Opdracht “Infrarood”

# Inleveren

Maak de onderstaande opdrachten **individueel** (behalve Opdracht 11, die mag je samen met je practicumpartner doen) en lever het volgende in:

* Een verzorgd document met de antwoorden op alle vragen, en korte rapportjes van je onderzoeken (zet ruwe meetdata in een bijlage)
* De code die je geschreven hebt in een ZIP-bestand (waarbij ook steeds duidelijk is welke code bij welke opdracht hoort).
* Een aansluitschema (Fritzing-tekening).
* Een softwareontwerp voor alle opgaves vanaf Opdracht 7.

Bedenk bij elke opdracht van tevoren welk resultaat je verwacht. Als je waardes ziet die afwijken van je verwachting, probeer dan te achterhalen waar dit door komt en beschrijf dit in je verslag.

# Beoordeling

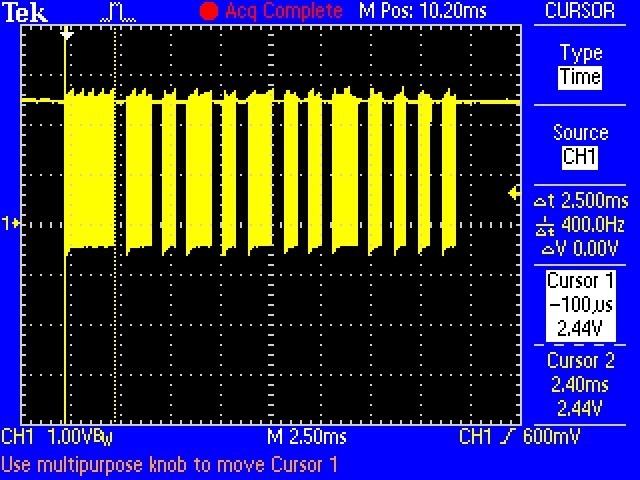
Je krijgt een voldoende voor de opdracht als de onderstaande vijf punten elk een voldoende opleveren:

1. De kwaliteit van de antwoorden op de vragen: zijn de conclusies die je trekt helder en worden ze onderbouwd door metingen en/of bronnen?
2. De kwaliteit van je software: nette en leesbare code (betekenisvolle namen voor variabelen en functies, commentaar waar nodig, goede verdeling over .ino, .c en .h-files, et cetera) en waar nodig voorzien van commentaar.
3. De aanwezigheid van een correct en duidelijk aansluitschema.
4. De kwaliteit van je software-ontwerpen. Relevant is hier dat zichtbaar is dat je van tevoren hebt nagedacht over de structuur (verdeling van code over verschillende bestanden) en de flow van je code. Lever ook state-diagrammen bij waar relevant (in elk geval bij Opdracht 11).
5. Je kunt een werkende demonstratie geven van het correct functioneren van het resultaat van Opdracht 11.

# Hoe werkt een infrarood-afstandsbediening?

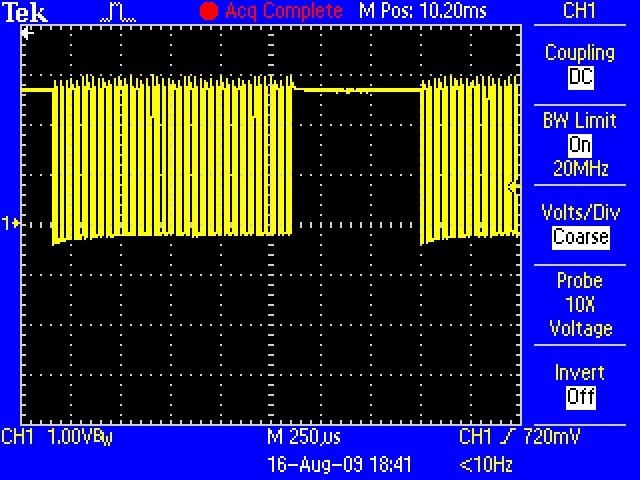
Op het moment dat je op een knop op de afstandsbediening van je tv drukt, zendt deze via een infrarood-led een signaal uit. Als je tv dit signaal opvangt, wordt dit geïnterpreteerd door de software op je tv, en zal er (mogelijk) iets gebeuren: het volume gaat omhoog, de volgende zender wordt opgezocht, er wordt gewisseld tussen HDMI-ingangen, etc.

