

Sensors & Interfacing

Tuur Vanhoutte

27 april 2020

Inhoudsopgave

1 Communicatie	1
1.1 Datacommunicatie in IoT	1
1.1.1 De 3 lagen	1
1.2 Data	1
1.3 Communicatie	1
1.4 Communicatieafspraken	2
1.5 Encoding/Decoding	2
1.6 Signalen	2
1.7 Communicatiemedia	2
1.7.1 Eigenschappen van media	3
1.7.2 Afspraken	3
1.7.3 Standaardiseren van	3
2 Analoog vs digitaal	4
2.1 Toestanden	4
2.1.1 Bepaalde toestand	4
2.1.2 Digitale toestanden	4
2.1.3 Analoge toestanden	4
2.2 Signalen	4
3 Analoge signalen	5
3.1 Omzetten van analoge signalen	5
3.1.1 Transducer	5
3.1.2 Sensoren en Actuatoren	5
3.2 Analoge communicatie	6
3.3 Analoog signaal	6
3.3.1 Eigenschappen	6
3.3.2 Wisselspanning - Eigenschappen	6
3.3.3 Periodieke signalen	7
3.3.4 Tijdsdomein en frequentiedomein	7
4 Digitale signalen	8
4.1 Eigenschappen	8
4.2 Duty Cycle	8
4.3 Flanken (edge)	9
4.4 Weergave digitale signalen	9
5 AD conversie	10
5.1 Analoog naar digitaal	10
5.1.1 Voorbeeld A	10
5.1.2 Voorbeeld B	10
5.2 Eigenschappen	11
5.2.1 Quantisatiefouten	11
5.2.2 Sample Rate / sample frequentie	11
5.2.3 Aliasing	12
5.2.4 Oversampling	13
5.3 Implementatie en types	13
5.3.1 Flash ADC	13
5.3.2 Successive approximation ADC	14
6 Digitaal naar analoog conversie	15

6.1	Simpele DAC	15
6.2	Simpele DAC met PWM	16
6.3	Andere types	16
6.4	Extra info over AD/DA conversie:	16
7	Modulatie	17
7.1	Demodulatie	17
7.2	Modem	17
7.3	Waarom?	18
7.4	Hoe?	18
7.5	Draaggolf	18
7.5.1	Parameters van een draaggolf	19
7.6	Simpele modulatie	19
7.7	Amplitude modulatie (=AM)	19
7.7.1	Overmodulatie	20
7.7.2	AM bandbreedte	21
7.7.3	SSB modulatie (USB/LSB)	22
7.7.4	ASK Modulatie	23
7.8	Frequentie modulatie (=FM)	24
7.8.1	FSK modulatie	25
7.8.2	FM Modulatie	25
7.8.3	Fase modulatie	26
7.8.4	PSK modulatie	26
7.8.5	Phase Shift Modulatie	27
7.8.6	QAM modulatie	27
7.8.7	Bandbreedte / vermogen	28
7.8.8	Waarom Narrow-band?	29
8	RF-spectrum	30
8.1	Beschikbare bandbreedte	30
8.2	Licenties en licentievrije banden	31
8.3	ISM banden	31
8.4	Interferentie	31
8.4.1	Co-channel interferentie	31
8.4.2	Signaalseparatie	32
8.4.3	Interferentie van naburige kanalen	32
8.4.4	Interferentie van andere toestellen	32
8.5	Meten van interferentie	33
8.6	Golf lengte	33
8.6.1	Reken voorbeeld	34
8.7	Antennesysteem	34
8.7.1	De dipool antenne	34
8.7.2	Polarisatie van antennes	35
8.7.3	Directionaliteit van antennes	35
8.7.4	Stralingspatroon van antennes	36
8.7.5	Gain van een antenne	36
8.8	Propagatie van RF-signalen	37
8.8.1	Path loss	38
8.8.2	Signaalverzwakking / path loss	38
8.8.3	Dynamic Rate Shifting (DRS)	39
8.8.4	Reflectie	39
8.8.5	Absorptie	40

8.8.6	Doordringbaarheid	40
8.8.7	Scattering	41
8.8.8	Refractie	41
8.8.9	Diffractie	42
8.8.10	Fresnel zones	42
8.8.11	Propagatie van RF-signalen	43
8.8.12	MIMO	44
8.8.13	Multiplexing	44
9	Seriele communicatie	44
9.1	Serieel vs parallel	44
9.2	Parallelle communicatie	45
9.2.1	Clock skew	45
9.3	Seriele communicatie	45
9.4	Serieel vs Parallel: wanneer gebruiken?	46
9.5	SERDIS	46
9.5.1	Implementatie	46
9.6	Duplex	47
9.7	Flow Control	47
9.8	Synchroon vs Asynchroon	48
9.9	Synchrone seriele communicatie	48
9.10	Asynchrone seriele communicatie	48
9.10.1	Parameters	49
9.10.2	Standaardnotering	49
9.10.3	Baudrate	49
9.10.4	Aantal databits	49
9.10.5	Pariteit	49
9.10.6	Stopbits	50
9.11	Snelheid	50
9.12	Meerdere deelnemers	50
9.13	Manchester encoding	50
9.14	Differentiële communicatie	51
9.15	Benoeming van signalen	52
9.16	Protocollen	52
10	Bussystemen	53
10.1	Communicatie op een bus	53
10.2	Master / Slave communicatie	53
10.3	Serial Peripheral Interface (SPI) communicatie	53
10.3.1	Werking	54
10.3.2	Eigenschappen	54
10.3.3	Timing Diagram	55
10.4	I2C communicatie	55
10.4.1	I2C schema	55
10.4.2	Eigenschappen	57
10.5	Flow Control	57
10.5.1	Problemen	57
10.5.2	Hardware flow control	57
10.5.3	Software flow control	57
10.5.4	Geen flow control	57
10.6	Belangrijke bus-interfaces	58

11 Foutdetectie en correctie	58
11.1 Integriteitscontrole	58
11.2 Detectie van fouten	58
11.2.1 Parity check	58
11.3 Foutcorrectie	59
11.3.1 Cyclic Redundancy Check (CRC)	59
11.4 Integriteitscontrole	61

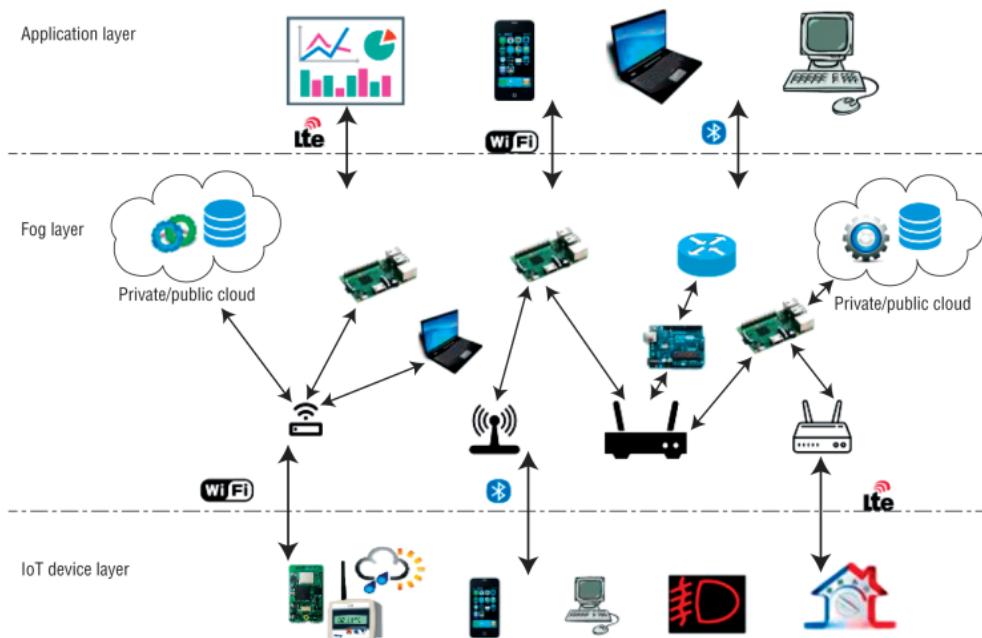
1 Communicatie

1.1 Datacommunicatie in IoT

1.1.1 De 3 lagen

1. Application Layer
2. Fog layer
3. IoT Device Layer

Datacommunicatie in IoT



Figuur 1: Datacommunicatie in IoT

1.2 Data

- "Pre-informatie"
- Gegevens waaruit informatie kan worden gewonnen
- Stelt een bepaalde toestand voor
- <https://en.wikipedia.org/wiki/Data>

1.3 Communicatie

= Overbrengen van informatie tussen deelnemers

- Boodschap

- Signaal
- Medium

1.4 Communicatieafspraken

- Coderen van informatie (encoding)
- Voorbeelden:
 - morse-code
 - Ascii-codering
 - * Codering voor alle gebruikte symbolen in symbolen
 - * Codering in 7 of 8 bit
 - * 1 byte = 1 teken
 - ...

1.5 Encoding/Decoding

3 stappen:

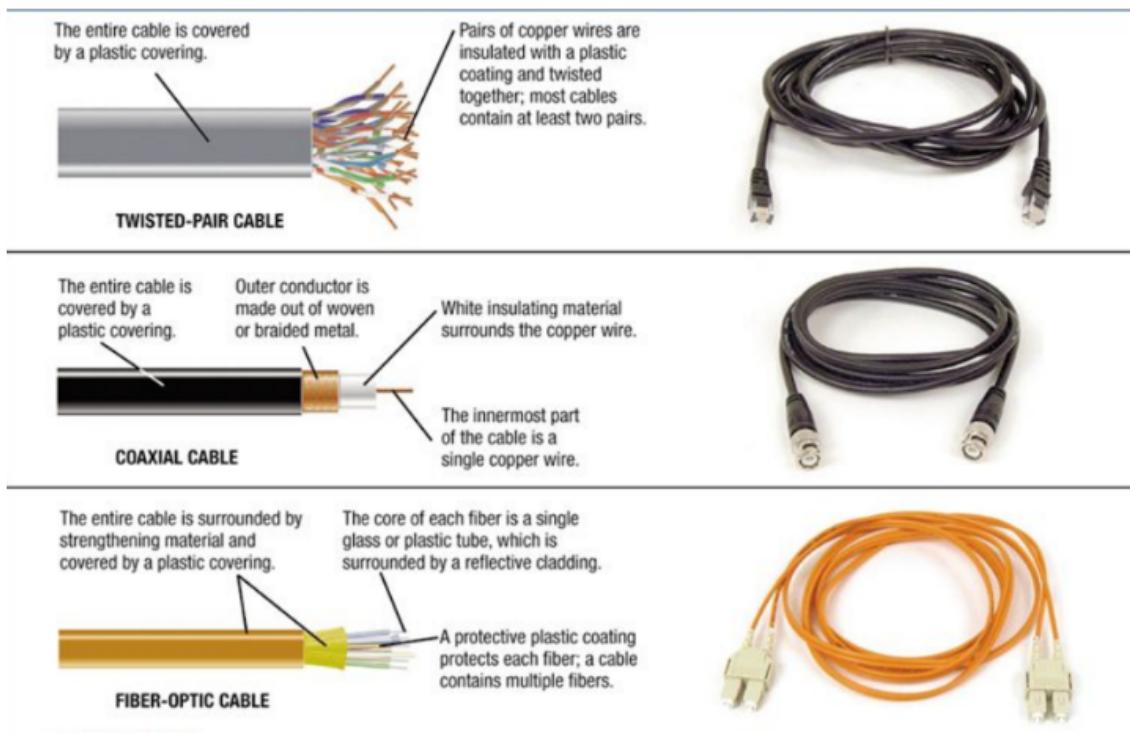
1. Codifying
2. Sending the message
3. Decodifying

1.6 Signalen

- Licht
- Geluid
- Elektriciteit
- ...

1.7 Communicatiemedia

- Twisted-Pair cable
- Coaxial cable
- Fiber-Optic cable



Figuur 2: Soorten kabels

1.7.1 Eigenschappen van media

- Vatbaarheid voor interferentie
- Overbrugbare afstand
- Praktisch
- Kostprijs
- ...

1.7.2 Afspraken

- Protocol
- Standaarden
- IEEE

1.7.3 Standaardiseren van ...

- Type media en zijn specificaties
- Het gebruikte signaal en zijn toleranties
- De elektrische interferentie
- De gebruikte codering

- Foutcorrectiecodes
- Protocol
- De gebruikte connector
- ...

2 Analoog vs digitaal

- **Digitaal:** Discrete waarden
- **Analoog:** Continue waarden

2.1 Toestanden

2.1.1 Bepaalde toestand

- Temperatuur
- Licht aan/uit
- Afstand
- Tijd
- ...

2.1.2 Digitale toestanden

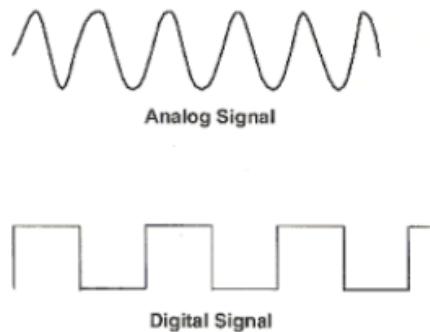
- Licht aan/uit
- Deur open/dicht
- Keuze van versnelling N - 1 - 2 - 3 - 4 - 5 - R
- Ruitenwisser interval uit - interval - traag - snel
- ...

