

Internet over licht

Hoe kan dataoverdracht over laserlicht op korte afstand een optimale snelheid bereiken?

J. Baas

S. van der Klugt

Onder begeleiding van Chris van den Berg

NT & NG Informatica

Metis Montessori Lyceum Amsterdam (6v2)

Abstract

Van een dataverbinding tussen een laser en een photodiode¹, beide aangesloten aan Arduino's² wordt onderzocht hoe de optimale overdrachtssnelheid kan worden bereikt. Hiervoor worden een aantal variabelen onderzocht. Ten eerste de frequentie waarmee bits³ worden verstuurd. Ten tweede de lengte van het deel van de data wat wordt verstuurd per keer (ook wel frame of pakket) en specifiek de lengte van de nuttige data hierin (payload). Ten derde de tijd waarin niks wordt verstuurd tussen frames (IFG). Ten vierde de lengte van het beginsignaal van een frame (preamble). En ten slotte de fysieke afstand tussen de zender en ontvanger. Uit de resultaten blijkt dat de optimale ingestelde frequentie ongeveer 1250 b/s is, de ideale lengte van een payload is ongeveer 384 bits, de IFG heeft een optimale lengte van 24 bits, de preamble moet minimaal 8 bits zijn, en de resultaten voor afstand geven geen duidelijk genoeg verband aan om conclusies uit te trekken. Deze variabelen samen leiden tot een maximale werkelijke snelheid van 450 b/s.

Inhoudsopgave

Abstract	2
Inhoudsopgave	3
Inleiding	4
Theorie	5
Manchester encoding	5
Frames	6
Foutdetectie en -correctie	7
Experiment	8
Benodigdheden	8
Methode - Programmeren	8
Overzicht dataflow	9
Methode - Experiment	9
Vorbereiding	9
Metingen	11
Resultaten	13
Symbolen bij de grafieken	13
Grafieken	14
Conclusie	19
Discussie	19
Aanbevelingen	20
Bronnenlijst	20

¹ Een ontvanger module die te gebruiken is in combinatie met een arduino om lichtsignalen om te zetten in bits.

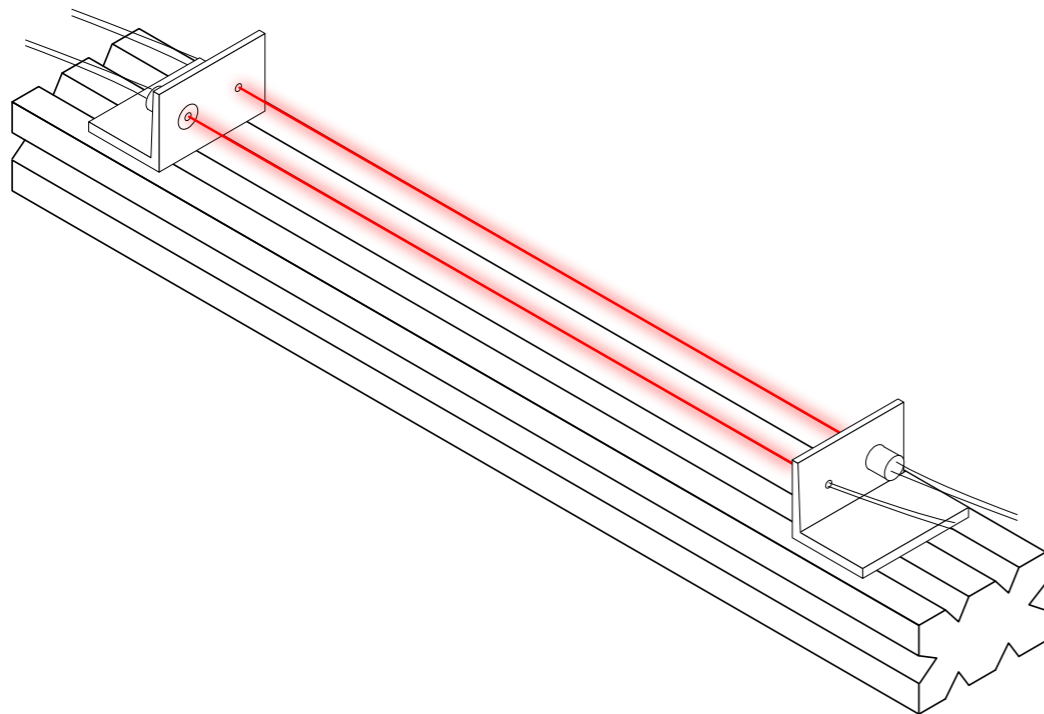
² Microcontroller die geprogrammeerd kan worden om elektronische onderdelen aan te sturen.

³ Een bit is de kleinste eenheid van data op een computer, een bit de waarde 0 of 1 hebben.

Inleiding

Er wordt tegenwoordig steeds meer van het internet gebruik gemaakt, steeds meer apparaten verzenden en ontvangen steeds meer data. Bij deze communicatie wordt vaak gebruik gemaakt van WiFi of 4G, deze hebben echter enkele nadelen. Zo raakt het electromagnetische spectrum vol, in 2020 zijn er voor voor meer dan 1,2 miljard euro frequenties geveild in Nederland (Agentschap Telecom, 2020). Daarom wordt er gezocht naar andere oplossingen. Bijvoorbeeld LiFi, dat is draadloos internet over licht. De voordelen hiervan zijn onder andere hoge mogelijke snelheden, meer beschikbare bandbreedte (omdat licht niet door objecten heen gaat is er minder interferentie) en mogelijkheid voor gebruik op plekken waar elektromagnetische interferentie een probleem oplevert (vliegtuigen en ziekenhuizen).

In dit onderzoek wordt onderzocht hoe dataoverdracht over laserlicht op korte afstand een optimale snelheid kan bereiken. Hierbij wordt gebruik gemaakt van onderdelen die beschikbaar zijn voor consumenten. Ten eerste wordt een aantal veel gebruikte protocollen en concepten uitgelegd in het [theoretisch deel](#). Daarna worden de [benodigdheden](#) om het onderzoek te herhalen opgesomd. Vervolgens wordt de methode van het onderzoek toegelicht in twee delen, hoe de [programma's zijn geschreven](#) en [de stappen om het experiment zelf uit te voeren](#). Hierop volgend zijn de [resultaten](#) gevisualiseerd in grafieken en wordt uitgelegd hoe dat is gedaan. Ten slotte komen de [conclusie](#) en de [discussie](#), waarin de resultaten worden geïnterpreteerd en er wordt geëvalueerd wat de tekortkomingen zijn van dit onderzoek en hoe vervolgonderzoek er uit zou moeten zien.



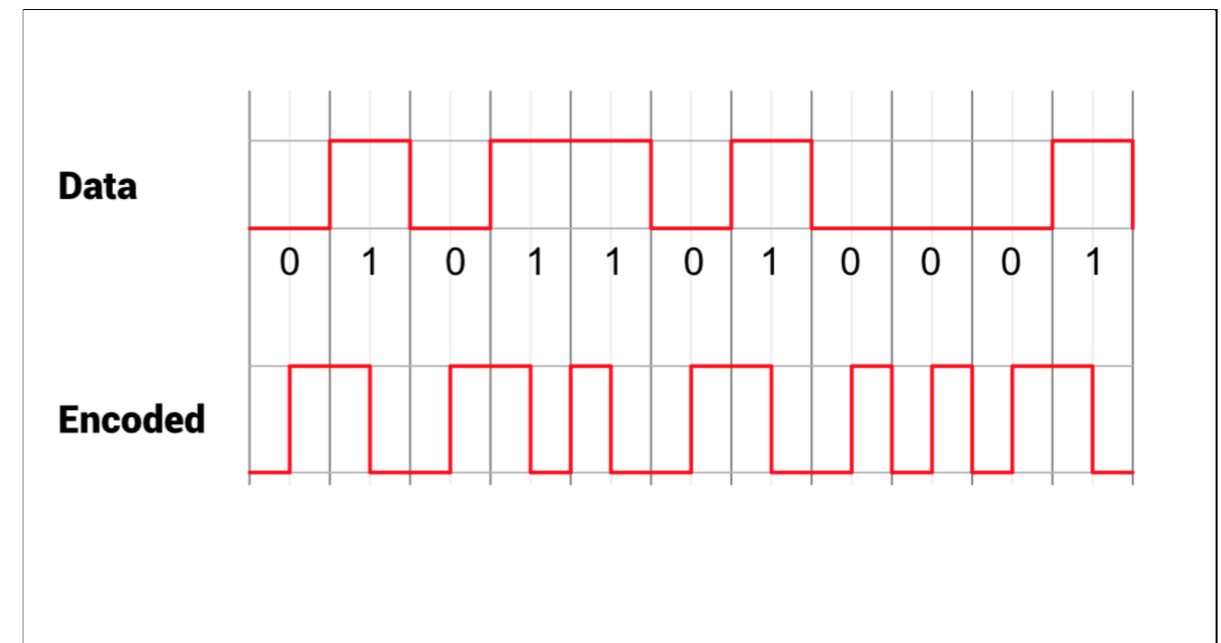
Theorie

Om data goed over te dragen moeten er tussen computers afspraken worden gemaakt over hoe dat gebeurt. Dit worden protocollen genoemd. In dit hoofdstuk worden enkele veel gebruikte protocollen uitgelegd. Dit zijn onder andere protocollen voor het opdelen van data in pakketten en voor het opsporen en corrigeren van fouten. Ook moet de goede hardware worden gekozen, zodat het signaal zo goed mogelijk kan worden overgebracht. Hierin zijn twee belangrijke opties te onderscheiden, namelijk LED's of lasers.

