

HBO-ICT & Ad ITSM

Embedded Systems

Opdrachten

Rolink, S. & Foppele, J.
9-19-2022



Embedded Systems Opdrachten

Inhoud

Week 1	2
Opdracht A – LED's en timing	2
Opdracht B – Seriële communicatie	4
Opdracht C – Port B	4
Week 2	5
Opdracht A – Digitale input	5
Opdracht B – Analoge input.....	5
Opdracht C – Sensoren	6
Week 3	7
Opdracht A – Servo	7
Opdracht B – Afstandssensor.....	7
Week 4	9
Opdracht A – Ethernet Shield	9
Opdracht B – I/O via website	10
Opdracht C – Http Request	10

Week 1

Beschrijving

De eerste week aan opdrachten. Om deze week compleet te hebben dienen opdrachten A en B te worden afgewerkt.

Doel

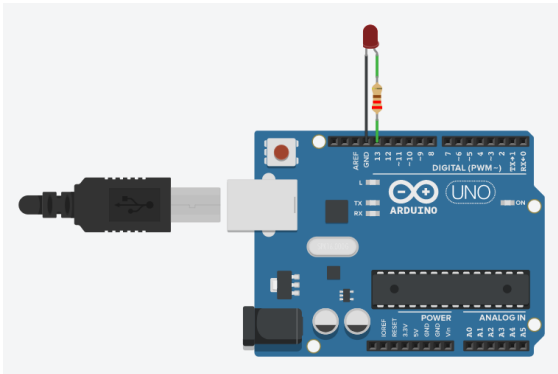
Het doel van deze opdracht is om te leren hoe de Arduino werkt en hoe je deze kunt aansturen met C code.

Opdracht A – LED's en timing

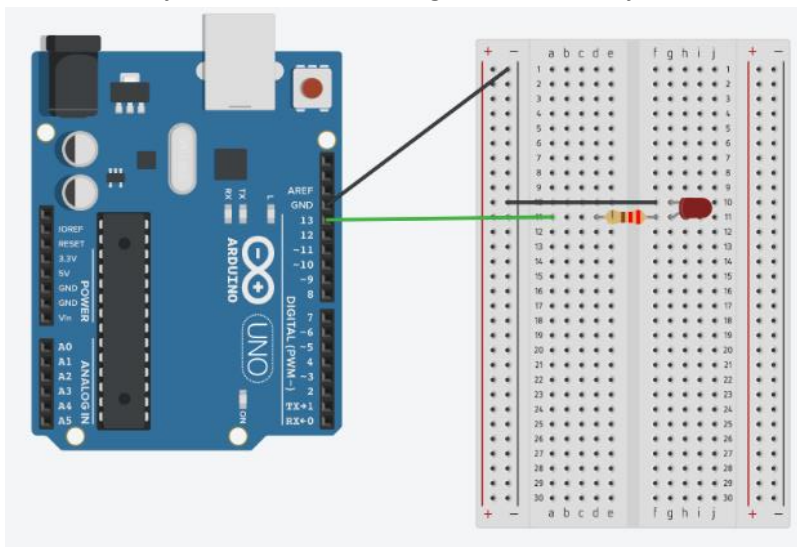
Loop de volgende stappen door:

1. Installeer de Arduino-IDE software. Deze is te vinden op www.arduino.cc, onder Software -> Downloads. Het is bij deze opdrachten ook mogelijk om de online Arduino Web Editor te gebruiken of simulatiesoftware zoals www.Tinkercad.com (let er daarbij op dat je je code als tekst maakt, en niet met codeblokken werkt).
2. **Veiligheid:** Zorg voor een veilige werkomgeving. Als je met een Arduino werkt kun je echt iets stuk maken als er iets verkeerd gaat. Niet alleen de Arduino, maar ook de apparatuur waar je de Arduino op aan sluit kan stuk als je kortsluiting maakt of een + en – verwisselt. Let dus goed op bij wat je doet, en houd de veiligheid altijd in het oog! Kijk ook goed naar de ondergrond die je gebruikt, leg de Arduino bijvoorbeeld niet op een metalen plaat. Ben je onzeker, simuleer dan eerst zonder Arduino wat je maakt.
3. Koppel je Arduino aan je Arduino-IDE, hierbij moet je onder Tools de juiste Arduino selecteren (waarschijnlijk de Uno) en de juiste poort instellen (waarschijnlijk iets met Serial, Arduino of COM in de naam). **LET OP:** ontkoppel je Arduino zodra je wat aan gaat sluiten zodat je de kans op kortsluiting verkleint.