2.1.3 Analoge toestanden

- Tijd (!)
- Temperatuur
- Luchtdruk
- Luchtvochtigheid
- Afstand
- ...

2.2 Signalen

- Analoog signaal
- Digitaal signaal



Figuur 3: Analoog vs digitaal signaal

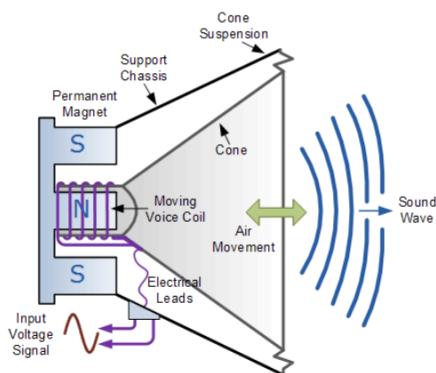
3 Analoge signalen

3.1 Omzetten van analoge signalen

3.1.1 Transducer

Omzetten van een analoog signaal naar een ander analoog signaal.

Voorbeeld: elektrisch signaal omzetten naar een geluidsignaal via een luidspreker (=de transducer)

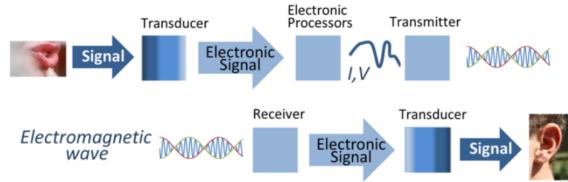


Figuur 4: Luidspreker

3.1.2 Sensoren en Actuatoren

- Sensor ⇒ meten van een fysieke eigenschap
- Actuator ⇒ beïnvloeden van een fysieke parameter ⇒ transducers

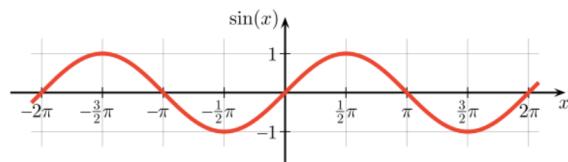
3.2 Analoge communicatie



Figuur 5: Analoge communicatie

3.3 Analoog signaal

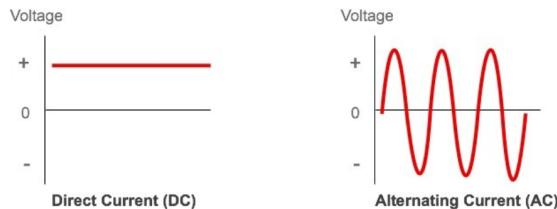
Sinusgolf als meest elementaire signaal



Figuur 6: Sinusgolf

3.3.1 Eigenschappen

- DC vs AC
- Polariteit blijft gelijk bij (pulserende) DC
- Polariteit verandert bij AC



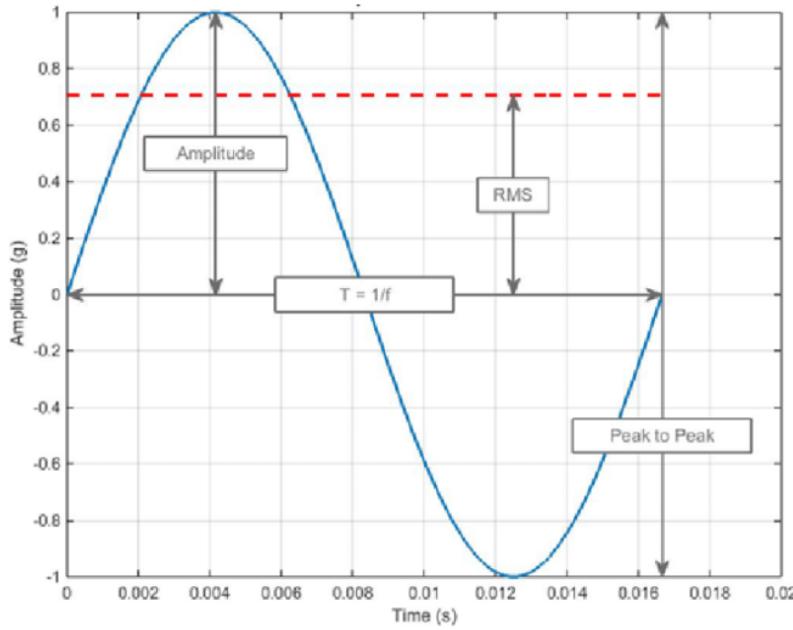
Figuur 7: DC vs AC

3.3.2 Wisselspanning - Eigenschappen

- RMS = Root Mean Square (= kwadratisch gemiddelde) = effectieve waarde (in geval van sinus)
 1. Som van alle kwadraten (= square)
 2. Die som delen door het aantal waardes (= mean)
 3. Neem de vierkantswortel van dat getal

– Wordt vaak gebruikt in de elektriciteit om het gemiddelde vermogen te vinden
- Frequentie

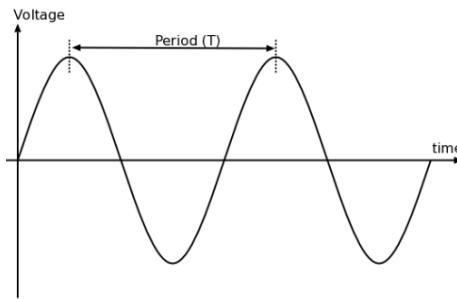
- Periode
- Amplitude
- Peak of top-to-top waarde



Figuur 8: Eigenschappen wisselspanning

3.3.3 Periodieke signalen

- 1 herhaling = 1 periode
- Periode (T) = tijdsduur (in s)
- Frequentie (f) = aantal periodes per seconde (in Hz)
- $F = \frac{1}{T} \Leftrightarrow T = \frac{1}{F}$

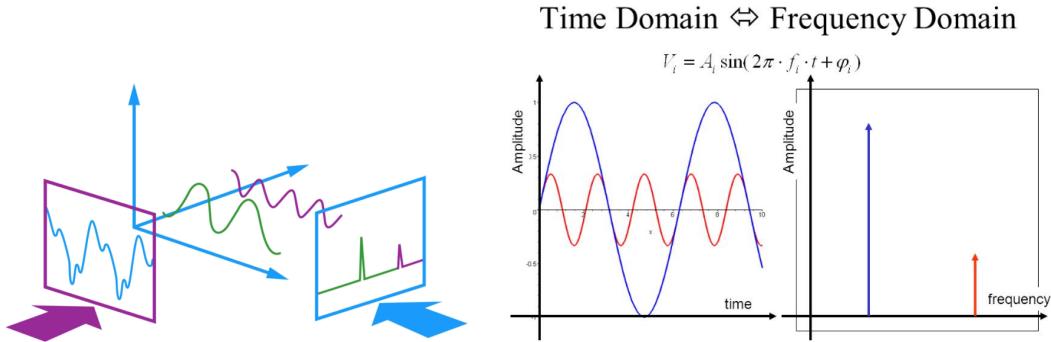


Figuur 9: Sinusgolf met periode T

3.3.4 Tijdsdomein en frequentiedomein

- Tijdsdomein: met een oscilloscoop

- Frequentiedomein: met spectraalanalyse



Figuur 10: Tijdsdomein vs Frequentiedomein

4 Digitale signalen

= Aan/uit

4.1 Eigenschappen

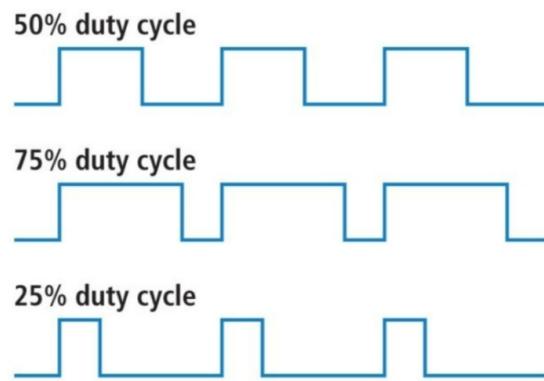
- Amplitude: Piek- of top-waarde, RMS-waarde
- Periode / frequentie
- Pulsbreedte
- Duty-cycle



Figuur 11: Eigenschappen digitaal signaal

4.2 Duty Cycle

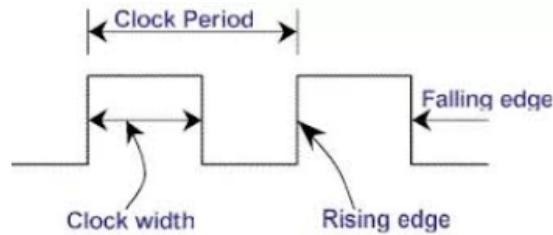
= Hoeveel procent van de tijd staat het signaal aan?



Figuur 12: Duty cycles

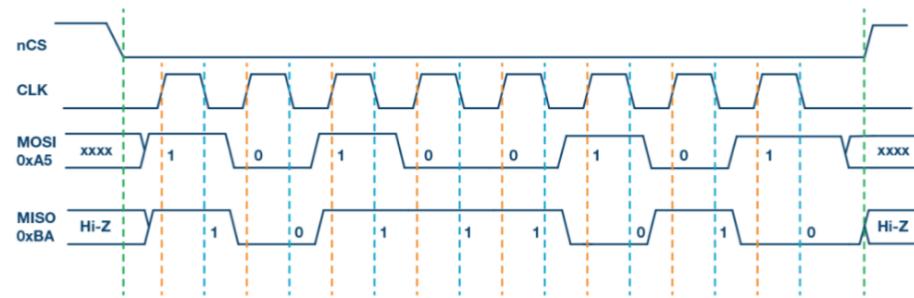
4.3 Flanken (edge)

- Stijgende flank
- Dalende flank
- Belangrijk bij kloksignalen



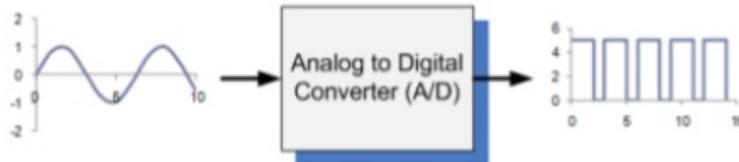
Figuur 13: Flanken

4.4 Weergave digitale signalen



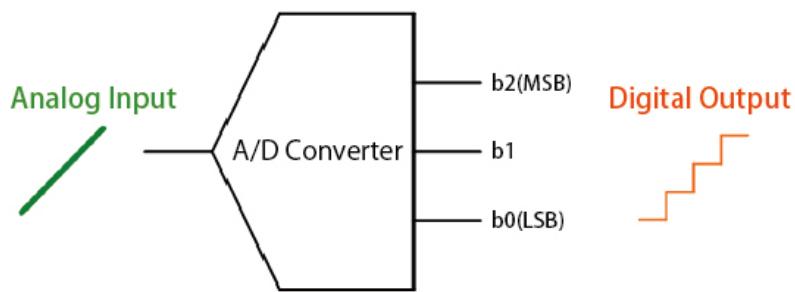
Figuur 14: Weergave digitale signalen

5 AD conversie



Figuur 15: Analoog naar digitaal (AD converter)

5.1 Analoog naar digitaal



Figuur 16: Analoog naar digitaal conversie

- **Range** = verschil tussen laagste en hoogste waarde
- **Resolutie** = aantal stappen of stapgrootte in bits
- **Belangrijk gevolg:**
 - beide parameters bepalen de exactheid en de afwijkingen

5.1.1 Voorbeeld A

- Range = 2V - 2.5V
- Resolutie = 8bits
- Dus aantal discrete stappen = $2^8 = 256 \Rightarrow 256 - 1 = 255$
- Stapgrootte (LSB) = $\frac{range}{255} = \frac{2.5V - 2V}{255} = \frac{0.5V}{255} = 0.00196..V/stap$
- ofwel $\approx 2mV/stap$

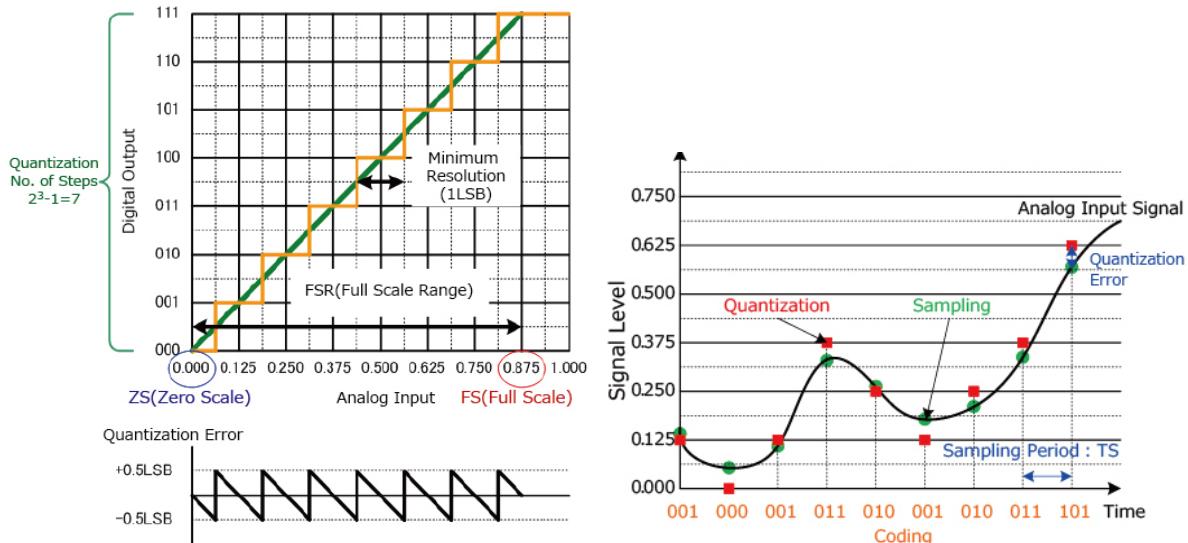
5.1.2 Voorbeeld B

- Range = 0V - 12V
- Resolutie = 12bits
- Dus aantal discrete stappen = $2^{12} = 4096 \Rightarrow 4096 - 1 = 4095$
- Stapgrootte (LSB) = $\frac{range}{4095} = \frac{12V - 0V}{4095} = \frac{12V}{4095} = 0.0029304..V/stap$