Protocollen

Protocollen zijn afspraken over hoe communicatie verloopt. Ze kunnen worden ingedeeld op laag in het OSI model, ontwikkeld door de International Organization for Standardization. Dat deelt het internet op in zeven lagen (Cloudflare, z.d.). Deze worden van beneden naar boven steeds abstracter: de onderste laag is de fysieke laag, de verbinding tussen verschillende hardware, de bovenste laag is de applicatielaag, over hoe de informatie aan de gebruiker wordt getoond. In dit onderzoek wordt er vooral naar laag 1 en 2 gekeken, waar laag 1 de fysieke transmissie van data is en laag 2 de data in frames⁴ opdeelt en fouten corrigeert. Laag 3 tot en met 5 zijn nodig bij netwerken van meer dan twee computers, en regelen dingen als de route van de data. Laag 6 en 7 komen ook terug in het onderzoek, deze zijn echter niet het onderwerp maar dienen alleen voor het compleet maken van de proof-of-concept. Laag 6 zorgt voor bijvoorbeeld het omzetten van de bits naar letters, en laag 7 gaat over hoe de data aan de gebruiker wordt getoond.

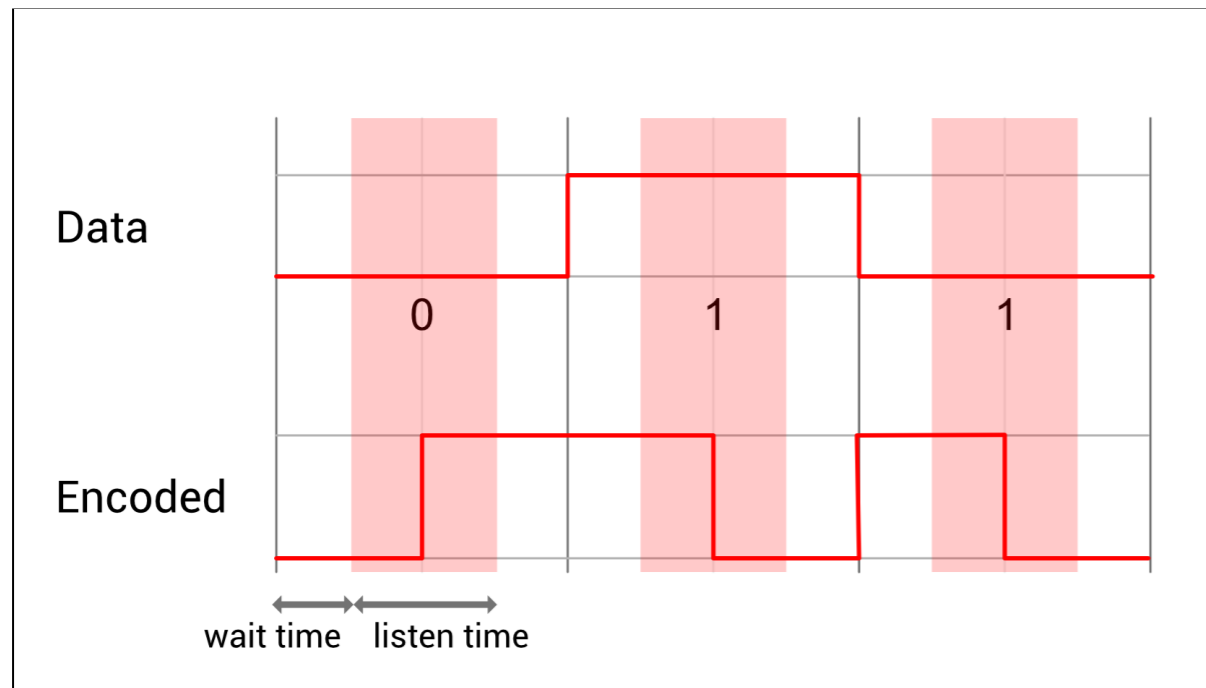
Manchester encoding



Figuur 1: Manchester Encoding

⁴ Data pakketjes, zie het hoofdstuk *frames*.

Bij communicatie tussen computers op laag 1 wordt sporadisch gebruik gemaakt van manchester encoding. Dit gebeurt bijvoorbeeld bij internet over kabels op lage snelheid (Eater, 2014). (Voor hogere overdrachtssnelheid zijn er modernere protocollen⁵, zoals QAM-OFDM, dat wordt bij snelle draadloze verbindingen gebruikt en werkt door middel van modulatie van golven.) Bij Manchester encoding wordt een data⁶ signaal en een clock⁷ signaal gecombineerd, dit is handig voor verbindingen waar maar één kanaal (bijvoorbeeld kabel) is. Dit betekent dat het signaal op twee momenten kan veranderen van hoog (1) naar laag (0) of vice versa, er kan een overgang plaatsvinden aan het begin van de bit, en er vindt altijd een overgang plaats in het midden van de bit. Het ontvangende apparaat negeert de eerste overgang als die er is en luistert alleen rond het midden van de bit naar de andere overgang, deze luisterperiode is in figuur 2A aangegeven met rood. Als deze overgang bijvoorbeeld van laag naar hoog gaat betekent het een "0" en van hoog naar laag een "1".



Figuur 2A: Listen time in Manchester encoding

PWM encoding

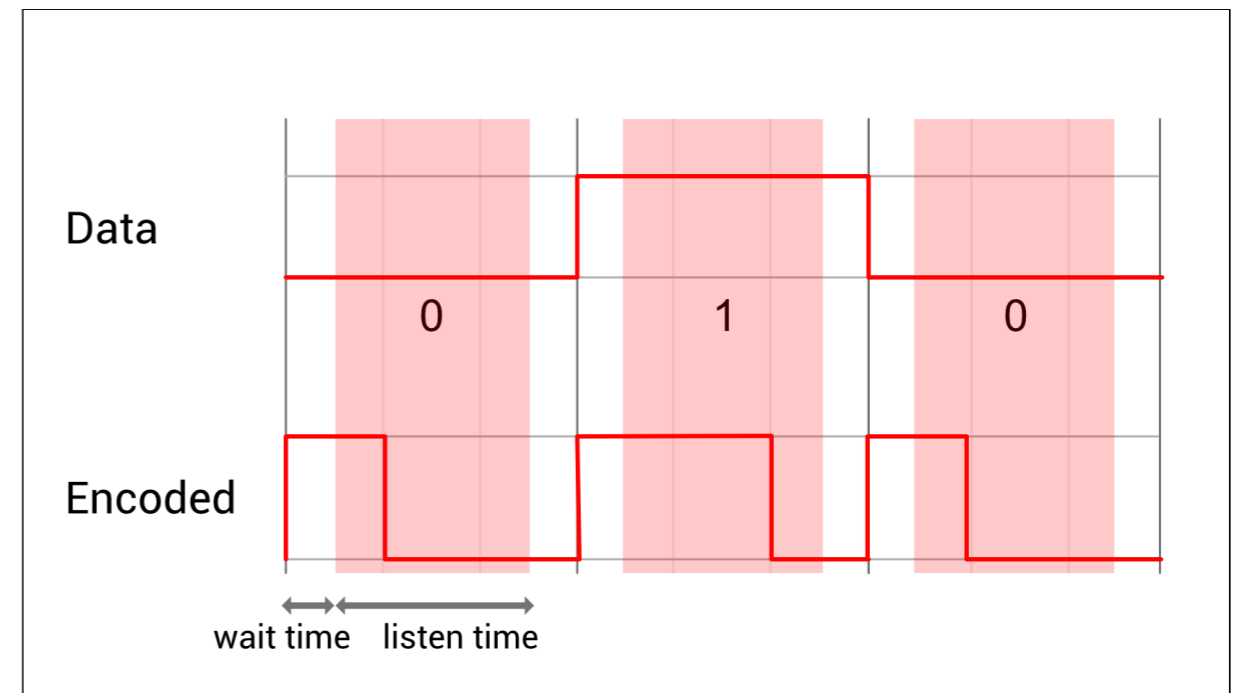
Pulse Width Modulation of PWM is een alternatief voor manchester encoding. PWM varieert de lengte van een pulse om een signaal over te brengen. Zo kan een pulse⁸ van 10ms van licht of spanning bijvoorbeeld staan voor een 0, en een pulse van 20ms voor een 1, zie figuur 2B. Dit kan ook uitgebreid worden met meer lengtes die staan voor meer symbolen.