Een afstandsbediening zendt niet gewoon maar wat licht uit, maar zendt een serie pulsjes uit die voldoen aan een bepaald protocol. Als je het ruwe signaal opvangt, kan dit er als volgt uitzien[[1]](#footnote-1):

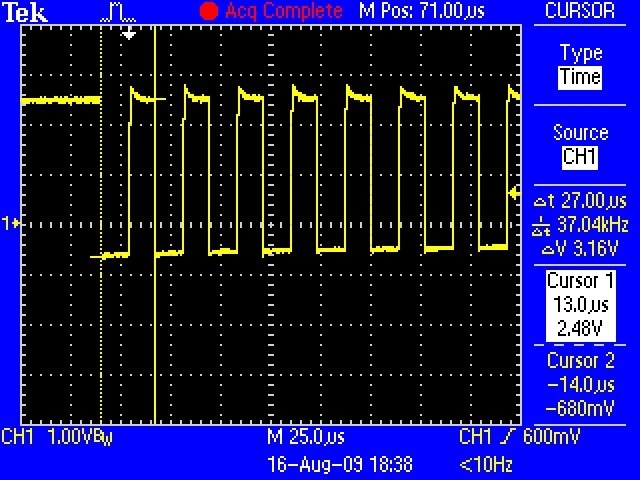


Je ziet dat het licht steeds even aan is, dan even uit, dan weer aan, et cetera. Er wordt dus een pulserend signaal gegeven. Zoals je ziet, duren niet alle pulsen even lang en zijn ook de “pauzes” tussen de pulsen niet allemaal even lang. De lengtes van de pulsen en de pauzes ertussen bevatten een codering die door de tv wordt opgevangen.

Als je verder zou inzoomen op het signaal, ontdek je dat het nog wat gecompliceerder ligt:



En nog wat verder ingezoomd:



De pulsen blijken zelf dus ook weer uit kleine pulsen te bestaan! Deze pulsen zijn wel allemaal even lang: de led blijkt te knipperen met een frequentie van enkele tienduizenden hertz (in dit voorbeeld is het zo’n 38 kHz). Dit heeft een aantal redenen. De twee belangrijkste: je bespaart energie (het ledje brandt immers maar zo’n 50% van de tijd), en de tv hoeft alleen maar te letten op signalen van rond deze frequentie.