- ofwel $\approx 3mV/\text{stap}$

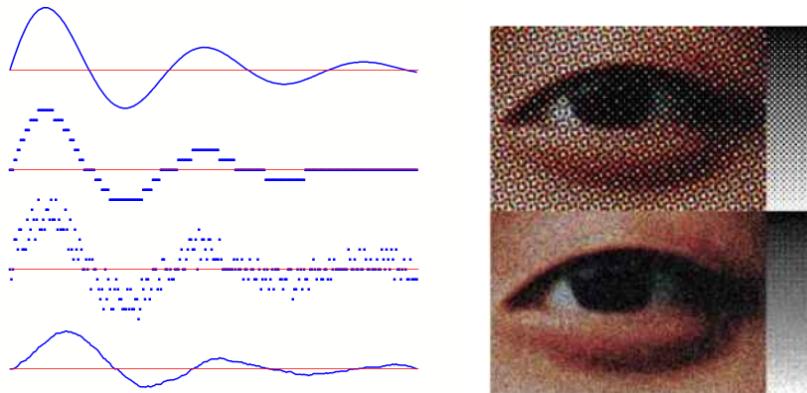
5.2 Eigenschappen

5.2.1 Quantisatiefouten



Figuur 17: Quantisatiefouten door AD conversie

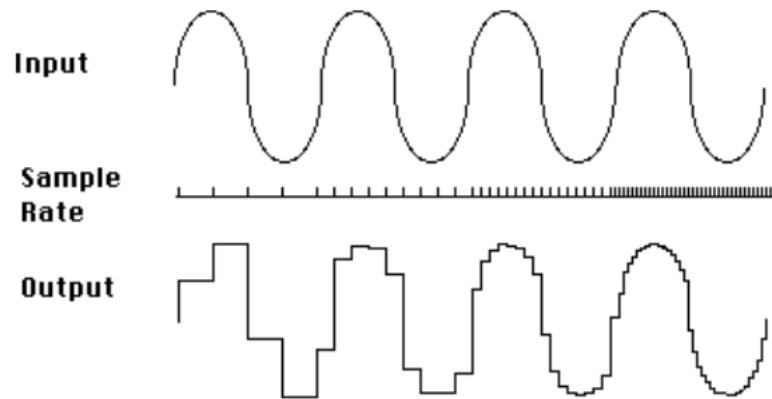
- Veroorzaakt quantisatieruis
- Quantisatiefouten \rightarrow dithering
- = vooraf (witte) ruis toevoegen aan signaal



Figuur 18: Dithering

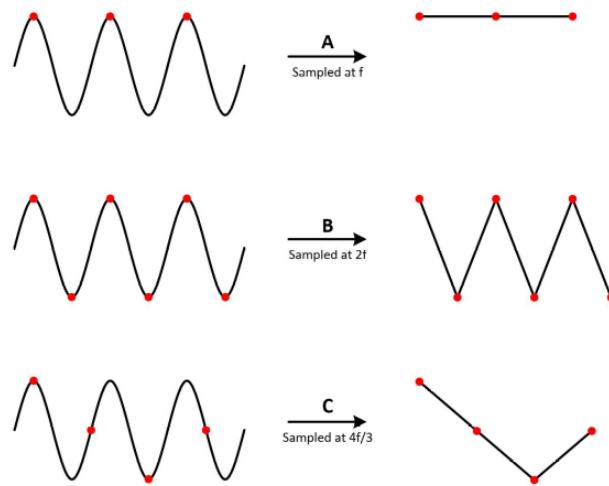
5.2.2 Sample Rate / sample frequentie

= aantal conversies per seconde



Figuur 19: Sample rate

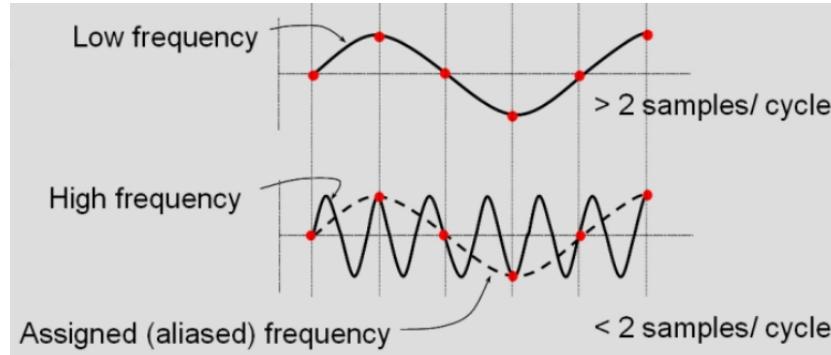
- **Nyquist rate** = Minimale sample rate = 2x de frequentie van het signaal
- Voorbeelden:
 - HiFi Audio CD: 44.1kHz sample rate
 - Oude telefoon toestellen: 8kHz sample rate
 - HD-DVD Audio: 192kHz



Figuur 20: Minimale sample rate

5.2.3 Aliasing

- = High Frequency signaal als Low Frequency 'spooksignaal' detecteren
 - Treedt op bij onvoldoende hoge sample rate
 - Anti-Aliasing filter (low-pass filter) beperkt signaal onder nyquist frequentie



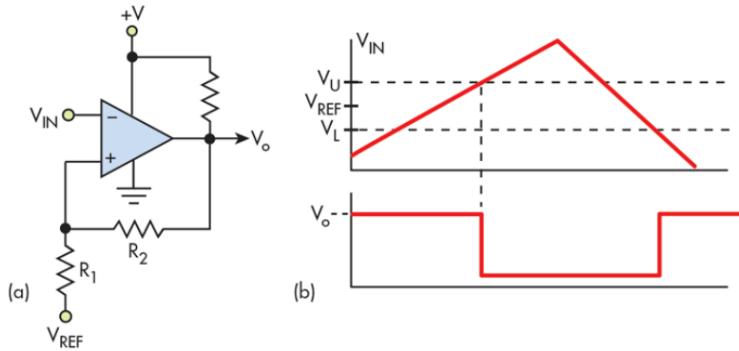
Figuur 21: Anti-aliasing filter

5.2.4 Oversampling

- Sampelen met veelvoud van nyquist frequentie
- Kan worden gebruikt om de resolutie op te voeren
- Kan worden gebruikt om digitaal (DSP) te filteren
- Verhoogt het effectieve aantal bits van de ADC
 - Voorbeeld: 20bit ADC met 256x OS = 24bit effectieve resolutie
- Undersampling → specifiek gebruik bij mixers

5.3 Implementatie en types

- De comparator
- Bekeken als 1-bit ADC

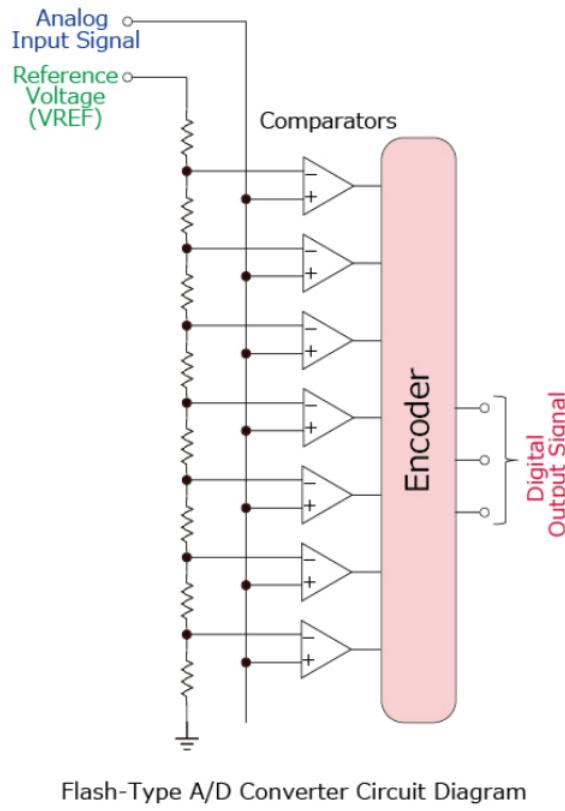


Figuur 22: Comparator

5.3.1 Flash ADC

- Comparator per 'level'
- Zeer snel = directe omzetting
- Complex & High power

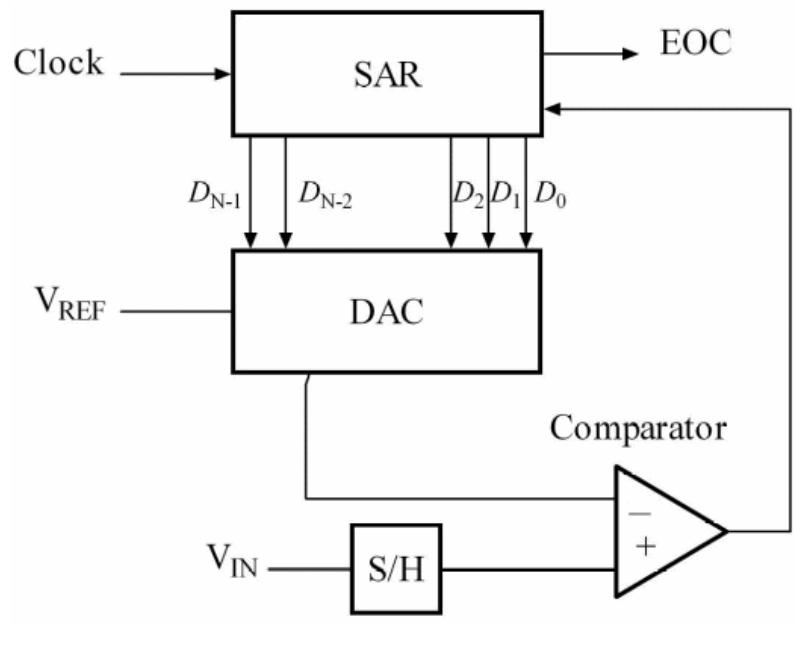
- Lagere resoluties



Figuur 23: Flash ADC

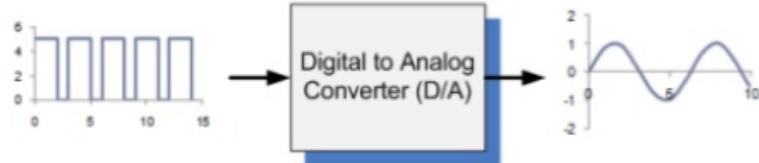
5.3.2 Successive approximation ADC

- Gebruikt 1 comparator
- Vergelijkt een opgewekte spanning met het signaal
- Hoge resolutie mogelijk
- Trager
- Relatief goedkoop



Figuur 24: Successive approximation ADC

6 Digitaal naar analoog conversie

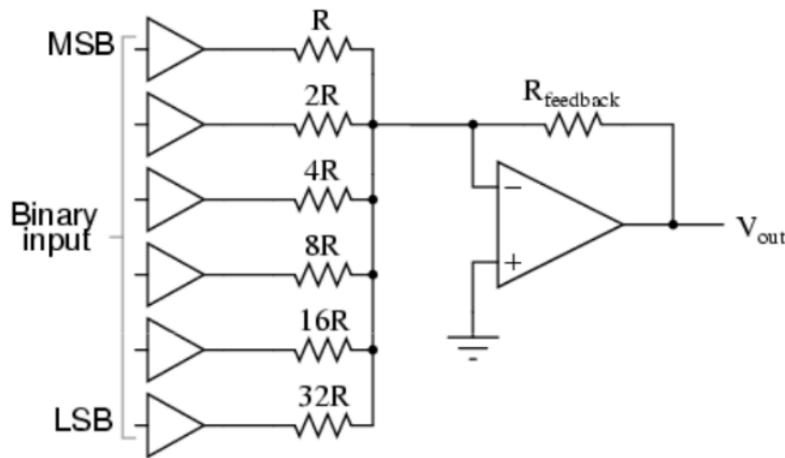


Figuur 25: Digitaal naar analoog (DA converter)

- Omzetten digitale naar analoge waarde
- Range
- Resolutie
- Samplefrequentie

6.1 Simple DAC

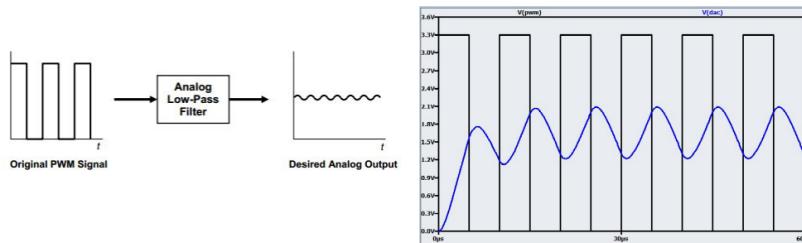
- Binair \Rightarrow analoge waarde
- Voorbeeld: weerstandsnetwerk



Figuur 26: Weerstandsnetwerk

6.2 Simpele DAC met PWM

- PWM == digitaal signaal
- Door variatie van duty-cycle kan de gemiddelde waarde worden gevarieerd
- Door filteren kan de blokgolf worden omgezet in een variabele analoge waarde



Figuur 27: DAC met PWM

6.3 Andere types

- $\Sigma\Delta$ (sigma delta) = herhaaldelijk downsampelen
- I^2S DAC
- Nog zeer veel andere overwegingen:
 - THD (Harmonische vervorming)
 - Faseruis
 - ...