4. Sluit een LED (light-emitting diode) en een weerstand aan op je Arduino zoals in onderstaand plaatje.

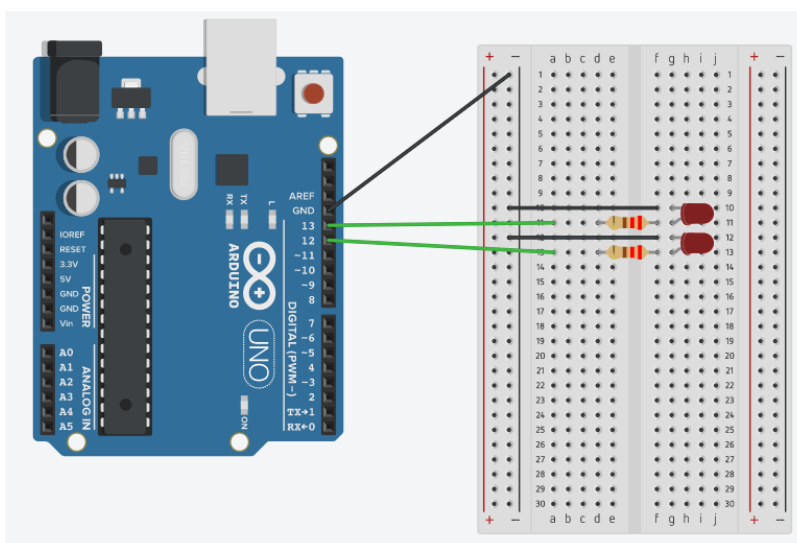


Je kunt hierbij ook een breadboard gebruiken zodat je niet hoeft te solderen.



Zorg er daarna voor dat je in de setup pin 13 als output definieert met de `pinMode()` functie. Laat daarna de LED knipperen door pin 13 afwisselend hoog en laag te maken door gebruik te maken van de `digitalWrite()` functie. Gebruik twee delays om de LED te laten knipperen.

5. Sluit een extra LED aan, en stuur deze aan met pin 12, bijvoorbeeld zoals in onderstaand schema.



Laat nu beide LED's gelijktijdig knipperen.

6. Pas je programma zo aan dat de LED's op verschillende snelheden knipperen. Krijg jij het voor elkaar om de ene LED 4x per seconde te laten knipperen en de andere LED 5x per seconde?
7. Behoud je setup zodat je deze in opgave C weer kunt gebruiken.

Opdracht B – Seriële communicatie

Loop de volgende stappen door:

1. We gaan nu de Arduino laten communiceren met je computer. Hiervoor gebruiken we de seriële poort, die we kunnen bekijken met de "Serial Monitor" die is ingebouwd in de Arduino-IDE. Stel deze in op 9600 baud (een stabiele maar langzame snelheid). Ook in de Arduino moet je deze snelheid gebruiken door in de setup `Serial.begin(9600);` uit te voeren. Zorg hierna dat er tijdens het starten van je Arduino eenmalig "Hello World!" verschijnt in je serial monitor door `Serial.println()` te gebruiken. Valt je op dat de Serial Monitor erg lijkt op de Console uit Visual Studio?
2. Stel nu de snelheid in op 115200 baud. Dit zorgt ervoor dat je meer characters in dezelfde tijd kunt doorsturen. Wat merk je van deze verandering in je programma?
3. Laat in het loop gedeelte van je code een getal iedere seconde optellen en stuur daarna dit getal via seriële communicatie naar je computer, zodat je kunt zien hoe lang je Arduino al draait.

Opdracht C – Port B

Loop de volgende stappen door:

1. Sluit nog 4 LED's op pinnen 8 tot 12. Bedenk hiervoor zelf hoe je dit aansluit.
2. Gebruik in je setup `DDRB = 0x3F;` om in een keer pinnen 8 t/m 13 op output te zetten, zodat je alle LED's kunt aansturen. Niet alleen scheelt dit regels code maar het is ook sneller omdat je nu in 1 instructie alle pins goed zet i.p.v. dat je de pins een voor een instelt.