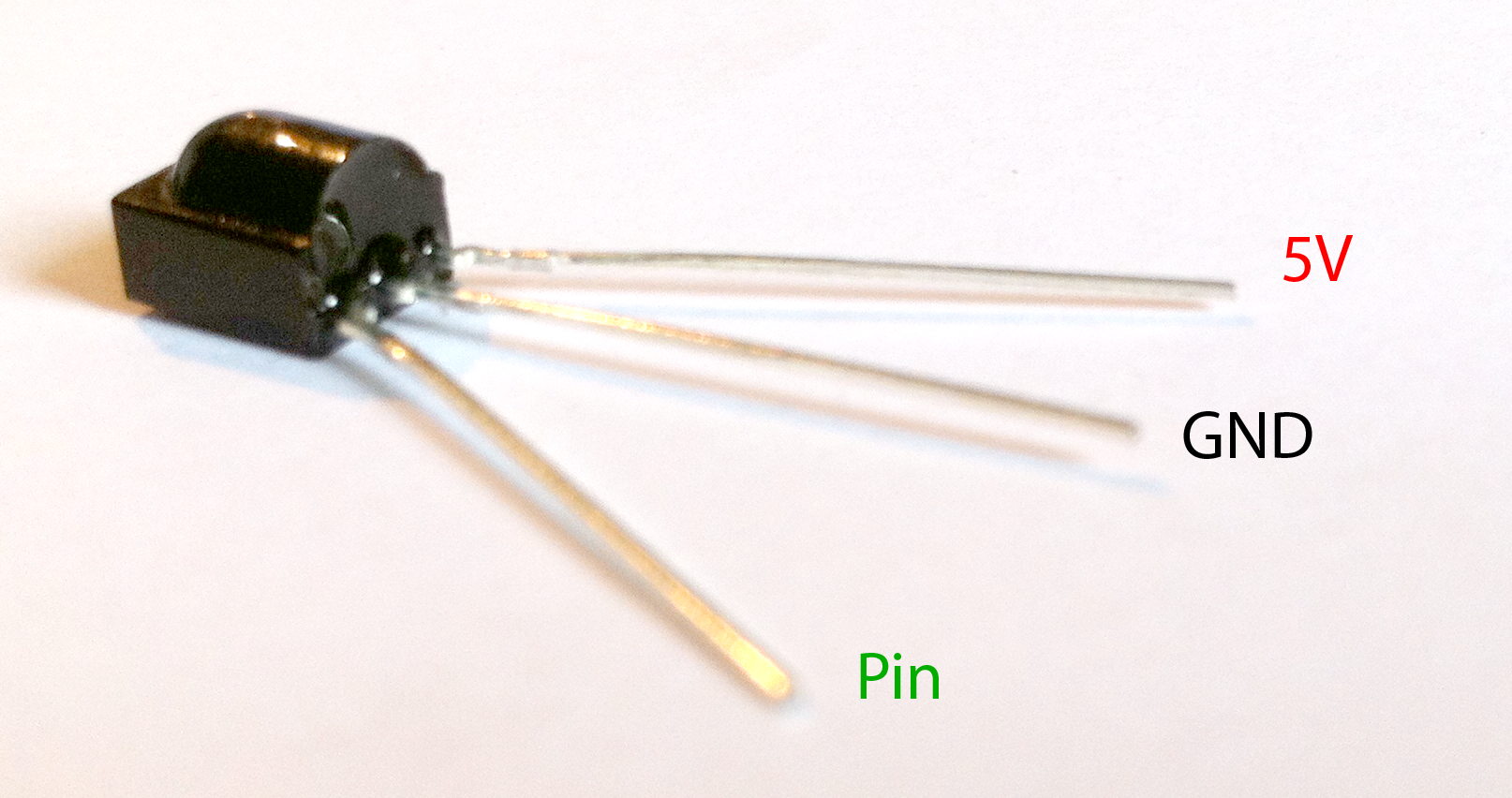
De sensoren die we gebruiken negeren de hoge frequentie en geven alleen de frequentie door die de codering van het signaal bevat.

1. Algemene vragen

Gebruik bronnen bij het beantwoorden van onderstaande vragen (ga dus niet zelf maar wat verzinnen) en refereer naar deze bronnen volgens de APA-normen.

1. Hoe kan het dat jij infrarood licht niet kunt zien, terwijl de tv dat wel kan?
2. Zijn er dieren die infraroodlicht kunnen zien? Worden zij verblind als ze door je woonkamer lopen terwijl je de tv bedient?
3. Welke “huis, tuin en keuken”-manieren zijn er te kunnen zien of je infraroodled wel echt licht geeft?
4. Er is in huis meer dan alleen je afstandsbediening die IR-licht uitstraalt. Wat nog meer bijvoorbeeld? En hoe kan het dat je tv daar niet af en toe per ongeluk op reageert?
5. Hoe kan je tv onderscheid maken tussen verschillende knoppen?
6. Hoe kan het dat je tv (over het algemeen) niet reageert op de afstandsbediening van je dvd-speler of die van de tv van je buurvrouw?
7. Hoe kan het dat sommige knoppen als ze ingedrukt worden steeds effect blijven houden (zoals de volumeknoppen) en andere niet (bijvoorbeeld de “2”-knop)?
8. Aansluiten en testen

Sluit de IR-sensor aan (zie Afbeelding 1) en voeg een ledje (met voor jou zichtbaar licht) toe. Schrijf een testprogramma dat het ledje aan laat gaan zodra er een IR-signaal binnenkomt, en weer uit zodra er (na een korte wachttijd) weer een IR-signaal binnenkomt. Als je klaar bent, kun je het ledje met een willekeurige druk op de afstandsbediening toggelen.



Afbeelding 1 Aansluiten IR-sensor

## Vragen

* Geeft de sensor “HIGH” bij het ontvangen van een signaal of juist bij afwezigheid ervan?
* Wat is de minimale wachttijd die je nodig hebt om te voorkomen dat het ledje gaat knipperen? Wat betekent dat?