⁵ Standaarden voor communicatie of notatie.

⁶ Signaal waar informatie wordt verstuurd.

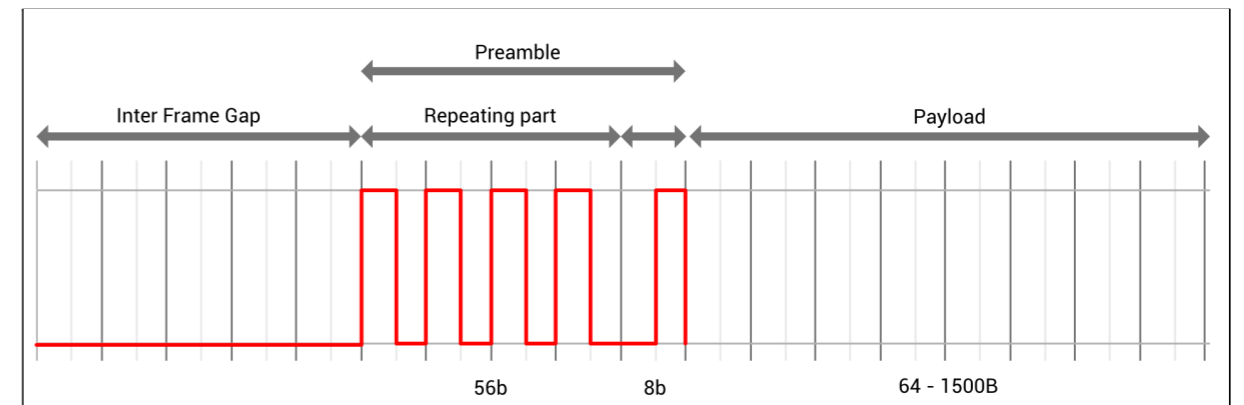
⁷ Afgesproken interval waarop een bit wordt verstuurd.

⁸ Korte piek van stroom of voltage.



Figuur 2B: Pulse Width Modulation Encoding

Frames



Figuur 3: opbouw van een frame.

Het is gebruikelijk om data niet in één keer te versturen, maar in kleinere pakketjes, dit gebeurt op laag 2 en op laag 4. Op laag 2 noemen we dit soort pakketjes frames. Frames. Deze bestaan uit een aantal fases (Eater, 2014). De eerste is de Inter Frame Gap (IFG), dit is een stuk van het frame waar er niks wordt verstuurd. Daarna komt de preamble, hier wordt er steeds "01" verstuurd zodat de ontvanger kan synchroniseren, daarna wordt er een keer "11" verstuurd om te ontvanger te laten weten dat de bruikbare data daarna begint. Daarna volgt de payload, dit is de data die wordt verstuurd of de bruikbare data. Soms wordt dan ook nog een end-of-frame symbol verstuurd, om te laten weten dat alle data is verstuurd.

Foutdetectie en -correctie

Tijdens het versturen van data kunnen verschillende factoren voor ruis en fouten zorgen. Om te zorgen dat alle data correct overkomt, moeten deze fouten kunnen worden gedetecteerd en eventueel gecorrigeerd, dit gebeurt op laag 2. Voor het detecteren kan er gebruik worden gemaakt van parity bits (3Blue1Brown, 2020). In een stuk data wordt het aantal enen altijd even gehouden door een extra bit die de parity bit wordt genoemd. De zender heeft de verantwoordelijkheid om een parity bit toe te voegen en daarmee het aantal enen even te maken. Als de ontvanger dan data ontvangt waarbij het aantal enen oneven is betekent dit dat er een fout is opgetreden, deze kan niet direct worden gecorrigeerd. Ook werkt dit niet als er twee fouten zijn opgetreden.

Een geavanceerde variant van parity bits zijn Hamming Codes (Jeurissen & Leon, 2004). Hierbij worden er meerdere parity bits gebruikt, die allemaal het aantal enen in een eigen blok data even houden. Die blokken zijn zo geplaatst dat er als er één fout optreedt er precies bepaald kan worden waar hij heeft plaatsgevonden. En omdat een bit alleen 1 of 0 kan zijn, kan de fout direct worden gecorrigeerd. Bij meerdere fouten kan het meestal wel worden gedetecteerd, maar niet gecorrigeerd.

Hamming codes zijn niet heel geschikt voor fouten bij internetverbindingen, omdat fouten vaak optreden in een groep van meerdere fouten achter elkaar. Daarom zijn er nog andere algoritmes ontwikkeld om fouten te detecteren en verbeteren, zoals Reed-Solomon Codes en Turbo Codes. Deze zijn echter wiskundig zeer ingewikkeld.

LED of laser

Een belangrijke factor bij communicatie over licht is de keuze tussen een LED-lamp(je) of een laser. Beiden zijn beter geschikt voor een eigen situatie. Bij LED kan men denken aan bijvoorbeeld een huiskamer situatie, waar apparaten zoals een laptop of telefoon een lichtsensor hebben die het knipperen van een lamp in het plafond kan detecteren. Bij laser kan men denken aan situaties waar het lastig is om kabels aan te leggen zoals bijvoorbeeld bij een eiland dichtbij⁹ van de kust. Het voordeel van lasers is dat het door de evenwijdige lichtstralen weinig intensiteit verliest bij langere afstanden, bij LED licht divergeren de lichtstralen en neemt de intensiteit kwadratisch af. Dit betekent dat bij communicatie met LED een veel gevoelige sensor nodig is, of een veel sterkere lichtbron. Daarentegen heeft LED het voordeel dat het niet gericht hoeft te worden.

Hardware

De gekozen onderdelen moeten aan bepaalde eisen voldoen, voor alle onderdelen geldt dat ze makkelijk beschikbaar moeten zijn voor consumenten, en dus niet te duur. Ze moeten allemaal voldoende snelheid hebben, voor de microprocessors betekent dit dat de processor op een snelheid van meerdere megahertz moet werken, voor de andere onderdelen

⁹ Op grotere afstanden wordt de kromming van de aarde en de lichtbreking van de atmosfeer een probleem.

betekent dit dat ze een niet te lange elektrische opwarm- of afkoeltijd moeten hebben, oftewel een niet te groot condenserend vermogen.

Voor een laser geldt verder dat hij gezien de veiligheid niet te krachtig moet zijn, ook omdat een krachtige laser een eigen stroomvoorziening nodig heeft is dat niet voordelig. Dit betekent dat een laser van klasse 2 van 0 tot 1 mW, deze leveren ook geen risico voor oogschade op.

Voor een LED geldt dat deze genoeg licht moet uitstralen om opgevangen te worden door de sensor, dit kan mogelijk niet met een kleine LED die direct aangesloten kan worden op een microprocessor. Daarom kan er gekozen worden voor een grotere LED-lamp die wordt aangestuurd met behulp van een MOSFET¹⁰ transistor.

Voor de lichtsensor geldt dat deze voldoende gevoelig moet zijn. Er kan ook gekozen worden om het signaal te versterken met een operational amplifier of OpAmp. Omdat de snelheid hier van belang is is een photodiode het best, LDR's¹¹ hebben namelijk over het algemeen een langere opwarmtijd, als het licht weg is, duurt het dus even voor de stroom ook weg is.

De microprocessors moeten een USB aansluiting hebben waarover gecommuniceerd kan worden met een computer. Verder moeten de lichtbron en lichtsensor aangesloten kunnen worden, de benodigde voltage voor deze moet dus overeenkomen met een voltage die de processor kan leveren.

¹⁰ MOSFETs kunnen werken met hogere voltages dan normale BJT transistors, een MOSFET is geschikt om met een klein signaal van bijvoorbeeld 5V een groot signaal te controleren van bijvoorbeeld 230V.

¹¹ Light Dependent Resistor.

Experiment

Voor het experiment is gekozen voor een opstelling met een laser en een lichtsensor. Dit omdat de voordelen van een laser erg voordelig zijn voor deze proof-of-concept.

