met de volgende aanpassing rekening houden:

```
#include <Ethernet.h> // Ethernet Shield 2? -> use <Ethernet2.h>
```

- De KaKu-schakelaars (APA3) kunnen, volgens opgave, 1500 W schakelen (ca. 6,5 A). Een flinke bouwlamp van 1000 W kan dus geschakeld worden, net als een koffiezetterapparaat van 1,5 kW, maar een waterkoker van 1800 Watt dus niet. Wees daar alert op. Lees de beschrijving achterop de schakelaars.

- Temperatuursensor

De zogenoemde “one-wire” temperatuursensoren, zoals de DS18B20, zijn populair op het Arduino-platform. Er zijn speciale libraries beschikbaar voor de sensor.

https://milesburton.com/Dallas_Temperature_Control_Library

<http://www.raywenderlich.com/38841/arduino-tutorial-temperature-sensor>

Er zijn ook (passieve of actieve) temperatuursensoren die gebaseerd zijn op een weerstandsverandering t.g.v. een temperatuurverandering, de zogenoemde NTC of PTC. Deze zijn relatief eenvoudig te interfaces, gewoon via een analoge ingang van de Arduino. Een nadeel van deze componenten is dat de weerstandsverandering (en ook de spanningsverandering) niet lineair verloopt met de temperatuurverandering. Er is een vrij complex verband met e-machten. Op het internet zijn deze lineariseringsformules wel te vinden, ook wel uitgeschreven in (Arduino) programmacode. Ook moeten dergelijks sensoren vaak gejikt worden. Als een exacte temperatuur niet nodig is, zijn het prima alternatieven.

- Lichtsensor

Het meten van licht, kan op veel verschillende manieren. De meest eenvoudige manier is die met een (passieve) lichtgevoelige of LDR. De weerstand van deze component is afhankelijk van de hoeveelheid opvallend licht. In een klein schakelingetje met nog een weerstand is het relatief eenvoudig de lichtverandering, via een weerstandsverandering om te zetten in een spanningsverandering. En die is weer meetbaar via één van de analoge ingangen van de Arduino. Een andere reden om een LDR te gebruiken is omdat deze (goedkope) component in veel van de Arduino-starterskits zit. Er zijn ook (actieve) componenten met complexere schakelingen en een eigen voeding. E.e.a. verondersteld meer kennis van elektronica.

- Android telefoon ontsluiten voor ontwikkelaars/programmeurs

Om (Xamarin-)app te ontwikkelen, debuggen en testen voor Android smartphone is het nodig op je telefoon te ontwikkellaarsopties in te schakelen. Vanaf Android 4.2 (JellyBean) is het ontwikkelmenu automatisch verborgen in de instellingen van je toestel. Om dit menu te “unlocken” ga je naar Instellingen >> Over toestel >> druk 7 keer op het buildnummer.

Vervolgens zal het menu ontwikkellaarsopties in je instellingen verschijnen. Doe het volgende om USB-foutopsporing in te schakelen: Instellingen >> Ontwikkellaarsopties >> USB-foutopsporing aanvinken.

Heb je een toestel met een lagere Android-versie dan kun je het ontwikkel-menu vinden onder Instellingen >> Ontwikkellaarsopties.

Na het aansluiten (via USB-kabeltje) van je telefoon op je PC kun je in Xamarin, via het startknopje, kiezen om i.p.v. voor een software-emulator direct te ontwikkelen voor jouw telefoon, die (automatisch?) in het lijstje komt te staan. Dit werkt op een vergelijkbare manier voor ook voor Android Studio.

- Android SDK

Bij het ontwikkelen van Android-apps moet je een keuze maken voor welke versie je wilt ontwikkelen en daarmee samenhangend voor de Android SDK. Waarschijnlijk verloopt dit, als onderdeel van de installatie, allemaal automatisch, maar soms is het nodig hier extra onderdelen aan toe te voegen, bijvoorbeeld i.v.m. de emulator die je gebruikt. Dit kan via Tools -> Android -> Android SDK Manager. Gelukkig geeft Xamarin (doorgaans?) een melding ("issue") wanneer er iets niet klopt en een advies om dit te "fixen".

- Xamarin, oriëntatie en layout

De designer van Xamarin, waarmee het gebruikersinterface van een app gemaakt wordt, kent een Design-mode en een Source-mode, met bijbehorende tabbladen. In de Source-mode is te zien dat het actuele design wordt beschreven in de file Main.axml. Het eenmaal ontworpen design wordt "gerenderd" naar een concreet design voor een Android-app. Eenvoudige apps zijn meestal gebaseerd op de doorgaans overzichtelijke "LinearLayout". De voorbeeld-app heeft dat ook, maar met daarbinnen echter nog een "RelativeLayout". Deze wordt aangewend om, over de hele lengte van het scherm, twee kolommen te genereren, met links (veelal) teksten of argumenten en rechts hun waardes.

Daarnaast is er nog de oriëntatie van de telefoon zelf. Als deze gedraaid wordt, bewust of vaker nog onbewust, wordt er in de code een nieuwe "Activity" gestart. Feitelijk start de app opnieuw met ook een her-initialisatie van de variabelen waardoor bijvoorbeeld een socketverbinding hersteld moet worden. In de programmacode kan dit uiteraard adequaat afgehandeld worden, maar soms, zoals in het gegeven voorbeeldprogramma, is het praktischer de layout vast in te stellen op "Portrait". De app zal daarmee niet reageren op draaiing van de telefoon. Indien dit ongewenst is, dient de toevoeging
"ScreenOrientation = Android.Content.PM.ScreenOrientation.Portrait"
uit de Activity-beschrijving van het voorbeeldprogramma gehaald te worden.
`[Activity(Label = ...)] // bovenin het programma`