1. Code opvangen met digitalRead

Schrijf een (nieuw) programma dat in de loop steeds een digitalRead op de IR-poort uitvoert. Laat het programma de lengte van een puls meten en de lengte naar de serial schrijven zodra de puls afgelopen is. Doe dan hetzelfde voor de afwezigheid van de puls, en blijf dit herhalen. Je krijgt dan dus een patroon te zien van hoe lang de HIGH heeft geduurd, hoe lang de LOW heeft geduurd, etc. Doe je metingen in microseconden, want de pulsjes duren heel kort.

Test je programma door een aantal keer dezelfde knop op de afstandsbediening te gebruiken en bekijk of het in de Serial geprinte resultaat steeds (ongeveer) hetzelfde is.

1. Onderzoek hoe lang een print naar Serial duurt

Schrijf een (nieuw) programma waarmee je, onafhankelijk van de IR-opdracht, meet hoe lang het eigenlijk gemiddeld duurt om een print naar Serial te doen. Let op: het “opstarten” van de Serial (*Serial.begin()*) kost wat tijd. Geef je programma die tijd (met een delay van bijvoorbeeld 250ms in de setup) en start daarna pas je metingen.

Herhaal (ook bij de volgende opgaven) alle metingen een aantal keer: noteer de gemiddelden en de standaarddeviatie, en doe je metingen niet in milliseconden, maar in microseconden. Houd er ook rekening mee dat een actie soms zo kort duurt dat het weinig nut heeft om een enkele uitvoer te meten: herhaal de actie dan bijvoorbeeld 1000 keer, en deel het resultaat door 1000. Doe dit meerdere keren om de standaarddeviatie te kunnen bepalen.

Beantwoord de volgende vragen:

* Maakt het uit hoeveel tekst je naar Serial stuurt?
* Kost het schrijven van een newline extra tijd?
* Maakt de baudrate uit?
* Maakt het uit of je Serial venster open staat (hint: om de metingen te kunnen inzien kun je het EEPROM-geheugen en een extra programma gebruiken)?
* Hierboven staat dat het opstarten van de Serial tijd kost. Is dat eigenlijk wel waar, en hoeveel tijd is dat dan?

1. Introduceer buffer met malloc en realloc

Het zou kunnen dat je metingen beinvloed worden door het steeds printen naar de Serial. Om dit uit te sluiten, gaan we pas printen zodra er een bepaalde tijd (zie Opdracht 2:) geen wijziging meer is ontvangen in de puls: de code is dan afgelopen. Uiteraard moeten de metingen wel opgeslagen worden. Doe dit in een dynamisch (niet-circulair) buffer waarbij je steeds als er een waarde wordt toegevoegd, de grootte van de buffer aanpast. Zodra de code is afgelopen en de waarden zijn geprint, verklein je de buffer weer naar 0 bytes. Schrijf een nieuw programma dat dit doet. Je mag de aan-/uit-tijden gewoon achter elkaar in 1 int-buffer zetten. Je krijgt dan dus weer een patroon van HIGH- en LOW-waarden na elkaar, maar kunt in de buffer niet zien welke LOW en HIGH worden (omdat een signaal altijd met een HIGH of LOW begint, kun je dat als het goed is echter wel afleiden als je wil).

Wat gebeurt er als je heel snel achter elkaar op een knop van de afstandsbediening blijft drukken?

1. Onderzoek hoe lang malloc, realloc en free duren

Schrijf een nieuw programma om te meten hoe lang een malloc, realloc en free duren. Beantwoord daarbij in elk geval de volgende vragen en noteer de waarden (uiteraard moet je alle metingen een aantal keer herhalen: noteer de gemiddelden en de spreiding):

* Hoe lang duurt een malloc? Maakt het uit hoeveel geheugen je alloceert?
* Hoe lang duurt een realloc? Maakt het uit hoeveel geheugen er al gereserveerd was? Maakt het uit hoeveel geheugen je extra alloceert? En is er verschil tussen het vergroten of juist verkleinen van de gealloceerde geheugenruimte?
* In de slides werd gezegd dat calloc langer duurt dan malloc. Is dat eigenlijk wel waar?
* Is “handmatig” initialiseren van de waarden na een malloc sneller of langzamer dan een calloc?
* Maakt het voor een free uit hoeveel geheugen er gealloceerd was?