Benodigdheden

- Arduino Uno
- Arduino Nano
- [Photodiode](#)
- [Laser](#)
- [Opamp](#)
- Breadboard
- Draadjes voor Arduino's
- USB kabels voor Arduino's
- 2x Computer
- Aluminium profiel
- 2x Bracket voor vasthouden laser en photodiode

Methode - Programmeren

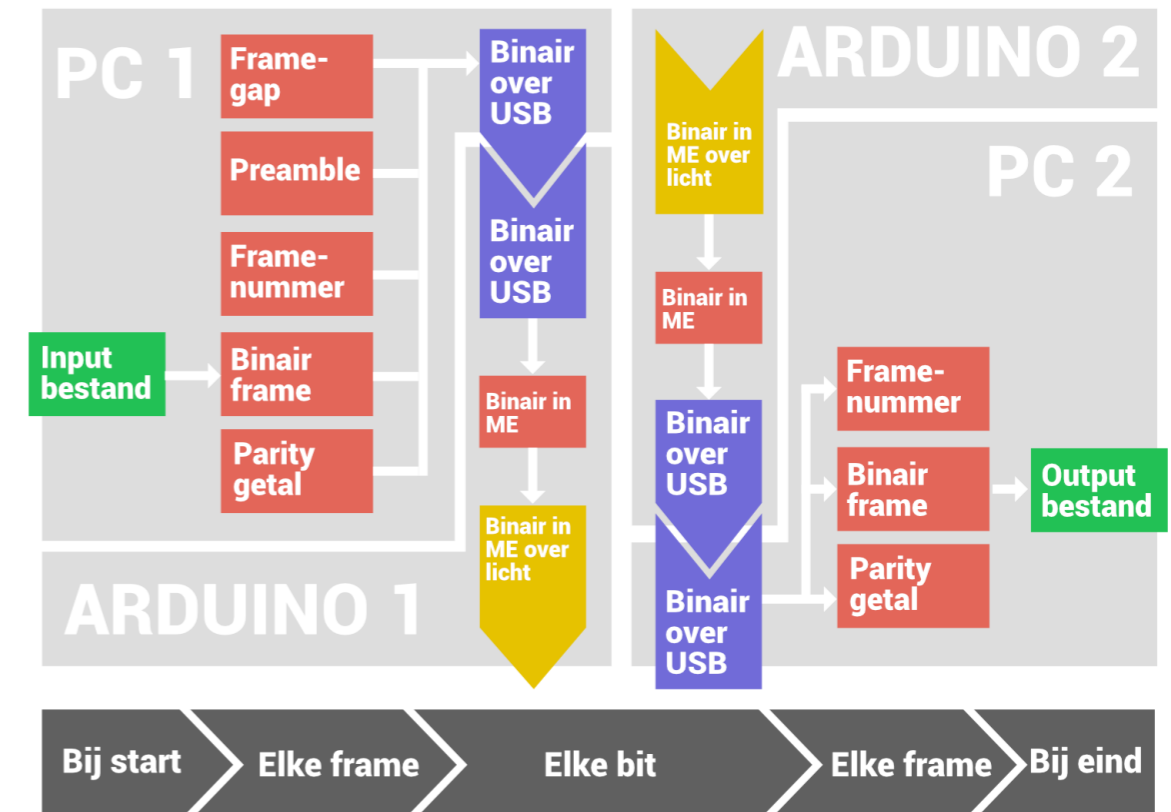
Alle programma's kunnen worden gedownload van [github](#). Daarbij staat ook `readme.md` met daarin instructies voor het installeren van de programma's.

In deze programma's wordt gebruik gemaakt van Manchester encoding voor de communicatie over licht tussen de Arduino's. Ook wordt er gebruik gemaakt van frames zoals weergegeven in figuur 3, er wordt geen end-of-frame verstuurd omdat de lengte van de frames niet variabel is. De foutdetectie wordt gedaan met een versie van parity bits. Hierbij wordt aan het einde van de payload een 16 bits cijfer¹² toegevoegd wat staat voor het aantal enen in de payload, dit volstaat ook voor grotere payloads van bijvoorbeeld 1024 bits. Als het aantal enen niet gelijk is aan dat cijfer dan is er ergens een fout opgetreden en wordt de frame niet opgeslagen, hiermee kan elk oneven aantal fouten worden gedetecteerd.

¹² Getal tussen 0 en 65536.

Overzicht dataflow

In figuur 4 is te zien hoe de datastromen tijdens het experiment lopen. De metingen hierbij gaan over de overdracht van de eerste tot de laatste USB verbinding, omdat het moeilijk is om tijdens de tussenstappen extra metingen te doen.