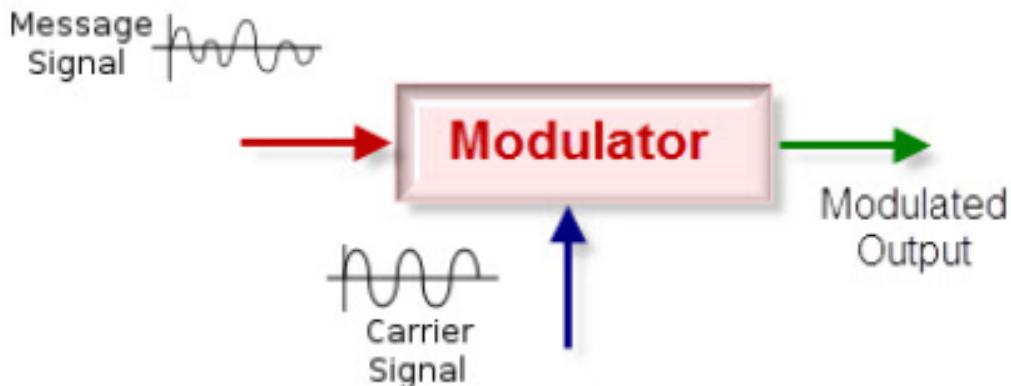
6.4 Extra info over AD/DA conversie:

- https://en.wikipedia.org/wiki/Digital-to-analog_converter

- https://en.wikipedia.org/wiki/Analog-to-digital_converter
- https://en.wikipedia.org/wiki/Nyquist%20%26%20Shannon_sampling_theorem
- https://en.wikipedia.org/wiki/Nyquist_rate
- https://en.wikipedia.org/wiki/Pulse-width_modulation
- https://en.wikipedia.org/wiki/Delta-sigma_modulation

7 Modulatie

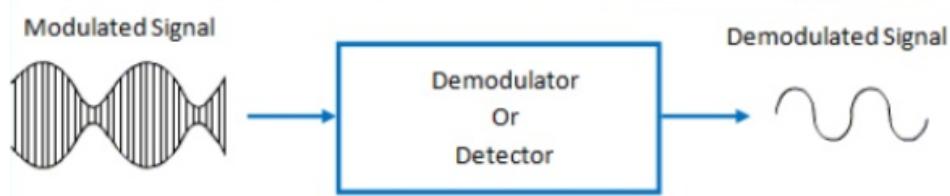
- Informatie toevoegen aan een draaggolf
- Door de variatie van minstens een van de eigenschappen van deze draaggolf



Figuur 28: Modulatie

7.1 Demodulatie

= Het terugwinnen van de informatie uit de gemoduleerde draaggolf



Figuur 29: Demodulatie

7.2 Modem

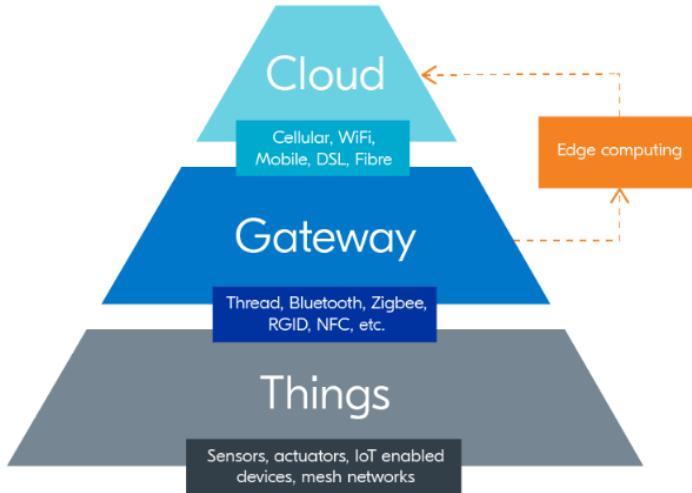
= **Modulator + Demodulator**



Figuur 30: Modem

7.3 Waarom?

- Interconnectie van IoT devices
- Vaak draadloos



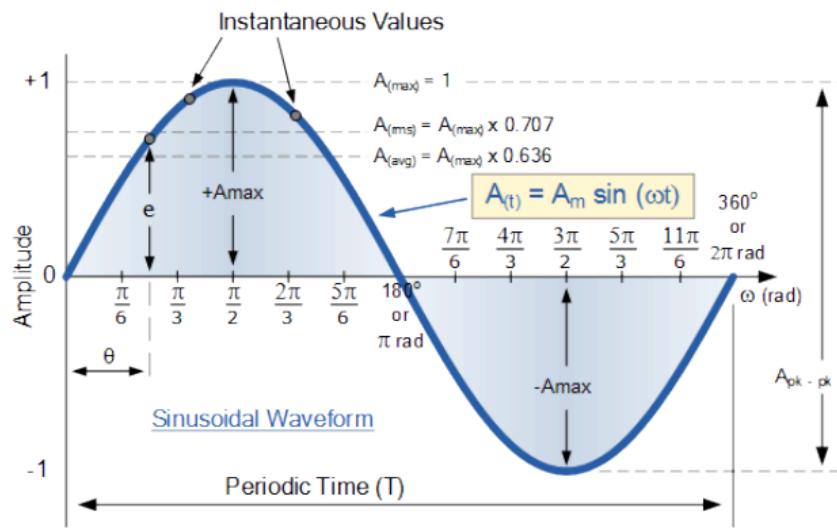
Figuur 31: Interconnectie van devices

7.4 Hoe?

- Draaggolf of carrier
- Signaal met een zekere (hogere) frequentie

7.5 Draaggolf

= carrier



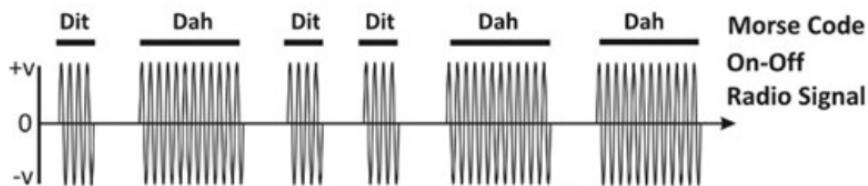
Figuur 32: Draaggolf

7.5.1 Parameters van een draaggolf

- Amplitude
- Frequentie / Periode

7.6 Simpele modulatie

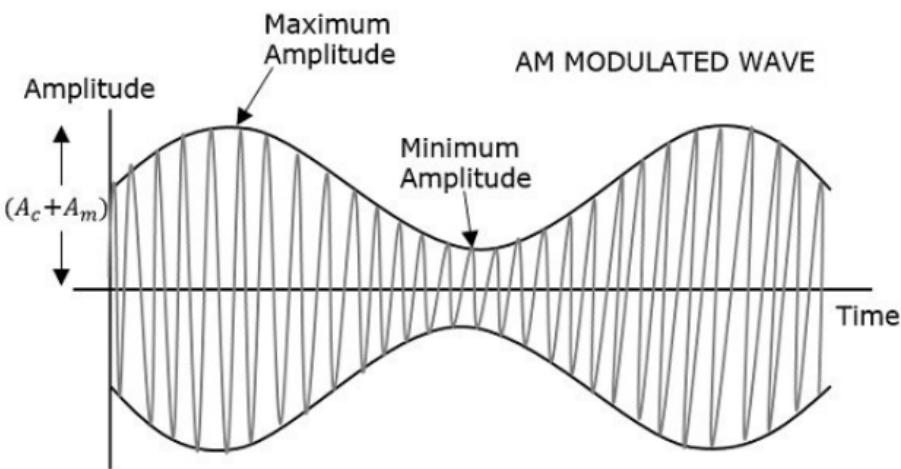
- Aan/uit schakelen van de draaggolf
- CW = continuous wave
- Bv: Morse code



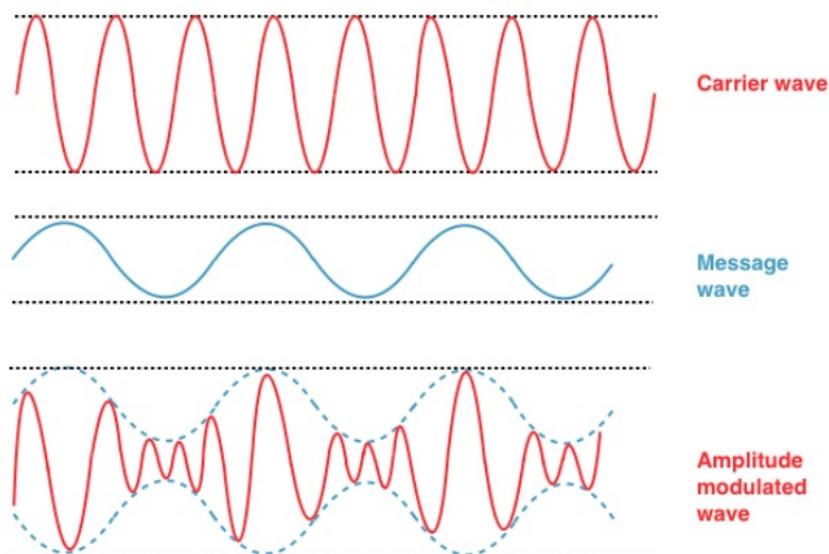
Figuur 33: Morse code

7.7 Amplitude modulatie (=AM)

- Aanpassen van de amplitude v/d draaggolf



Figuur 34: Amplitudemodulatie (1)

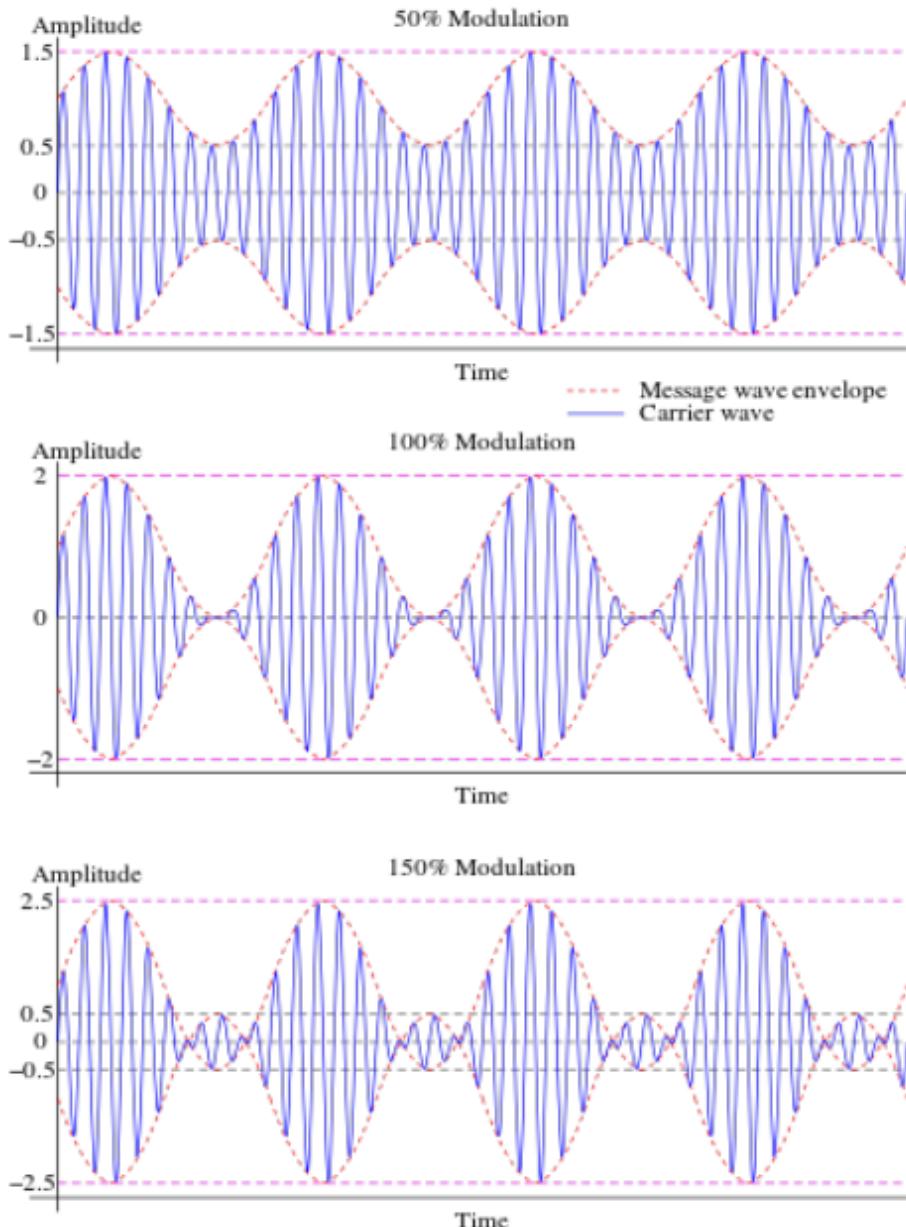


Figuur 35: Amplitudemodulatie (2)