0x3F is een getal, waarbij het 0x gedeelte aangeeft dat het een hexadecimaal getal is. Hexadecimale getallen hebben als grondtal 16, i.p.v. het standaard grondtal 10 (decimaal talstelsel) bij alledaagse getallen.

0x3F is gelijk aan 63 in het decimale talstelsel. Je kunt hieronder zien dat bij het hexadecimale getal de afzonderlijke bits veel makkelijker te identificeren zijn.

Dus:

`DDRB = 0x3F;`

`DDRB = 63;`

`DDRB = B00111111;`

zijn gelijk.

3	F
B00111111	

3. Op dezelfde manier kun je in een keer alle pins aan of uitzetten met de instructies `PORTB = 0x3F;` en `PORTB = 0;`. Probeer met deze statements je LED's te laten knipperen.
4. Maak nu een looplicht door eerst het getal 3 op PORTB te zetten en daarna iedere iteratie dit getal met 2 te vermenigvuldigen. Denk na over hoe je dit op een mooie manier overnieuw laat beginnen. **TIP:** als je een variabele buiten de setup en loop methoden maakt, dan is deze variabele overal beschikbaar.
5. Krijg jij het voor elkaar om het looplicht 2 kanten op te laten lopen (als een Cylon of Knight Rider looplicht)?

Week 2

Beschrijving

Week 2 staat geheel in het teken van complexere aansluitingen en meerdere functies in 1 systeem integreren.

Doel

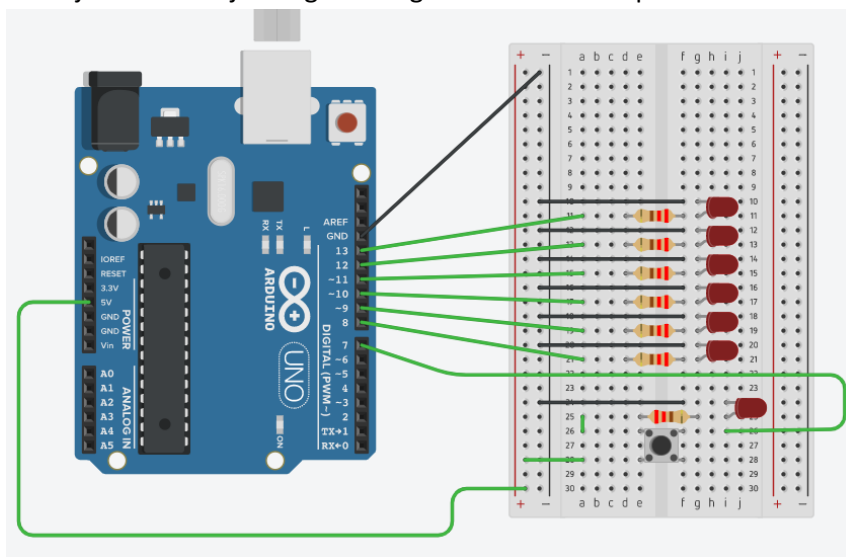
Het doel van deze opdracht is om te leren werken met:

- Digitale input
- Analoge input
- Simpele sensoren

Opdracht A – Digitale input

Voer de volgende opdrachten uit:

1. Breid je circuit dat je vorige week gemaakt hebt uit op onderstaande manier.



We gaan nu met de drukknop ervaren hoe je signalen van buiten de Arduino inleest. De LED bij de drukknop is bedoeld om visueel te testen of de drukknop ook daadwerkelijk werkt. De drukknop zit aangesloten op pin 7. Om hier iets mee te kunnen moet je deze pin ook instellen als INPUT, doe dit in de setup, door gebruik te maken van de `pinMode()` functie. Laat nu de status van de pin zien in de serial monitor door in de loop het resultaat van de functie `digitalRead(7)` met `Serial.println()` naar je computer te sturen. Zie je de status veranderen als je op de knop drukt?

2. Zorg ervoor dat je een looplicht hebt (zoals je vorige week in opgave A moest maken) en dat je met het indrukken van de knop je looplicht aanzienlijk sneller of langzamer laat lopen.