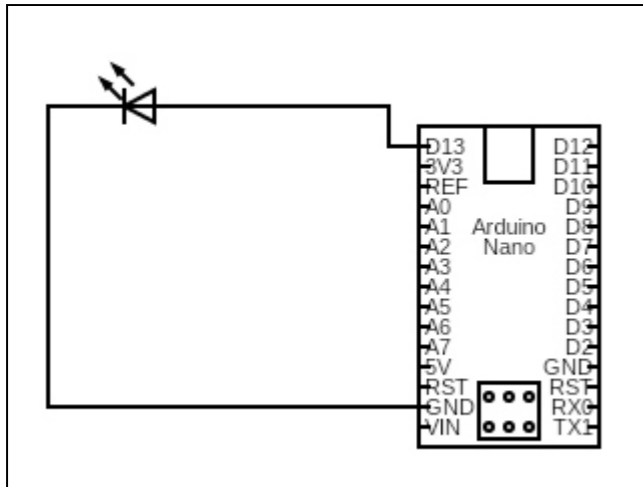


Figuur 4: overzicht van datastromen

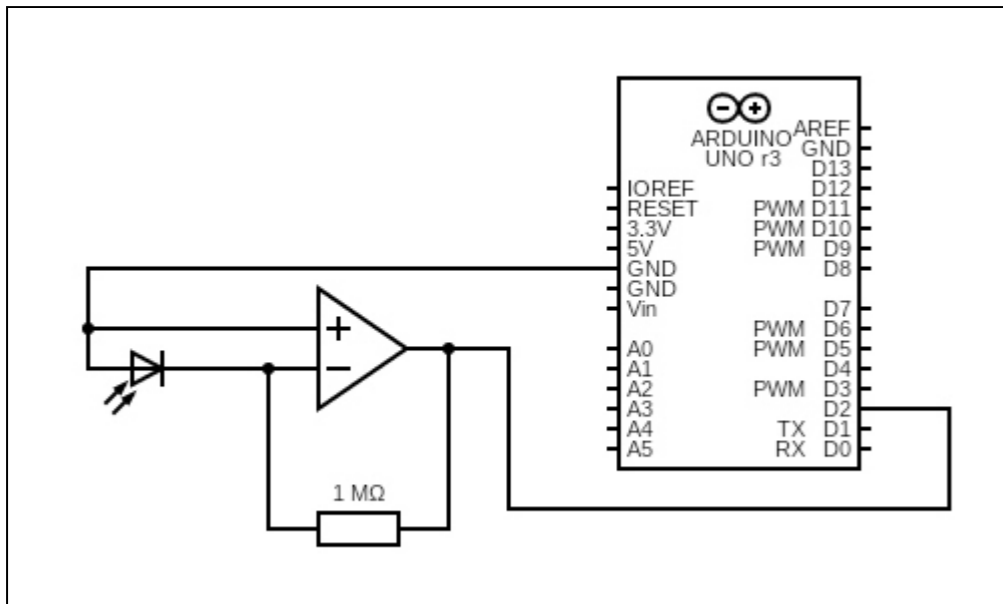
Methode - Experiment

Vorbereiding

1. Sluit de arduino's aan volgens de schema's in figuur 5 en figuur 6.

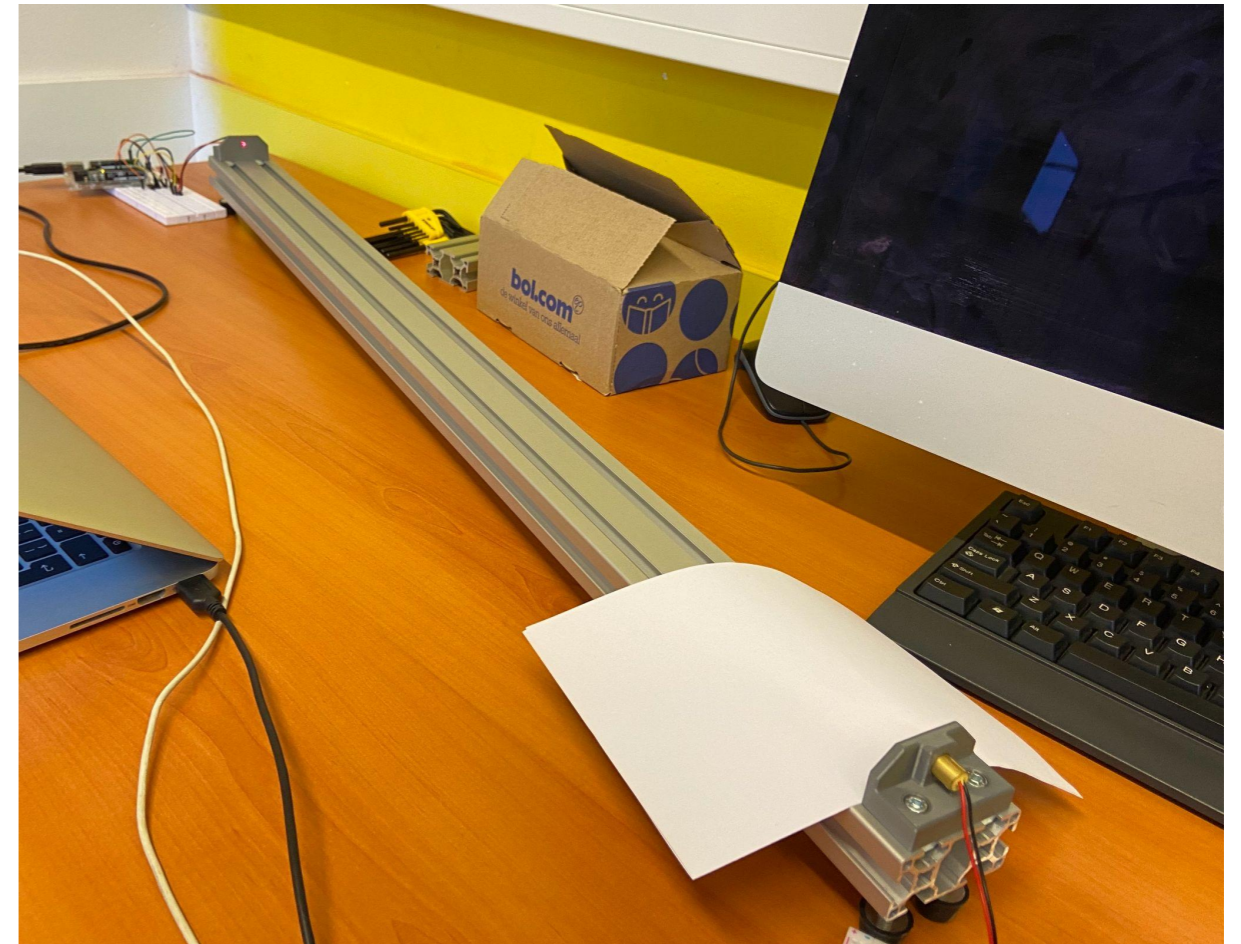


Figuur 5: schema Arduino Nano en Laser (Laser is afgebeeld als een LED)



Figuur 6: schema Arduino Uno, Opamp en Photodiode

2. Maak de laser en de photodiode vast in de brackets en bevestig ze op het aluminium profiel zoals in figuur 7.
3. Richt de laser zodat hij precies de fotodiode raakt (om de laser tijdelijk aan te zetten kan hij op 5V aangesloten worden).



Figuur 7: Opstelling

4. Open de file constants.py en vul voor `RECEIVINGDEVICE` en `SENDINGDEVICE` de adressen van respectievelijk de Arduino Uno en Arduino Nano in.
5. Controleer of alle waarden die gemarkeerd zijn in tabel 1 overeenkomen met de waarden in constants.py.

(Voor hogere accuraatheid kan `TESTTIME` in constants.py veranderd worden naar een hogere waarde (standaard is 180 s). Ook kan besloten worden om tests meerdere keren te doen.)

6. Zorg dat er geen andere programma's open staan en verbreek de internetconnectie zodat er geen automatische updates worden uitgevoerd. Dit alles zodat er geen andere variabelen veranderen die de meetwaarden kunnen beïnvloeden. Houd ook de lichtomstandigheden gelijk en indien mogelijk de temperatuur. Verwijder ook USB apparaten zoals muizen en toetsenborden.

Metingen

Herhaal steeds de volgende stappen. Raak de opstellingen niet aan tijdens het doen van de metingen, behalve voor de afstandsmetingen.

1. Open de file constants.py

2. Verander steeds één variabele naar een niet gemarkeerde waarde uit de tabel. Voer ook nog een aantal testen uit met alle standaardwaarden, dit om een idee te krijgen over de gemiddelde afwijking.
3. Upload het programma `sendingManchester.ino` naar de Arduino Nano.
4. Upload daarna het programma `manchesterInterrupts.ino` naar de Arduino Uno.
5. Open een terminal en voer deze opdracht uit: `python3 sendingPC.py`.
6. Open nog een terminal en voer deze opdracht uit: `python3 receivingPC.py`.

Het programma meet nu 3 minuten lang de dataoverdracht en zet automatisch de resultaten in `results.csv`.

Als alle metingen zijn gedaan, doe nog een meting met alle beste waarden van de variabelen om de algemene maximale overdrachtssnelheid te meten.

Tabel 1: Te onderzoeken variabelen en waarden.

DATARATE (s)	GAP LENGHT (b)	PREAMBLELE NGHT (b)	DISTANCE (CM)	PAYLOAD LENGHT (b)
0.004	8	8	3	32
0.003	16	16	25	64
0.002	24	24	50	128
0.0018	32	32	75	192
0.0015	40	96	95	256
0.0013	48			320
0.001	64			384
0.0008				448
0.0005				512
0.00065				640
				768
				896
				1024

Resultaten

Results.csv met daarin alle tabellen is te downloaden van [github](#). Hieronder staan grafieken gemaakt met R¹³. In de grafieken wordt gesproken over *snelheid correct aangekomen frames*, deze waarde wordt berekend door de hoeveelheid correct aangekomen frames te vermenigvuldigen met de lengte van een heel frame in bits. Dit wordt dan gedeeld door de tijdsduur van de test. Voor de *snelheid correct aangekomen payloads* geldt hetzelfde maar dan wordt er vermenigvuldigd met de lengte van één payload in plaats van de hele frame. De *snelheid correct aangekomen frames* geeft aan hoe snel er nuttige data verstuurd kan worden en kan ook *werkelijke snelheid* genoemd worden.

Bij meerdere tests worden er boxplots gemaakt om de spreiding en accuraatheid van de resultaten te laten zien, ook wordt er eventueel automatisch een kromme gegenereerd rond de medianen. In de grafieken wordt het *percentage correct aangekomen frames* aangegeven met blauw, de *snelheid correct aangekomen frames* met groen en de *snelheid correct aangekomen payloads of werkelijke snelheid* met rood.