- Radio LW/MW/SW AM
- Typisch gebruikt op 'lagere' HF banden: 100kHz - 60MHz

7.7.1 Overmodulatie

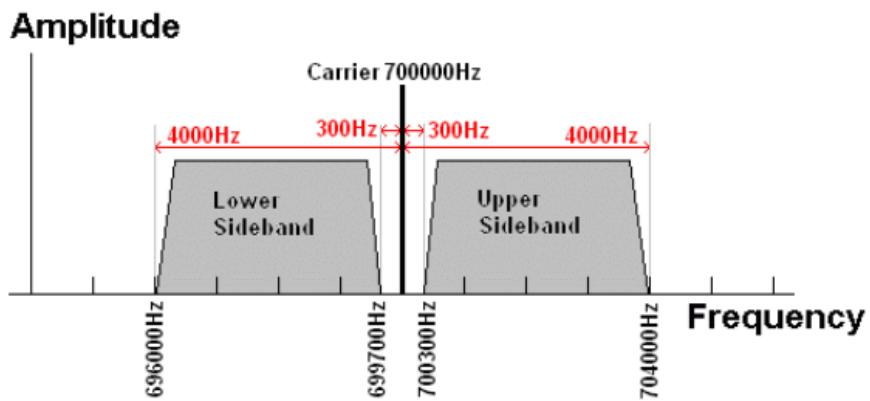
- Modulatiediepte (modulatie index)



Figuur 36: Overmodulatie

7.7.2 AM bandbreedte

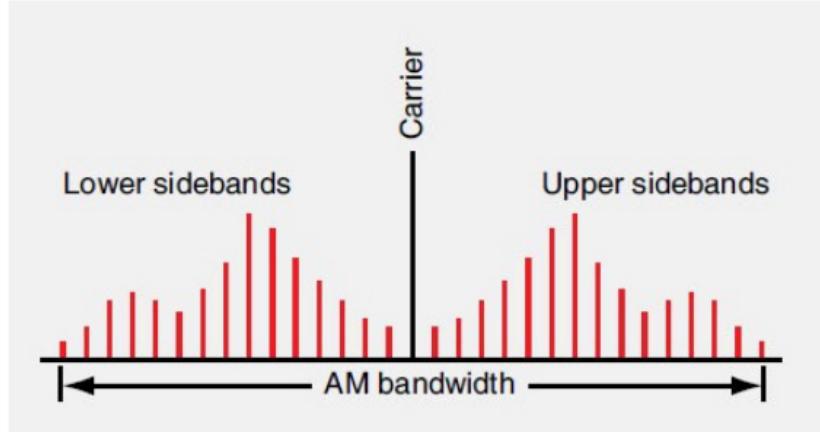
- Bandbreedte van een AM signaal:
 - Centerfrequentie = draaggolf frequentie
 - 2x frequentie van gemoduleerd signaal in totaal



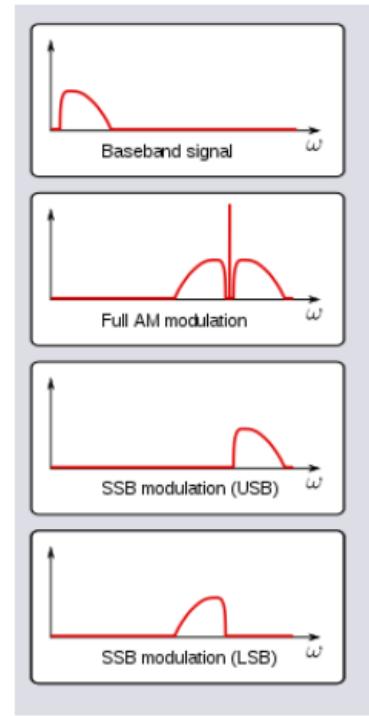
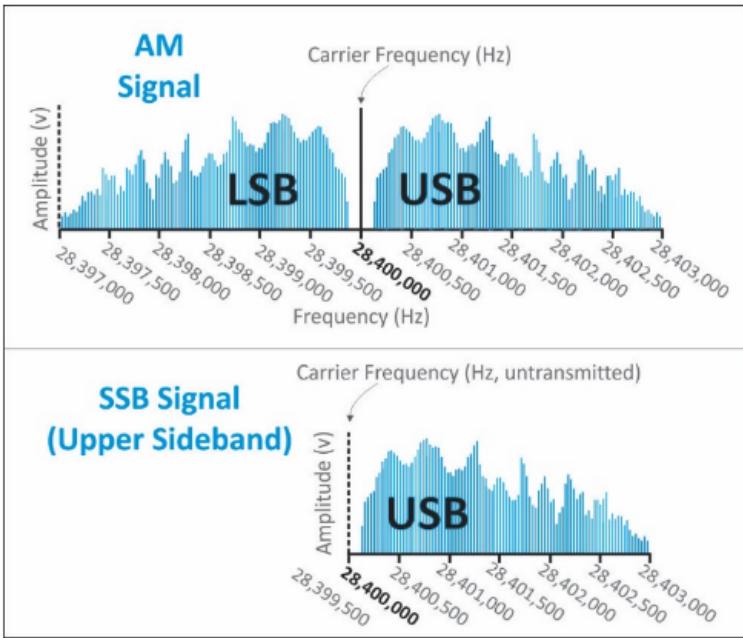
Figuur 37: AM bandbreedte

7.7.3 SSB modulatie (USB/LSB)

- Alle informatie zit in elke sideband (zijband) bij AM
- Carrier + 1 sideband wegfilteren = reductie van de bandbreedte
- Efficienter gebruik van het spectrum
- Moeilijk om goed te demoduleren



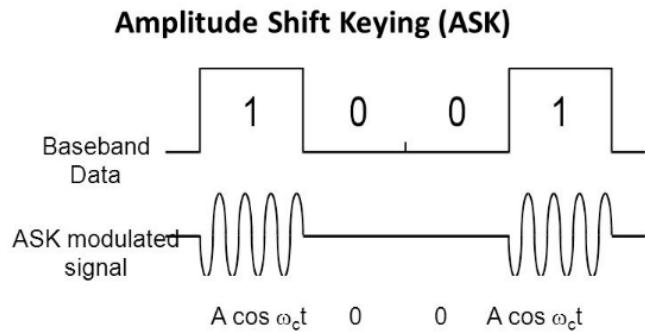
Figuur 38: Zijbanden AM band



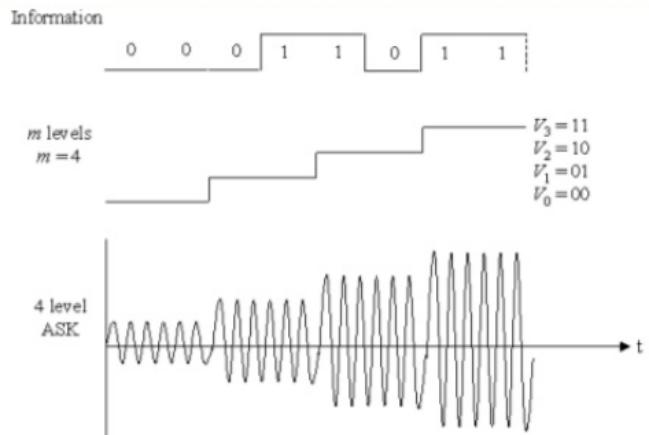
Figuur 39: SSB Modulatie: we zenden alleen 1 van de zijbanden

7.7.4 ASK Modulatie

- Amplitude shift keying
- Vorm van AM-modulatie voor digitale signalen
- Mogelijk met meerdere signaalniveaus
- 4-level ASK



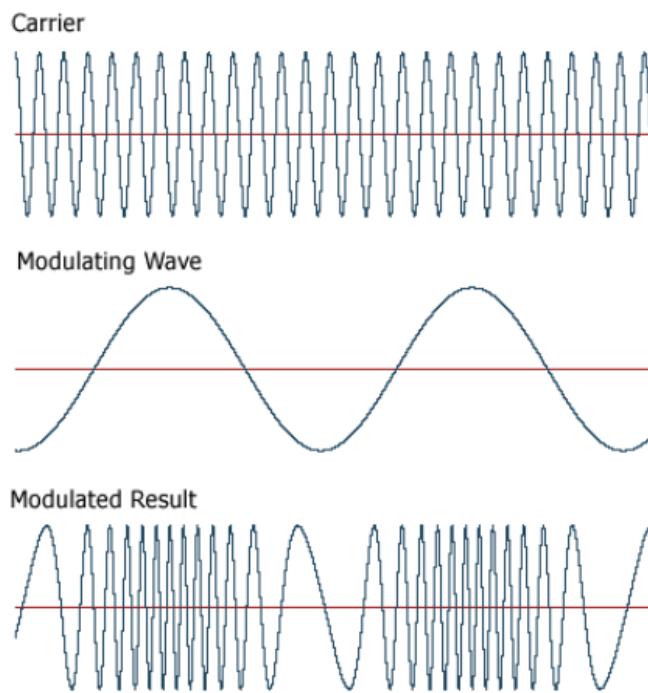
Figuur 40: ASK



Figuur 41: 4 level ASK

7.8 Frequentie modulatie (=FM)

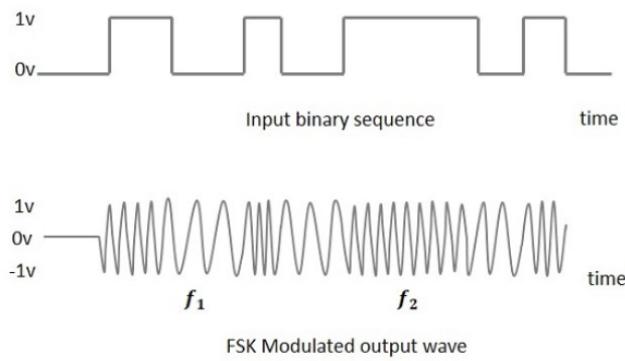
- Variatie in de frequentie van de draaggolf
- FM radio 88MHz - 108MHz
- VHF Maritieme radio
- UHF PMR Radios



Figuur 42: Frequentie modulatie

7.8.1 FSK modulatie

- Frequency Shift Keying
- Vorm van FM
- Wisselen tussen 2 of meer frequenties

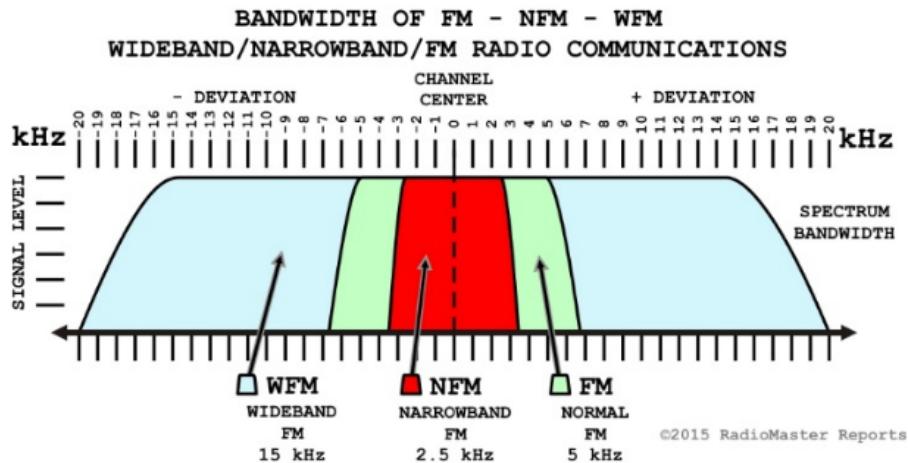


Figuur 43: FSK

7.8.2 FM Modulatie

- Carrier is altijd op 100% amplitude aanwezig tijdens transmissie

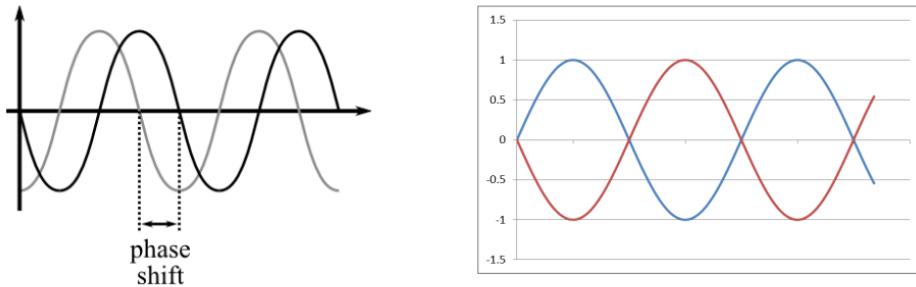
- Minder ruis, betere kwaliteit voor audio dan AM
- Hogere bandbreedte \Rightarrow Meer stroomverbruik
- WFM (Wide FM), NFM (Narrow FM), FM