1. Gebruik een circulair buffer

Ook realloc en malloc kunnen de metingen beïnvloed hebben. Om dat uit te sluiten, gaan we gebruik maken van een circulair buffer dat realtime niet aangepast hoeft te worden. Doe hetzelfde als bij Opdracht 5:, maar maak gebruik van een circulair buffer waarvan de grootte compile-time bepaald is. Schrijf hiervoor een nieuw programma. Treedt er nog steeds een probleem op? Hoe groot moet je buffer minimaal zijn om geen gegevens kwijt te raken? Sla wederom zowel LOW- als HIGH-waarden op, achter elkaar, in microseconden.

1. Maak gebruik van interrupts

Misschien is digitalRead wel het probleem. Schrijf een nieuw programma gebaseerd op opgave 7, waarbij je geen digitalRead meer gebruikt, maar interrupts (op zowel wijziging naar “hoog” als naar “laag”).

Let op: omdat Serial ook gebruik maakt van interrupts, kun je problemen krijgen (deadlocks) als je de Serial gebruikt in je eigen interruptfunctie. Laat het testen of er voldoende tijd verstreken is en het printen van de buffer dus buiten de interruptfunctie gebeuren.

1. Lees de ruwe pin-data

Een andere manier om om digitalRead heen te werken, is het uitlezen van de ruwe pin-data. DigitalRead doet dat ook, maar bevat wat overhead die je nu niet kunt gebruiken.

Lees de toelichting op <http://www.arduino.cc/en/Reference/PortManipulation>

Gebruik de volgende code, waarbij je moet zorgen dat IRpin\_PIN de poort bevat, en IRpin het (hardwarematige) poortnummer zoals je dat kunt vinden op <http://arduino.cc/en/Hacking/PinMapping168>. Gebruik je digitale pin 2 van je Arduino Uno, dan zijn de waarden respectievelijk PIND en 2. Let op: Het gebruik van digitale pinnen 0 en 1 is over het algemeen geen goed idee, maar als je ze op deze manier aanspreekt, is het zeker onverstandig.

In de setup:

DDRB = 0xFF;

DDRD &= ~(1 << IRpin);

Elders in je programma (geeft boolean terug):

(IRpin\_PIN & (1 << IRpin))

Schrijf een nieuw programma, gebaseerd op je code bij Opdracht 7:, waarbij je bovenstaande constructie gebruikt.

1. Vertraagde afstandsbediening

Sluit een infrarood-led en een knop aan op je Arduino en breid je programma zo uit (maak eerst een kopie) dat het in de buffer opgeslagen signaal op de IR-led wordt afgespeeld op het moment dat je op de knop drukt.

Test het programma door een signaal te laten afspelen vlakbij het apparaat waarbij de afstandsbediening hoort (tv, dvd-speler). Als het werkt, weet je zeker dat je het goed hebt gedaan. Werkt het niet, dan kan er iets mis zijn met je programma, maar het kan ook zo zijn dat de golflengte van de IR-led niet dezelfde is van je afstandsbediening en dat de ontvanger in je tv gevoelig is. Je kunt het dan nog met een andere apparaat proberen.

Alternatief is dat je het opgeslagen signaal met een factor 10 of 100 vertraagd laat afspelen op een zichtbare led. Dat is minder leuk, maar geeft wel een mooie lichtshow ;-)

1. Lampen schakelen met je afstandsbediening

*Deze laatste opdracht mag je samen met je practicumpartner doen (zorg wel dat je allebei het software-ontwerp en de software inlevert!). Ook de verplichte demonstratie mag je samen doen. Je mag zelf kiezen van wie je de IR-code als basis gebruikt.*

Sluit (minimaal) drie ledlampjes aan op je Arduino en zorg ervoor dat je die kunt aan- en uitzetten met de knoppen 1, 2 en 3 op je afstandsbediening. Daarnaast moet er een knop zijn waarmee je ineens alle ledjes kunt aan- en uitzetten.

Tips om het onderscheid tussen de knoppen te maken, vind je in de Powerpoint “IR-signaal lezen” op #OnderwijsOnline.

1. Afbeeldingen van signalen zijn afkomstig van Adafruit.com. [↑](#footnote-ref-1)