Symbolen bij de grafieken

n: aantal tests

t (s): lengte test in seconden

f (b/s): frequentie in bits per seconden

l (cm): afstand tussen verzender en ontvanger in centimeter

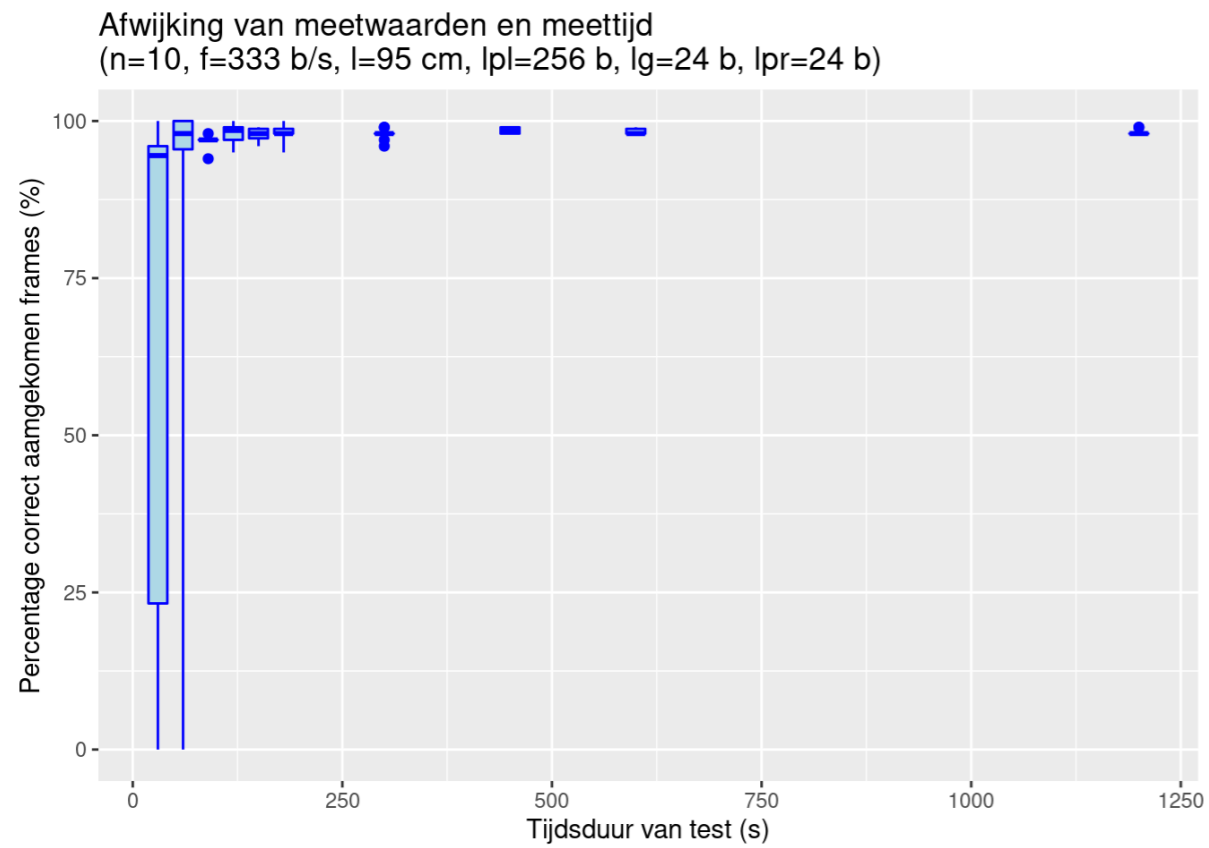
lpl (b): lengte van payload in bits

lg (b): lengte van IFG in bits

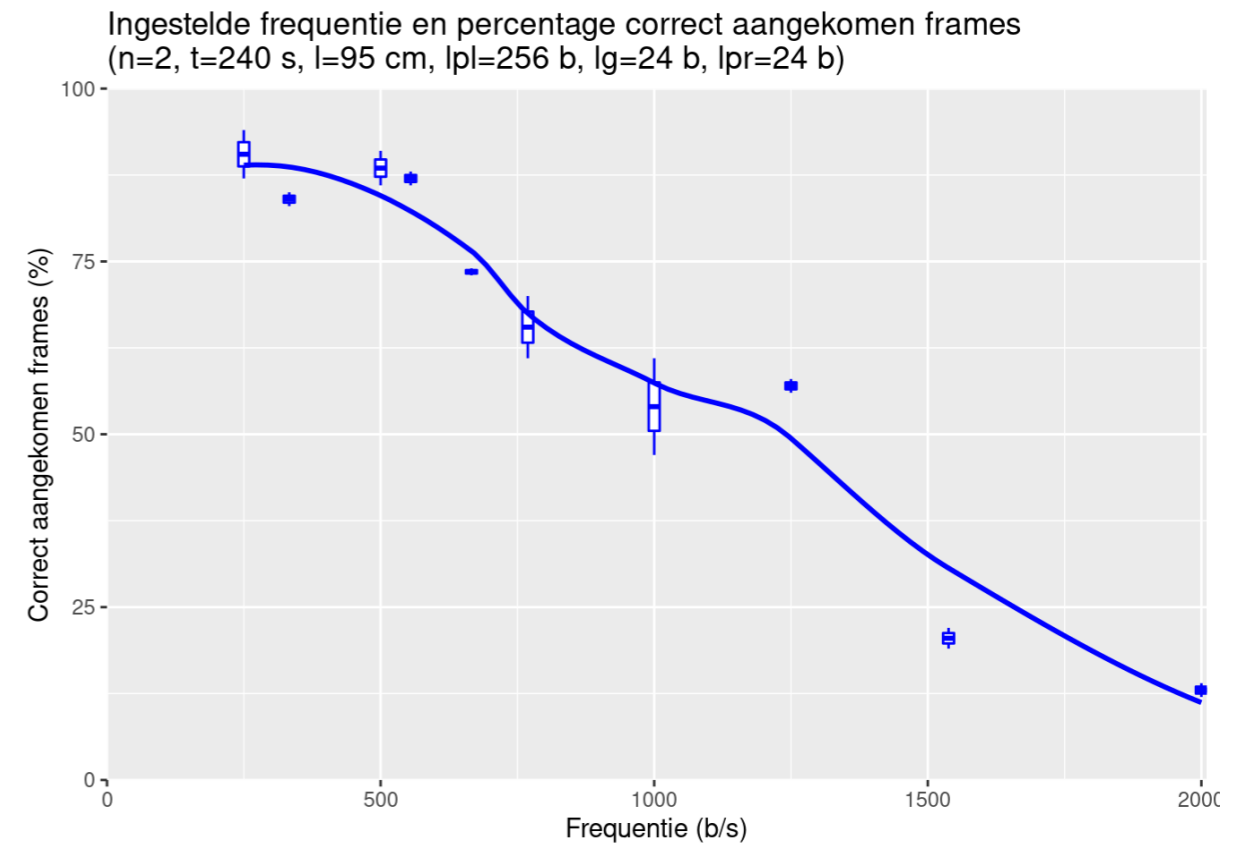
lpr (b): lengte van preamble in bits

¹³ Programmeertaal voor visualisatie van data.

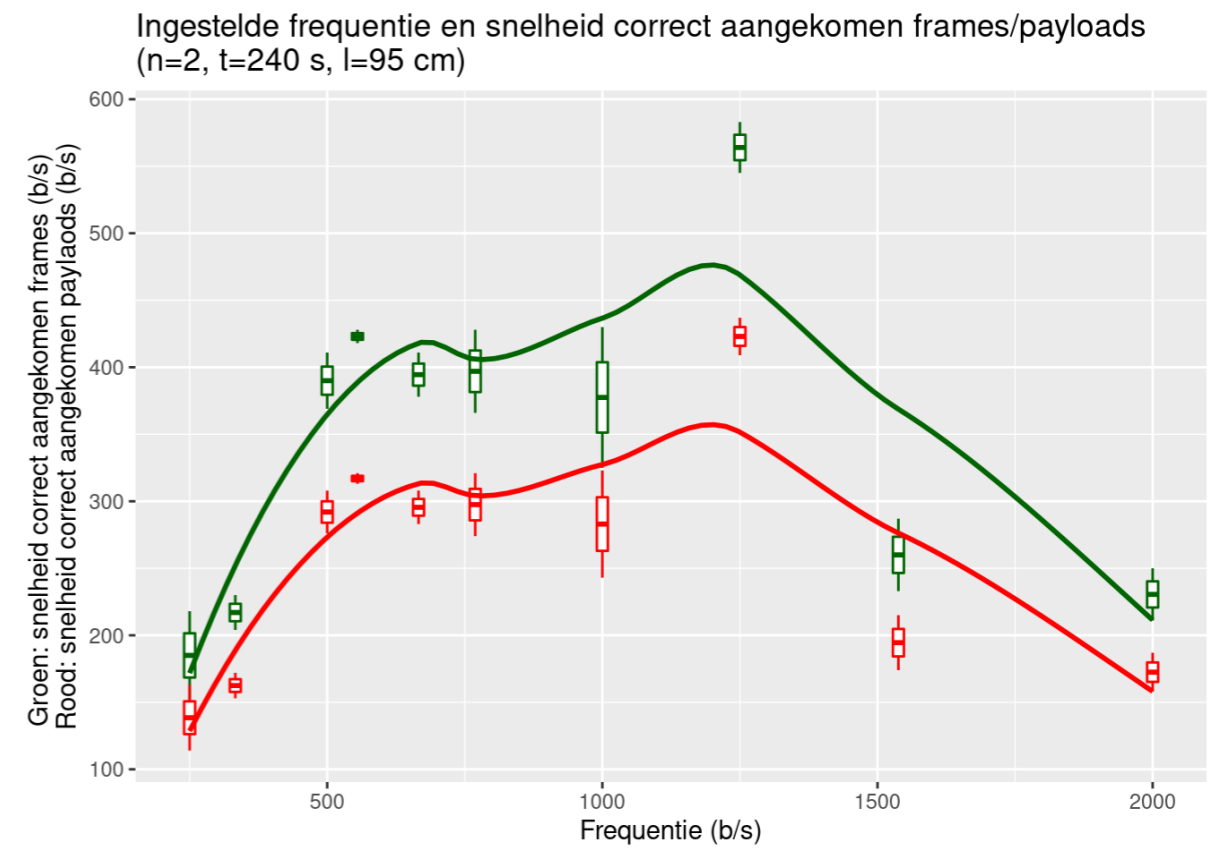
Grafieken



Figuur 8: Afwijking in resultaten wordt lager naarmate testen langer zijn. Vanaf 180 s is acceptabel, na 450s is er weinig verbetering.