Figuur 44: Bandbreedte FM - NFM - WFM

7.8.3 Fase modulatie

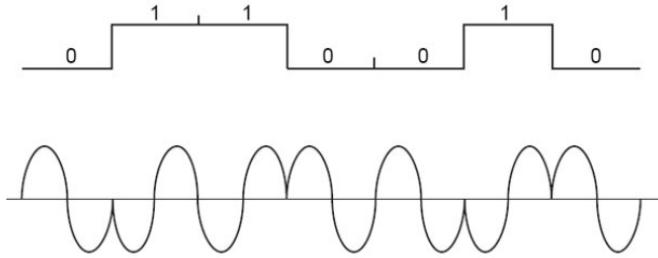
- Faseverschuiving van een signaal



Figuur 45: Faseverschuiving

7.8.4 PSK modulatie

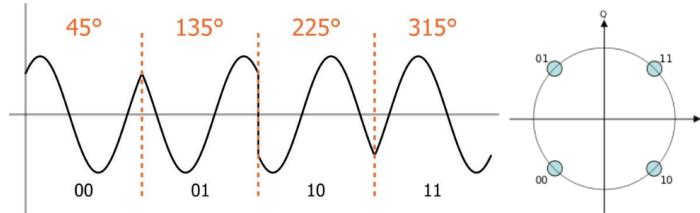
- Phase shift keying
- Bij wisselen van bit \Rightarrow fase omkeren



Figuur 46: Fasemodulatie

7.8.5 Phase Shift Modulatie

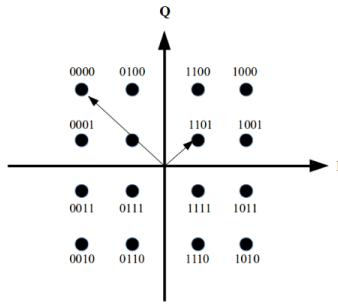
- BPSK \Rightarrow Binary PSK = 2 fase
- QPSK \Rightarrow Quadrature PSK = 4 fase



Figuur 47

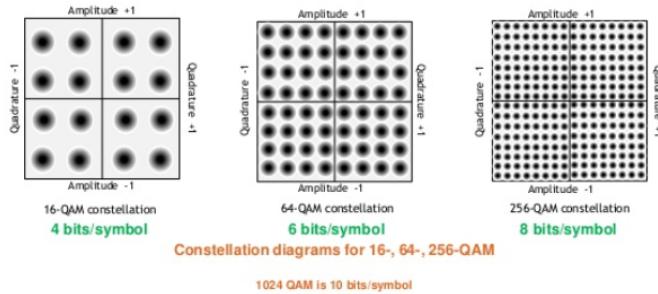
7.8.6 QAM modulatie

- Quadrature amplitude modulatie
- Informatie zit ...
 - de amplitude (zoals bij ASK)
 - de fase (zoals bij PSK)
- Meerdere symbolen
 - 4-QAM
 - 16-QAM
 - 64-QAM
 - ...



Figuur 48: QAM modulatie

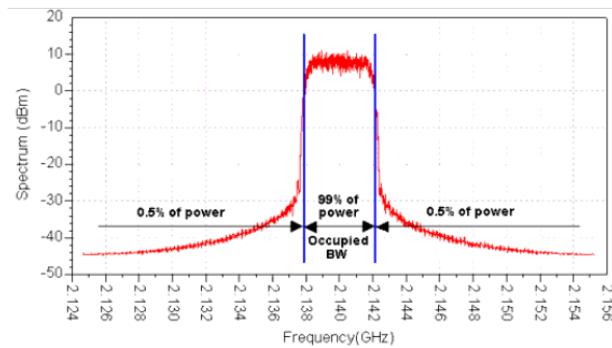
- DAB+ \Rightarrow 4/16/64 - QAM
- Hogere transmissiesnelheid
- Gevoeliger voor fouten



Figuur 49: 16-, 64-, 256-QAM

7.8.7 Bandbreedte / vermogen

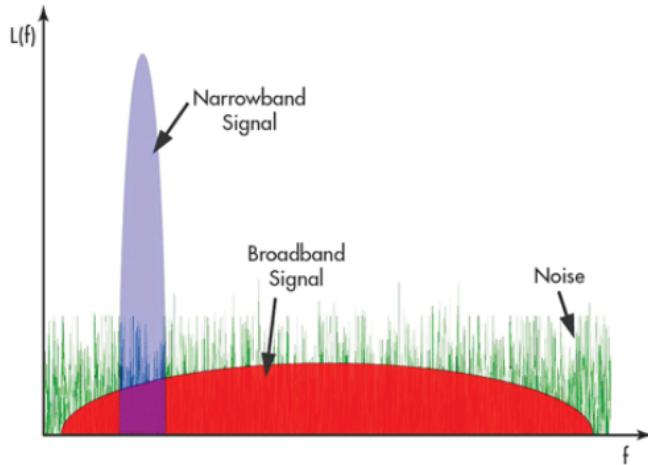
- Meer bandbreedte = hogere snelheid
- Meer bandbreedte = hoger vermogen nodig



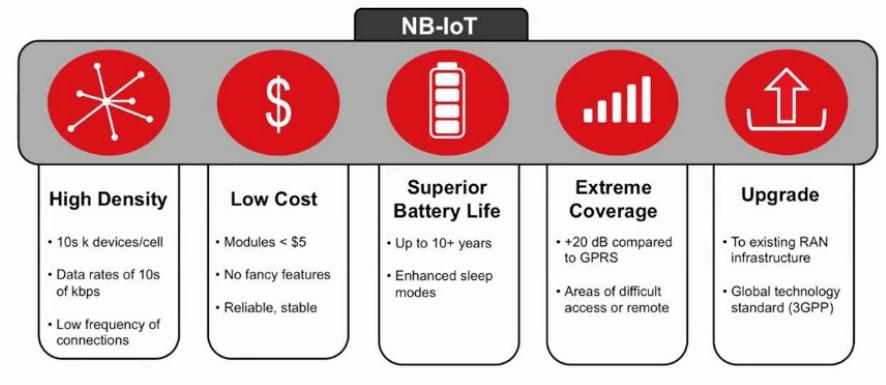
Figuur 50

7.8.8 Waarom Narrow-band?

- Oppervlakte van het signaal = Power
- Bij smalbandige signalen \Rightarrow betere SNR (signaal-naar-ruis) verhouding bij hetzelfde vermogen
- Om met een klein vermogen heel grote afstanden overbruggen
- Zeer traag



Figuur 51: Waarom Narrow-band: narrow-band kan makkelijker boven ruisniveau

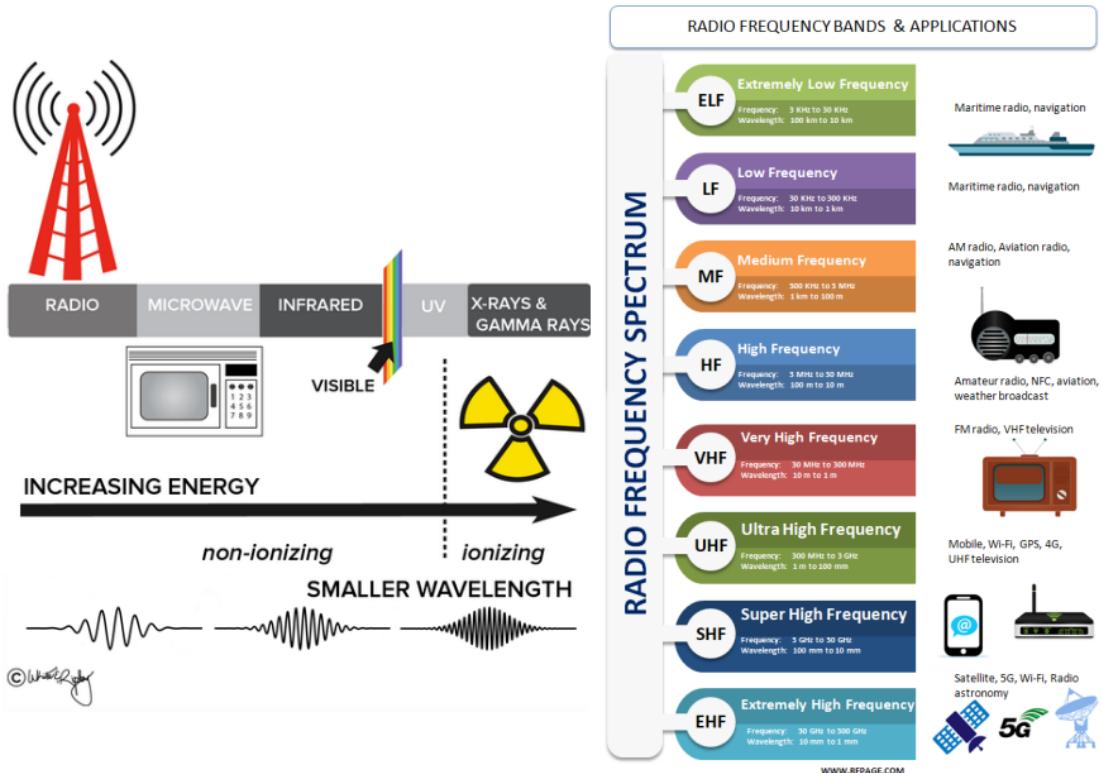


Figuur 52: Voorbeeld: NB-IoT



Figuur 53: Voorbeeld: Telefoonmodem (<https://www.youtube.com/watch?v=ckc6XSSh52w>)

8 RF-spectrum



Figuur 54: Radiogolven op het spectrum

8.1 Beschikbare bandbreedte

- In de hogere frequentiebanden is meer bandbreedte beschikbaar
- Voorbeeld:
 - 2.4GHz WiFi: 20MHz bandbreedte
 - 5GHZ 802.11ac WiFi: 80MHz bandbreedte
 - 1 WiFi kanaal > volledige AM-radio LW + MW + SW band

- Meer bandbreedte = mogelijk hogere datarates

Number of Streams	Bitrate	Channel Width		
		20 MHz	40 MHz	80 MHz
1 stream	87 Mbps	200 Mbps	433 Mbps	
2 streams	173 Mbps	400 Mbps	866 Mbps	
3 streams	289 Mbps	600 Mbps	1300 Mbps	

Figuur 55: Impact van hogere bandbreedtes op datarates

8.2 Licenties en licentievrije banden

- Beheer van het spectrum
- In Belgie door het BIPT
- In de VS door het FCC
- Harmonisatie in Europa / VS / Azië
- <https://www.bipt.be/nl/operatoren/radio/frequentiebeheer/frequentieplan/tabel>

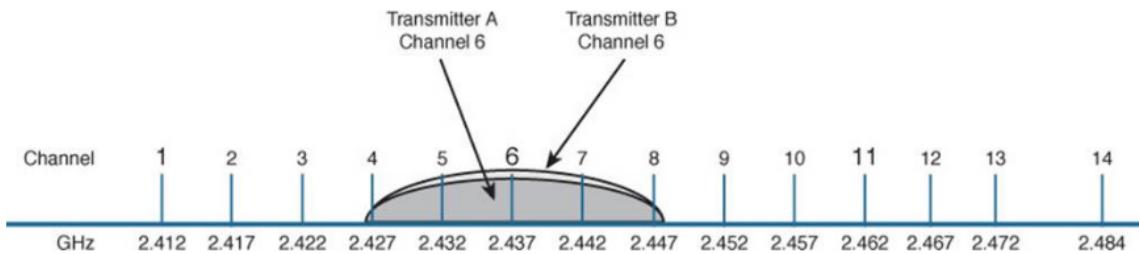
8.3 ISM banden

- = Industrial / Scientific / Medical - Band
- Geen vergunning nodig voor gebruik
- Is aan strikte voorwaarden verbonden
 - Maximaal vermogen
 - Maximale transmissietijd
 - Maximale periodiciteit van transmissies
 - Maximale bandbreedte
 - ...
- Specifieke banden zijn hiervoor vrijgegeven

8.4 Interferentie

8.4.1 Co-channel interferentie

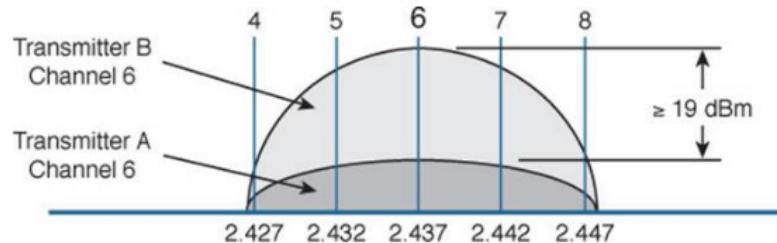
Wat als meerdere systemen dezelfde band gebruiken?