Opdracht B – Analoge input

Voer de volgende opdrachten uit:

1. Breid je schakeling uit met een potentiometer (kortweg potmeter). Hierbij moet je de buitenste twee pinnen aansluiten op de +5V en de GND en de middelste pin op de A0.
2. Bekijk of je de potmeter in de loop kunt uitlezen, door het resultaat van de functie `analogRead(A0)` te laten zien in de serial monitor.
3. Test wat er gebeurt als je de potmeter niet op de 5V maar op de 3.3V aansluit. Heb je hier een verklaring voor?
4. Laat nu het looplicht afhankelijk van de instelling van de potmeter sneller of langzamer gaan.

Opdracht C – Sensoren

Voer de volgende opdrachten uit:

1. Breid je circuit uit met een andere sensoren, bijvoorbeeld een temperatuursensor (tmp36) en een lichtgevoelige weerstand (LDR). Zoek zelf uit hoe deze aangesloten moeten worden (simuleer dit eerst in tinkercad zodat je geen hardware stuk maakt). **LET OP1:** De LDR heeft een zogenaamde pull-down weerstand nodig! **LET OP2:** De ultrasone afstandssensor is ook een sensor, maar hoeft nog niet. Deze komt volgende week terug in het huiswerk.
2. Test je sensoren door de waardes die er af komen in de serial monitor te laten zien.
3. Kun je ervoor zorgen dat de waardes begrijpelijk worden? De een temperatuur in graden celsius?
4. Bedenk een uitbreiding waarbij je de gemeten waarde van een sensor laat zien op de LEDs. Bijvoorbeeld door met een hogere waarde van je sensor meer LEDs te laten branden. **TIP:** Je kunt dit makkelijk testen met de potmeter.
5. **OPTIONEEL:** Zorg er nu voor dat je met een druk op de knop van sensor kunt wisselen. Je kunt nu al een mini-weerstation maken met een temperatuur en een lichtsensor. Mocht je de beschikking hebben over een LCD dan kun je je weerstation nog verder uitbreiden.

Week 3

Beschrijving

Week 3 staat geheel in het teken van de (micro) servo en de ultrasone afstandssensor.

Doel

Het doel van deze opdracht is om te leren werken met een:

- Servo
- Ultrasone Afstandssensor

Opdracht A – Servo

Voer de volgende opdrachten uit:

1. Sluit een micro servo-motor (kortweg servo) aan op een pwm pin van de arduino, bijvoorbeeld pin 9. Voor het gebruik van de servo heb je de servo software library van Arduino nodig. Deze kan als volgt gebruikt worden:

```
#include <Servo.h> // header file voor de servo-library
Servo servo1;      // nieuwe servo (met naam servo1)
servo1.attach(9);  // servo op pin 9
```

PAS OP: Een min-servo kan gevoed worden vanuit de Arduino, ook als de deze op haar beurt weer gevoed wordt vanuit de USB van je computer. Gebruik echter in deze configuratie geen zwaardere of grotere servo's. Deze nemen aanzienlijk meer stroom (en vermogen) waardoor er problemen kunnen ontstaan met (de voeding vanuit) je laptop.

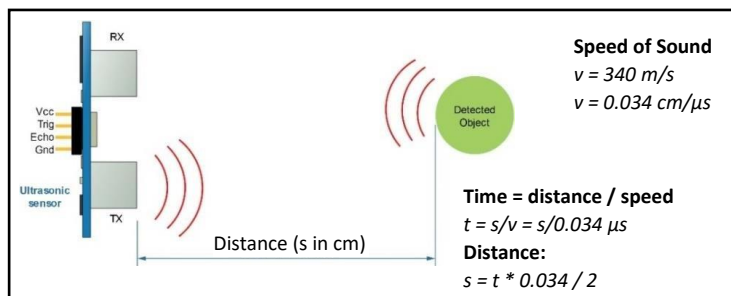
2. Schrijf een programma waarmee je de stand van (de rotatie-as van) de servomotor kunt laten variëren met behulp van een sensor naar keuze. Je kunt een krachtsensor gebruiken, maar een sensor die een ander grootte meet, zoals licht of temperatuur, mag ook. Zorg dat je de schalen van sensor en servo op elkaar afstemt, waarbij een minimaal niveau (geen kracht/licht) leidt tot geen uitslag en een maximaal niveau (maximaal kracht/licht) tot een maximale uitslag.