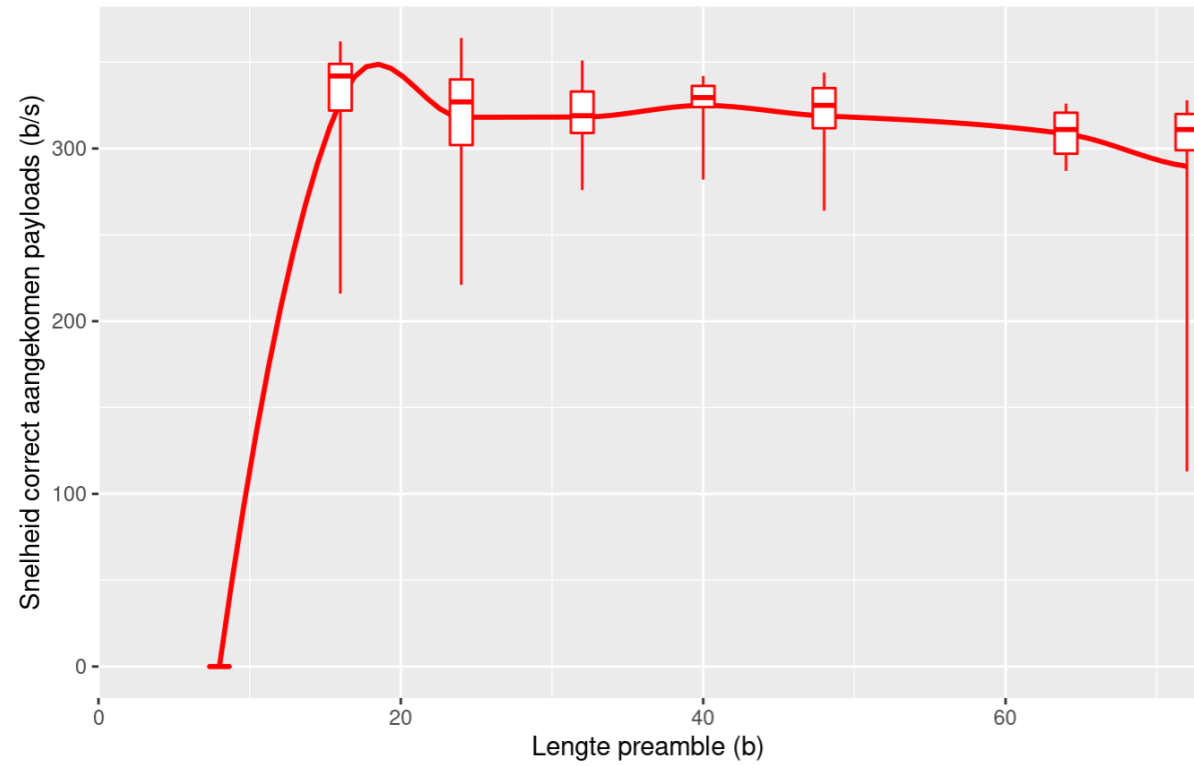


Figuur 9: Meer fouten bij hogere snelheden.



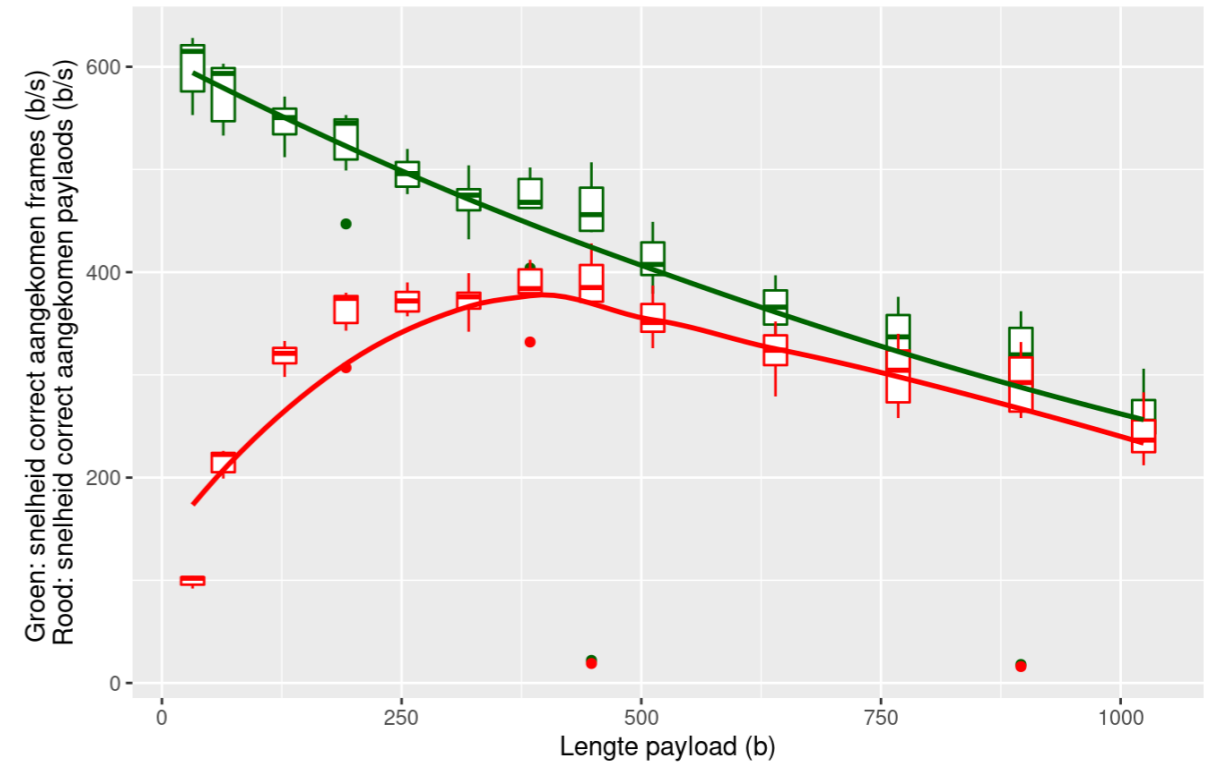
Figuur 10: Snelheid correct verstuurd data is optimaal bij een frequentie van 1250 b/s.

Lengte van preamble en snelheid correct aangekomen bits
($n=12$, $t=180$ s, $f=500$ b/s, $l=95$ cm, $lpl=256$ b, $lg=24$ bb)



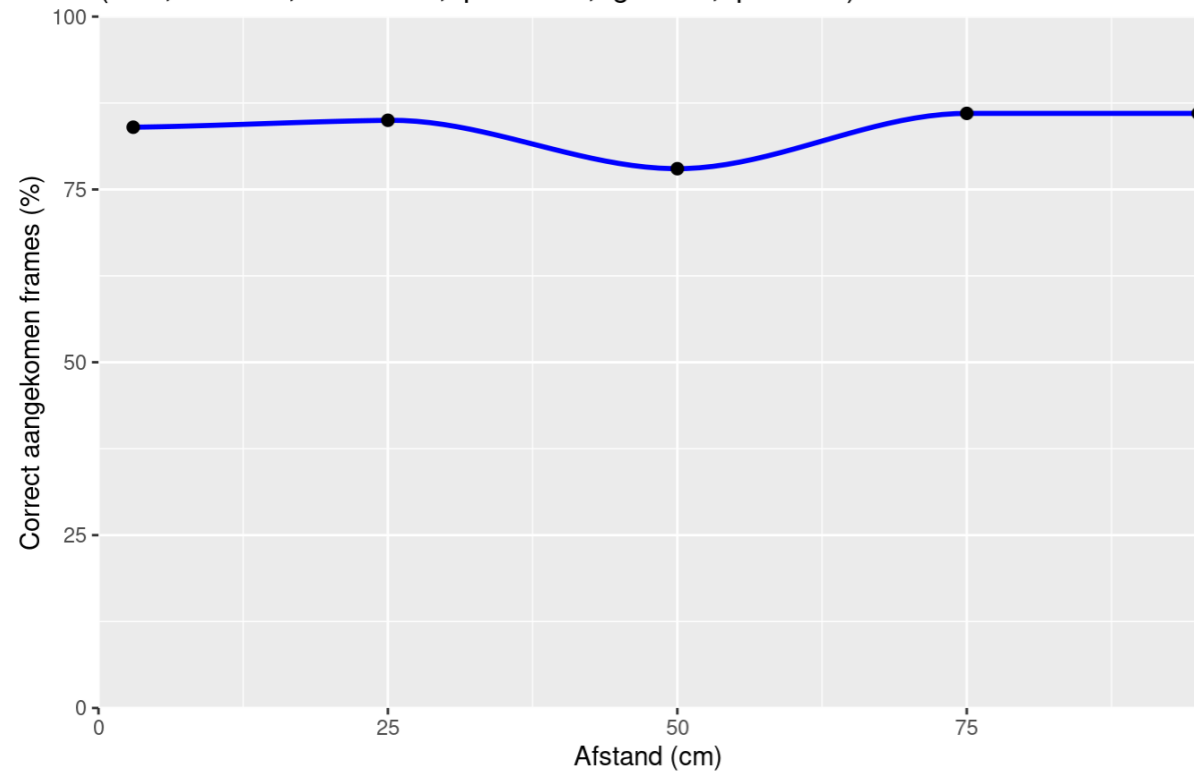
Figuur 11: Preamble van 16 bits is optimaal.