Figuur 56: Co-channel Interferentie

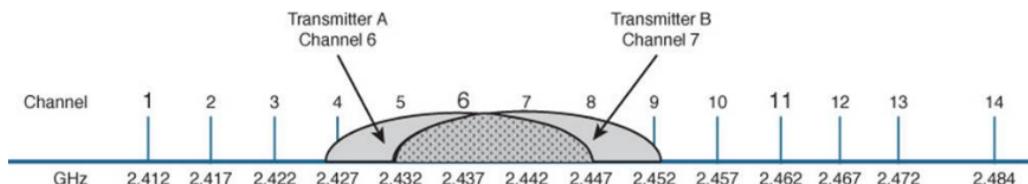
8.4.2 Signaalseparatie

- Minimaal verschil is afhankelijk van modulatietype
- Voorbeeld:
 - BPSK > 10dB
 - 256-QAM > 40dB



Figuur 57: Signaalseparatie

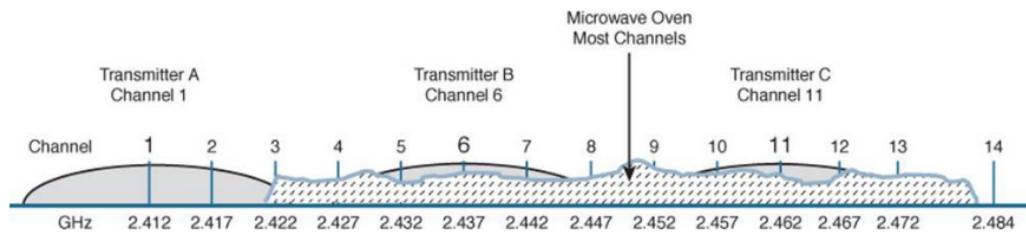
8.4.3 Interferentie van naburige kanalen



Figuur 58: Interferentie van naburige kanalen

8.4.4 Interferentie van andere toestellen

- Ruis / atmosferische storingen
- SNR \Rightarrow Signaal / Ruis verhouding



Figuur 59: Interferentie van andere toestellen

8.5 Meten van interferentie

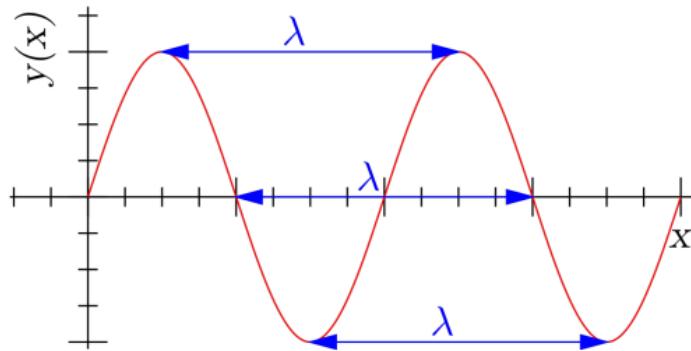
- Meten van WiFi kanalen kan met software
- Software is niet in staat niet-WiFi signalen te meten
- Kan worden gemeten met een spectrum analyzer



Figuur 60: Meten van interferentie

8.6 Golvlengte

- Golvlengte in meter is een andere wijze om de frequentie weer te geven
- Relevant bij antennes en transmissielijnen
- $\lambda = \frac{c}{f}$
- c = lichtsnelheid = 299 792 458 m/s



Figuur 61: Golflengte λ

8.6.1 Rekenvoorbeeld

Frequentie van 102.1 MHz

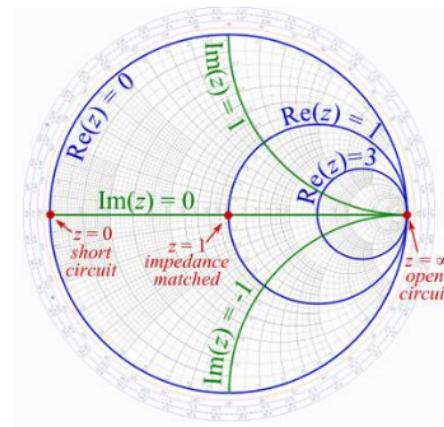
- $\lambda = \frac{299\ 792\ 458 \text{ m/s}}{102.1 \text{ MHz}} = 2.936 \dots \text{ m}$

”De 70 cm band”:

- $f = \frac{(299\ 792\ 458)}{\lambda} = \frac{(299\ 792\ 458)}{0.7 \text{ m}} = 428\ 274\ 940 \text{ Hz} = 430 \text{ MHz}$

8.7 Antennesysteem

- Een antenne stuurt / ontvangt zoveel mogelijk van de RF-energie op de gewenste frequentie
- \Rightarrow resonantie van het antennesysteem op de gewenste frequentie
- Formaat van de antenne is vaak beperkende factor

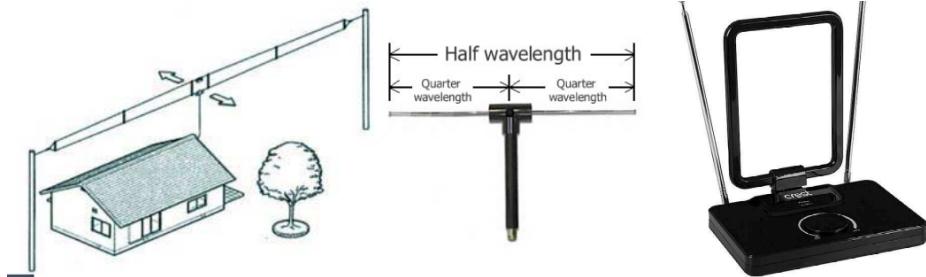


Figuur 62

8.7.1 De dipool antenne

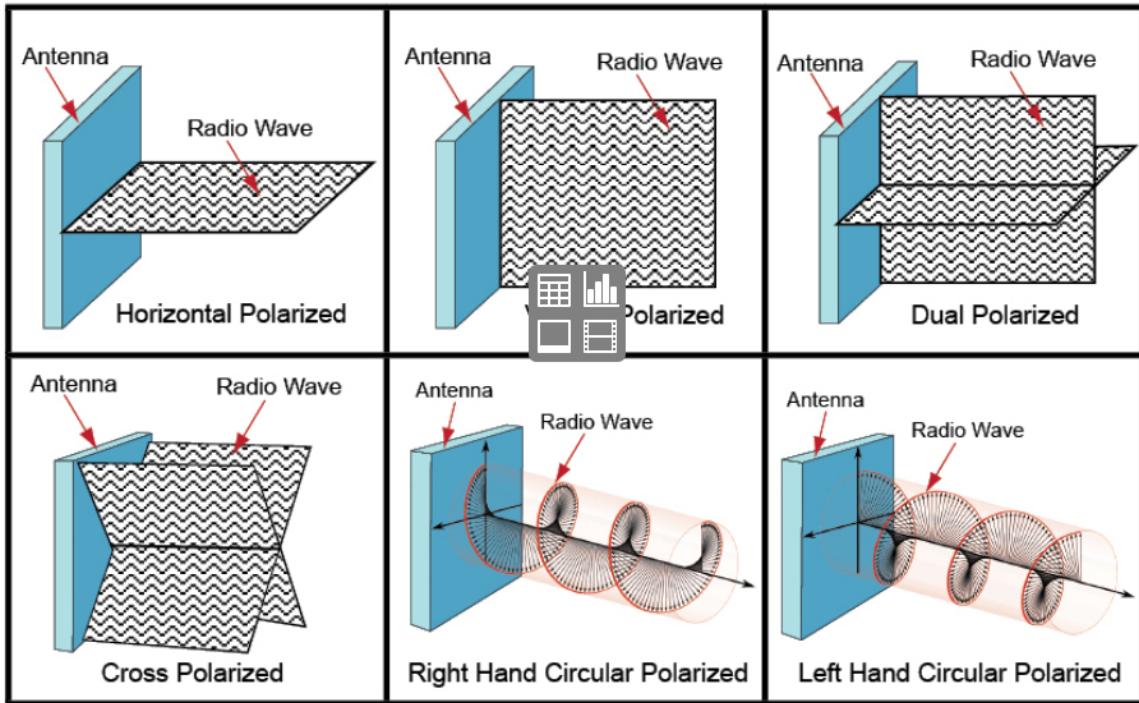
- Zeer eenvoudig
- $\frac{1}{2}\lambda$ groot

- Vaak toegepast
- Kan worden 'opgeplooid'



Figuur 63: Dipool antenne

8.7.2 Polarisatie van antennes

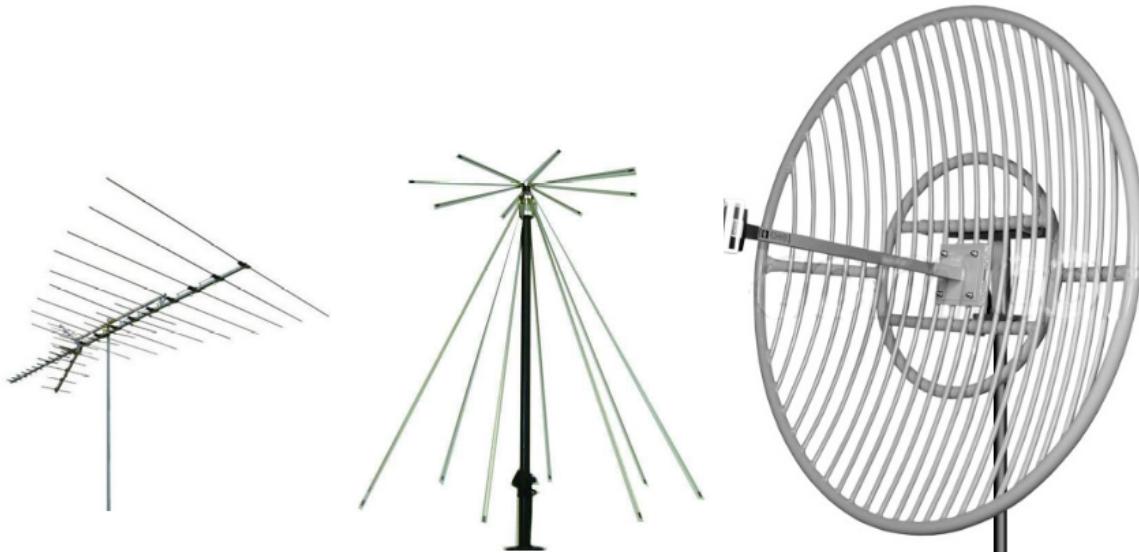


Figuur 64: Polarisatie van antennes

8.7.3 Directionaliteit van antennes

- Omnidirectionele antenne
 - Meestal dipolen of end-fed (=stukje draad)
 - Discone
- Directionele antenne (beam)
 - Schotelantenne

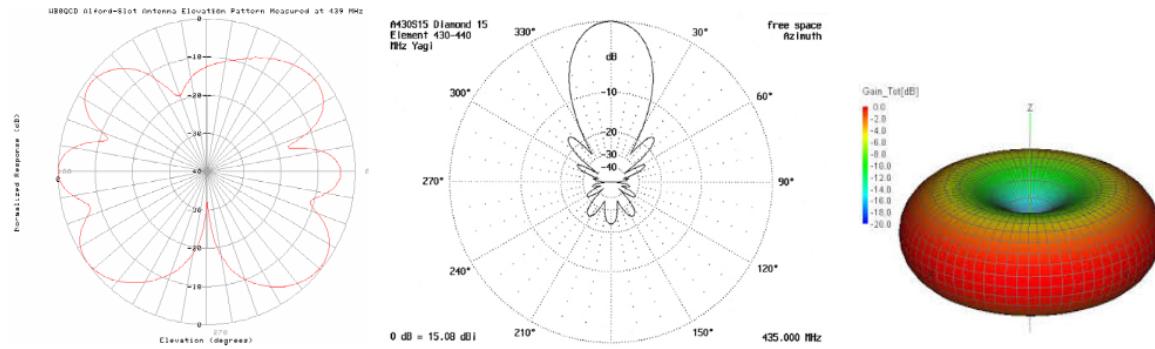
- Yagi
- Patch



Figuur 65: Directionaliteit van antennes

8.7.4 Stralingspatroon van antennes

- Gevoeligheid van antennes is niet overal hetzelfde
- Zeer afhankelijk van constructie en type antenne