Onderzoek hierbij het gebruik van de functie `map()`.

Opdracht B – Afstandssensor

Voer de volgende opdrachten uit:

1. De ultrasone afstandssensor bepaalt een afstand tussen de sensor en een object op basis van geluidsgolven. Gegeven is de afbeelding hiernaast die schematisch weergeeft hoe de berekening van de afstand tot stand komt.



Speed of Sound
 $v = 340 \text{ m/s}$
 $v = 0.034 \text{ cm}/\mu\text{s}$

Time = distance / speed
 $t = s/v = s/0.034 \mu\text{s}$

Distance:
 $s = t * 0.034 / 2$

 - a. Verklaar waar het getal 0.034 vandaan komt.
 - b. Waarom zit in de afstandsberekening de deling door 2?
 - c. Wat is de invloed van de omgevingstemperatuur op de berekening?
2. Sluit de afstandssensor aan op je Arduino. Kijk daarbij goed of je een variant met 3 pinnen of een variant met 4 pinnen hebt. In beide gevallen moet de sensor aangesloten worden op de +5V en GND pinnen, en de overige pinnen op de digitale input/output pinnen. Beide configuraties werken in de basis hetzelfde. Je moet eerst een puls geven aan de ultrasone sensor (door de pin bijvoorbeeld 2 tot 5 microseconden hoog te maken met behulp van

`delayMicroseconds(2);`) en daarna de puls te meten. De beste pulslengte kan variëren, experimenteer hiermee totdat je de voor jou meest betrouwbare pulslengte hebt gevonden.

Doe dit laatste door gebruik te maken van:

```
long duration = pulseIn(7, HIGH);
```

Hierna kun je de duration omrekenen naar een afstand in centimeters en in de serial monitor laten zien.

TIP: om de afstandssensor voldoende tijd te geven voor een meting, is het verstandig niet te vaak te samplen per seconde. Aanbevolen wordt om een delay van 250 milliseconde in te bouwen voordat je een nieuwe meting uitvoert.

3. Zorg er nu voor dat je de meting van de afstandssensor in een aparte methode zet, zodat je loop wat overzichtelijker wordt.
4. **OPTIONEEL:** Zorg ervoor dat de servo uit opdracht A draait als een object dichterbij komt (of juist verder weggaat).

Week 4

Beschrijving

Week 4 staat geheel in het teken van het Ethernetshield.

Doel

Het doel van deze opdracht is om te leren werken met een:

- Ethernetshield
- Server

Opdracht A – Ethernet Shield

Voer de volgende opdrachten uit:

1. Zet de ethernet shield op je Arduino en laad de file *WebServerIO.ino* in (deze kun je vinden op Blackboard). Neem hierbij wel eerst de volgende zaken aandachtig door voordat je de Arduino opstart:
 - Let bij het plaatsen van je shield op dat de usb poort van de Arduino geen contact (= sluiting) maakt met je Ethernet Shield, dit kan bij sommige shields voorkomen. Denk er in dat geval over na om bijv. afstand headers (Shield Stacking Headers) aan te schaffen.
 - De Arduino moet aangesloten worden op de usb poort, maar nu ook via een ethernetkabel (waarschijnlijk een UTP cat5 kabel). Je kunt deze direct aan je computer verbinden, of dit doen via een router. Als je computer/laptop naast deze bedrade verbinding nog over de mogelijkheid beschikt van een wifi-connectie, dan kun je die gebruiken om met het internet te verbinden. Dat kan handig zijn om bijvoorbeeld op Blackboard te kunnen.
 - De verbinding tussen Arduino en computer werkt met IP-adressen. Het is hierbij handig om gebruik te maken van IP-adressen in de range 192.168.1.x, waarbij x verschilt per device. Het voordeel van deze range is dat het strikt lokaal is. De informatie van computers in deze range worden niet doorgegeven over het internet.
 - Kijk naar statische en dynamische IP-adressen en het al dan niet gebruiken van een DHCP-server. Zorg ervoor dat je instellingen gebruikt zodat je Arduino en computer met elkaar kunnen verbinden. Voor de rest van de opgave gaan we er van uit dat je laptop/computer gebruik maakt van IP 192.168.1.2 en de Arduino van 192.168.1.3.
 - Je Ethernet Shield heeft naast een IP-adres ook een MAC-adres, welke is gekoppeld aan de hardware. Neem jouw MAC-adres van de Ethernet Shield over in de software. Wordt je Ethernet Shield zonder sticker geleverd, verzin dan zelf een MAC-adres (deze bestaat uit 6 cijfers, bijv.: 0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF).