Lengte van payload en snelheid correct aangekomen frames/payloads
($n=6$, $t=180$ s, $f=666$ b/s, $l=95$ cm, $lg=24$ b, $lpr=24$ b)



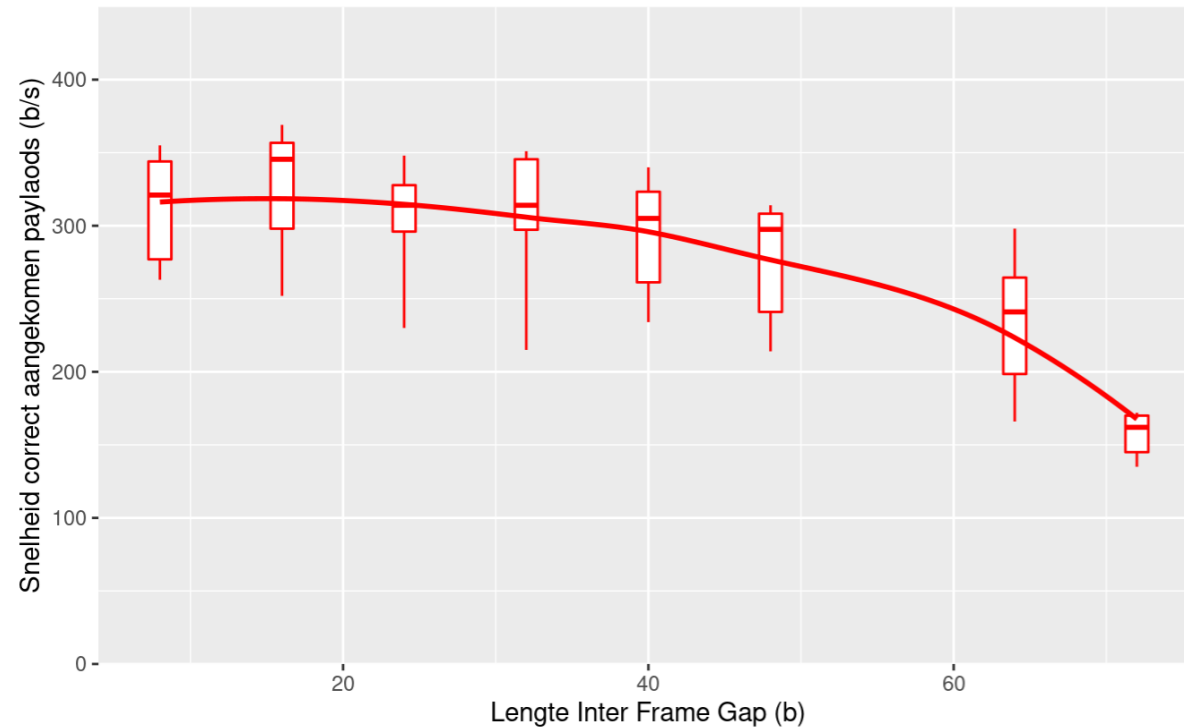
Figuur 13: Kortere frames komen het vaakst goed aan maar hebben te veel onbruikbare data, optimum ligt bij 384 bits.

Afstand en percentage correct aangekomen frames
($n=1$, $t=180$ s, $f=500$ b/s, $lpl=256$ b, $lg=24$ b, $lpr=24$ b)



Figuur 12: Afstand heeft weinig invloed.

Lengte van Inter Frame Gap (IFG)
en snelheid correct aangekomen frames/payloads
($n=12$, $t=180$ s, $f=500$ b/s, $l=95$ cm, $lpl=256$ b, $lpr=24$ b)



Figuur 14: Langere IFG zorgt voor meer correcte frames maar ook meer onbruikbare data, optimum ligt bij 24.

Conclusie

Uit de grafieken blijkt een aantal dingen.

- Er zijn meer fouten in de dataoverdracht bij hogere snelheden (figuur 9).
- Bij frequenties hoger dan 1250 b/s weegt de hogere overdrachtssnelheid niet meer op tegen het aantal fouten (figuur 10).
- Bij een langere IFG komt het frame vaker correct aan, maar bij een IFG langer dan 24 bits is hier weinig profijt meer van (figuur 14).
- Voor de lengte van de preamble geldt dat een langere preamble zorgt voor minder correct aangekomen frames. De optimale lengte van de preamble ligt bij 16 bits, hieronder is die te kort en komt er bijna niks aan (figuur 11).
- De afstand tussen verzender en ontvanger lijkt een kleine invloed te hebben, maar deze verschillen kunnen ook verklaard worden door afwijkingen in de metingen (figuur 12).
- Een langere payload zorgt voor een kleinere kans dat de frame correct aankomt, maar zorgt wel voor minder onbruikbare data (IFG, preamble, framenummer, etc). Het optimum hiervan ligt tussen 128 en 512 bits (figuur 13).
- De hoogst gemeten werkelijke snelheid is 450 b/s.

Deze snelheden zijn nog verre van praktisch toepasbaar, normaal internet behaalt snelheden van 25Mbps (FCC, 2018) oftewel 50.000 maal sneller. Desondank kan er gezegd worden dat er relatief snel data verstuurd kan worden, het experiment kan gezien worden als een proof-of-concept. Er kan nog veel verbeterd worden, maar LiFi technologie is zeker veelbelovend.

Discussie

De metingen van dit experiment kunnen op een aantal manieren verbeterd worden. De meeste testen duurde 3 minuten, door dit langer te maken kunnen preciezere resultaten behaald worden. Dit blijkt uit de grafiek met significantie (figuur 8). Bij de testen voor afstand moest de photodiode steeds verplaatst worden, dit kan ervoor hebben gezorgd dat de laser net anders op de sensor viel, dit kan invloed hebben gehad op de gemeten waarden. De resultaten voor afstand kunnen daarom worden genegeerd. De testen voor frequentie zijn maar twee keer uitgevoerd ($n=2$), daarom zijn ze minder precies dan de andere testen. Van de extra buiging in de grafiek voor frequentie kan niet zeker worden gezegd of het komt door een meetfout of niet.

Voor grotere snelheden kunnen er nog een aantal verbeteringen worden overwogen. Er kan eventueel gebruik gemaakt worden van de ingebouwde UART¹⁴ module in Arduino voor snellere en betrouwbaardere data overdracht. De optimale frequentie van 1250 b/s kan een aantal redenen hebben. Het zou bijvoorbeeld kunnen komen doordat de hardware van de Arduino niet snel genoeg is om nog sneller bij te houden. De laser heeft ook een zeker condenserend vermogen (het houdt energie vast en het duurt even voordat hij op volledige kracht komt), het kan dat daardoor de lichtpulsen langer duren dan de ingestelde bittijd. De

¹⁴ Ingebouwde module in Arduino voor seriële communicatie (bits na elkaar sturen over een kanaal in plaats van tegelijk over meerdere kanalen).

overdrachtssnelheid kan ook nog verbeterd worden door het gebruik van Hamming codes, waardoor meer frames bruikbaar zijn. Ook kan er nog worden onderzocht of de Arduino's vervangen kunnen worden door snellere chips of FPGAs¹⁵, die specifiek voor deze taak kunnen worden ingesteld.

Aanbevelingen

Bij vervolgonderzoek kunnen Hamming codes worden gebruikt om meer frames bruikbaar te maken. Om de situatie realistischer te maken kan er bij worden gekeken naar een bidirectionele opstelling, waarbij de zender ook data ontvangt en andersom. Hierbij kunnen er ook acknowledgements worden verstuurd om aan te geven dat een frame correct is ontvangen, of verzoeken om bepaalde frames opnieuw te versturen. De code voor deze acknowledgements is al deels aanwezig in de programma's, dus dit kan vrij eenvoudig worden toegepast.

Bronnenlijst

Agentschap Telecom. (2021, 21 juli). KPN, T-Mobile en VodafoneZiggo verwerven frequenties via Multibandveiling. Geraadpleegd op 14 januari 2022, van <https://www.agentschaptelecom.nl/actueel/nieuws/2020/07/21/kpn-t-mobile-en-vodafoneziggo-o-verwerven-frequenties-via-multibandveiling>

3Blue1Brown. (2020, 4 september). How to send a self-correcting message (Hamming codes). Geraadpleegd op 8 november 2021, van <https://www.youtube.com/watch?v=X8jsijhIIIA>

3Blue1Brown. (2020, 4 september). Hamming codes part 2, the elegance of it all. Geraadpleegd op 8 november 2021, van https://www.youtube.com/watch?v=b3NxrZOu_CE

Cloudflare. (z.d). What is the OSI Model?. Geraadpleegd op 9 maart 2022, van <https://www.cloudflare.com/en-gb/learning/ddos/glossary/open-systems-interconnection-model-osi/>

Eater, B. (2014, 30 oktober). Analyzing actual Ethernet encoding | Networking tutorial (4 of 13). Geraadpleegd op 14 januari 2022, van <https://www.youtube.com/watch?v=i8CmibhvZ0c>

Eater, B. (2014, 30 oktober). The importance of framing | Networking tutorial (5 of 13). Geraadpleegd op 8 november 2021, van <https://www.youtube.com/watch?v=xrVN9jKjIKQ>

FCC. (2018, 2 februari). 2018 Broadband Deployment Report. Geraadpleegd op 9 maart 2022, van <https://www.fcc.gov/reports-research/reports/broadband-progress-reports/2018-broadband-deployment-report>

Jeurissen, Ruud., & Broek, Leon. (2004). Nullen en Enen. Utrecht, Nederland: Epsilon Uitgaven.

Yu-Chieh, C., Hsieh, D., Tsai, C., Chen, H., Kuo, H., & Lin, G. (2015). 450-nm GaN laser diode enables high-speed visible light communication with 9-Gbps QAM- OFDM. OPTICS EXPRESS 13051, 23(10).

¹⁵ Field Programmable Gate Array: Chip waarvan de hardware kan worden "geprogrammeerd" om specifieke taken uit te voeren.