Probeer nu de voorgeprogrammeerde website zichtbaar te maken door in je browser de volgende URL in te voeren: <http://192.168.1.3>
Als alles juist is gegaan dan zie je de website die hiernaast is afgebeeld.

Embedded I/O-Webserver

Week 4 Opdracht A van het vak embedded systems
Dit voorbeeld is gebaseerd op het script op [deze site](#)
De website is dynamisch gemaakt door sensorwaarden van kanaal 0 toe te voegen.
Breid het programma uit met de mogelijkheid om variabelen mee te geven.
Dit kan o.a. door GET-variabelen, via de URL (192.168.1.3/?p8=1).
Gebruik de functie `parseHeader(httpHeader, arg, val)`

Analog sensor, channel 0: ?

No IO-pins to control

2. Gekoppeld aan het IP-adres is ook een poort. Deze poort staat in de code standaard ingesteld op 80. Lukt het jou om de poort te veranderen naar een ander nummer? Kies bijvoorbeeld poort 25565. Om nu je site te benaderen moet je de poort ook opgeven in je browser (<http://192.168.1.3:25565>). Zet de poort daarna voor je eigen gemak weer op de default poort 80.

3. Lukt het je om de site visueel aan te passen? Je kunt wellicht een lettertype of een andere kleurstelling kiezen die beter bij jou past.

Opdracht B – I/O via website

Voer de volgende opdrachten uit:

1. Sluit een sensor of een potmeter aan op analoge poort 0 en bekijk wat het effect hiervan is op jouw website. Kun je het voor elkaar krijgen om de sensorwaarde rood te maken als deze boven een bepaalde waarde komt?
2. In de gegeven code worden zogenaamde GET-variabelen al opgevangen en doorgestuurd naar de functie `parseHeader()`. Pas deze functie zo aan dat je pin 8 en pin 9 kunt besturen met de volgende voorbeeldaanroepen:

Voorbeeldaanroep	Effect
http://192.168.1.3/?p8=1	Pin 8 wordt hoog
http://192.168.1.3/?p9=0	Pin 9 wordt laag

Je kunt dit testen door bijvoorbeeld LEDs te koppelen aan pinnen 8 en 9.

Merk op dat de pinnen 10 t/m 13 in gebruik zijn door het Ethernet Shield en daardoor niet beschikbaar zijn om aan te sturen.

3. Breid je programma verder uit zodat je ook een feedback krijgt op je website wanneer een pin is gewijzigd (of wanneer dit is mislukt).
4. Maak op de website knopjes om pinnen 8 en 9 aan en uit te kunnen zetten.
5. **OPTIONEEL:** plaats een SD-kaart in je Ethernet Shield en laadt de webpagina van de SD-card.

Opdracht C – Http Request

Voer de volgende opdrachten uit:

1. Bekijk de code op <https://www.arduino.cc/en/Tutorial/LibraryExamples/WebClient> en laad de voorbeeldcode in je Arduino. Stel de juiste gegevens (MAC, IP, DNS) in. Je ziet dat je er hierbij voor moet zorgen dat de Arduino bij het internet kan. Dit kun je voor elkaar krijgen met een router waarvan je de DNS kunt gebruiken.
2. Bestudeer de code, zie dat deze code ipv een website hosten een connectie maakt met een website (in dit geval www.google.com) en daar informatie vandaan haalt. Bekijk via de Serial Monitor of dit ook lukt.
3. **OPTIONEEL:** verbind met een andere site dan google, bijvoorbeeld met een weersite zoals openweathermap.org of een nieuwssite zoals newsapi.org. Toon de binnengehaalde informatie, bijvoorbeeld op een display of in de Serial Monitor.