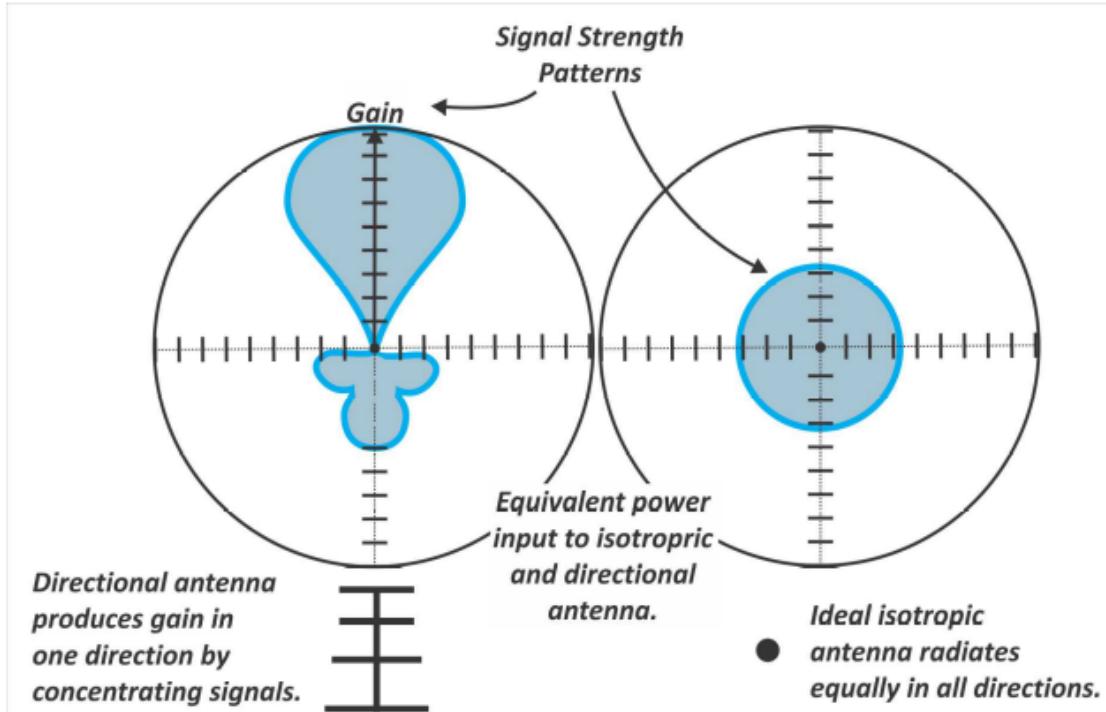


Figuur 66: Stralingspatroon van antennes

8.7.5 Gain van een antenne

- Gain = versterking
- Uitgedrukt in dB
- Nooit 'magisch', er is altijd een trade-off
 - Directionaliteit

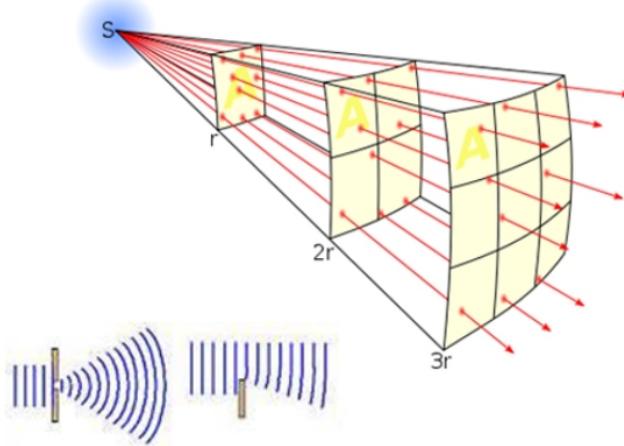
- Bandbreedte



Figuur 67: Gain van een antenne

8.8 Propagatie van RF-signalen

- Absorptie
- Reflectie
- Scattering
- Refractie
- Path loss
- ...



Figuur 68: Propagatie van RF-signalen

8.8.1 Path loss

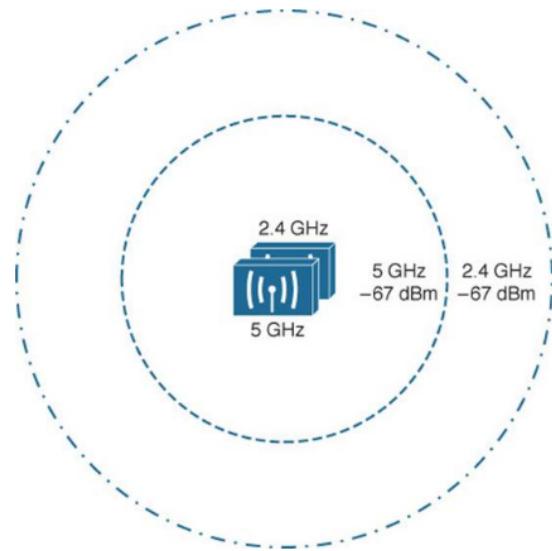
- Verzwakking van het signaal met afstand zonder obstakels \Rightarrow free space path loss (FSPL)
- **Oorzaak** uitdeinen van het signaal
- Verzwakking is exponentieel met de afstand
- Enkel afhankelijk van frequentie en afstand
- Hogere frequenties hebben hogere path loss

$$\text{FSPL (dB)} = 20 \cdot \log_{10}(d) + 20 \cdot \log_{10}(f) + 32.44$$

- d = afstand in km
- f = frequentie in MHz

8.8.2 Signaalverzwakking / path loss

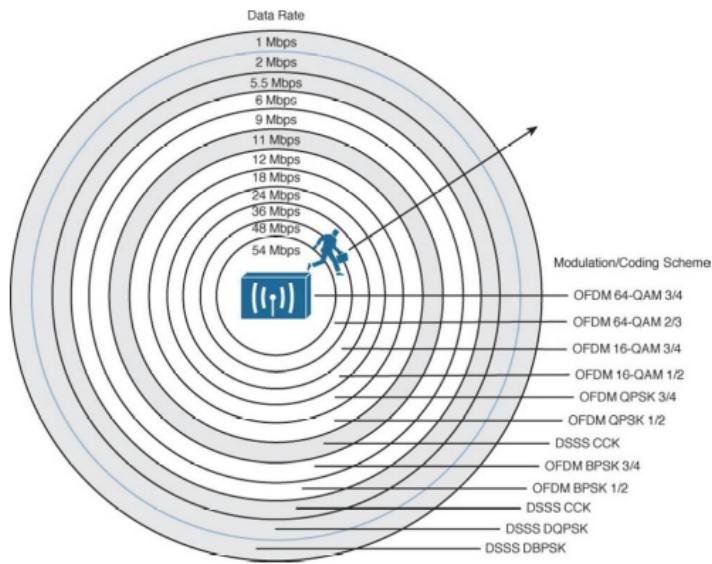
- Meer verzwakking bij hogere frequenties
- Minder bandbreedte bij lagere frequenties
- Wat is beter?
 - 2.4GHz WiFi
 - 5GHz WiFi



Figuur 69: Signaalverzwakking / path loss

8.8.3 Dynamic Rate Shifting (DRS)

- Modulatietechniek dynamisch aanpassen aan de signaalcondities
- Minder gunstige SNR \Rightarrow Lagere datarate kiezen
- Andere modulatievorm

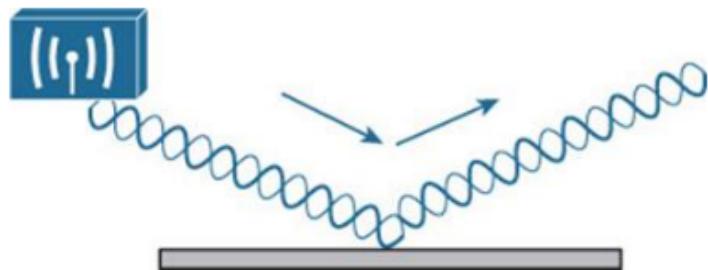


Figuur 70: Dynamic Rate Shifting

8.8.4 Reflectie

- Signaal wordt gereflecteerd

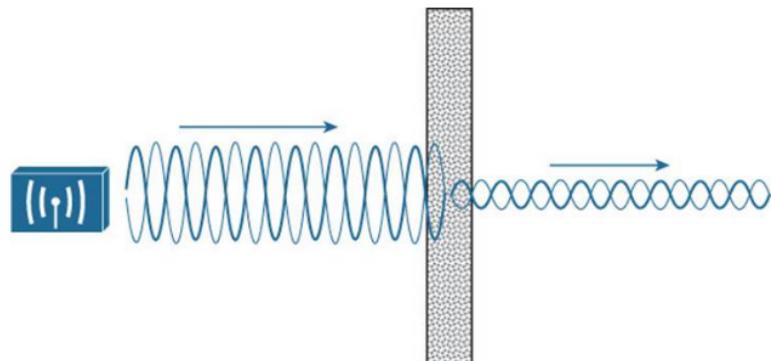
- Bv:
 - Door metalen objecten
 - Water
 - ...



Figuur 71: Reflectie

8.8.5 Absorptie

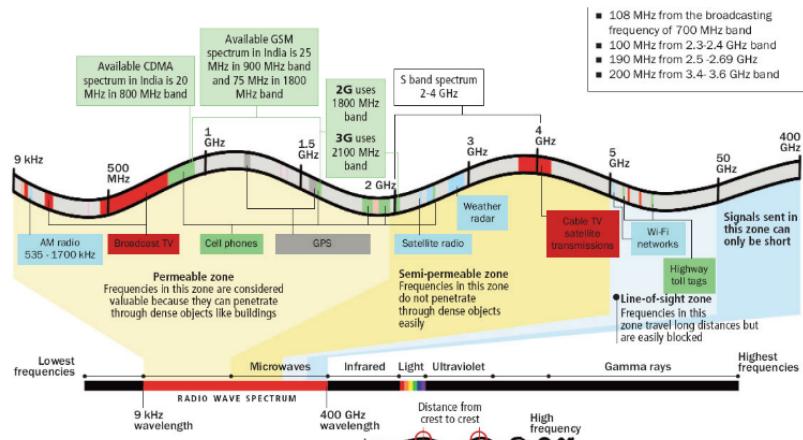
- Signaal kan deels of volledig worden geabsorbeerd
- Resulteert in verzwakking van bruikbare signaal
- Verzwakking = attenuatie typisch in dB



Figuur 72: Absorptie

8.8.6 Doordringbaarheid

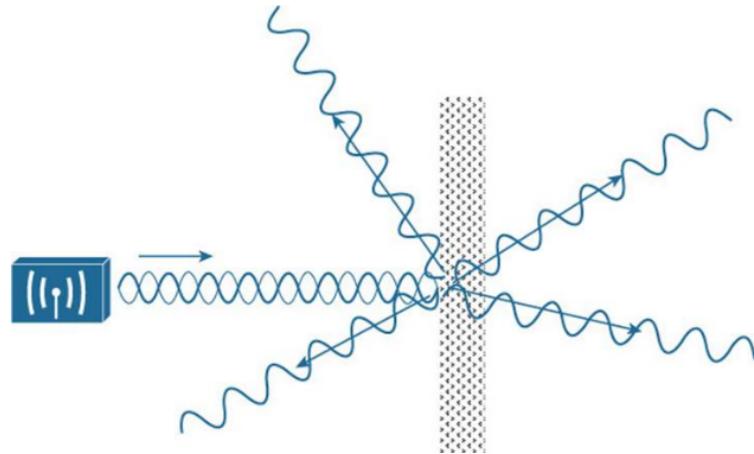
- Hogere frequenties worden gemakkelijk tegengehouden door objecten
- Lagere frequenties kunnen gemakkelijker door objecten heen



Figuur 73: Doordringbaarheid

8.8.7 Scattering

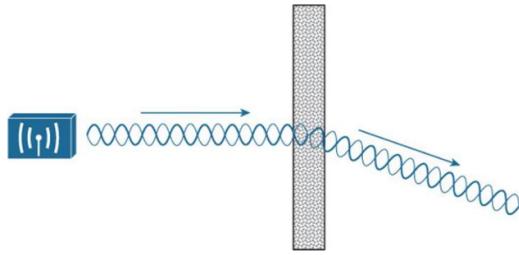
- Signaal wordt gereflecteerd in diverse richtingen
- Oneffen oppervlaktes
- Wolken materiaal bv zand, ...



Figuur 74: Scattering

8.8.8 Refractie

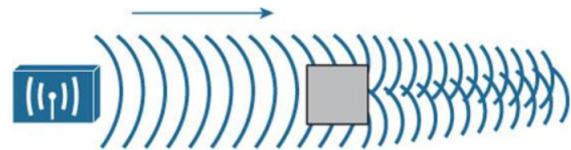
- Bij 2 media met verschillende dichtheid
- Afbuiging van het signaal



Figuur 75: Refractie

8.8.9 Diffractie

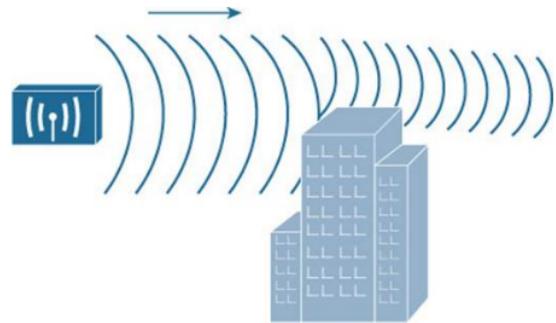
- RF-signalen worden beïnvloed door obstakels
- Gaat er niet door maar “rond”, zoals water rond een paaltje in een rivier
- Verstoort / vermindert het RF-signalen



Figuur 76: Diffractie

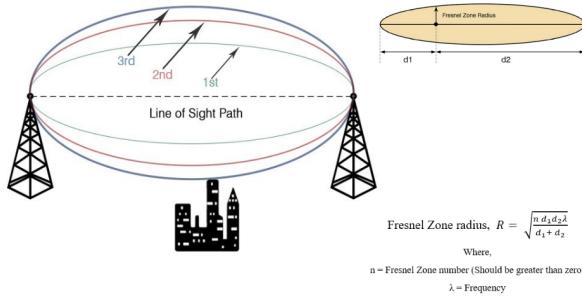
8.8.10 Fresnel zones

- Golfmodel RF-signalen
- Worden ook beïnvloed door objecten in nabijheid
- Niet enkel door objecten in line-of-sight



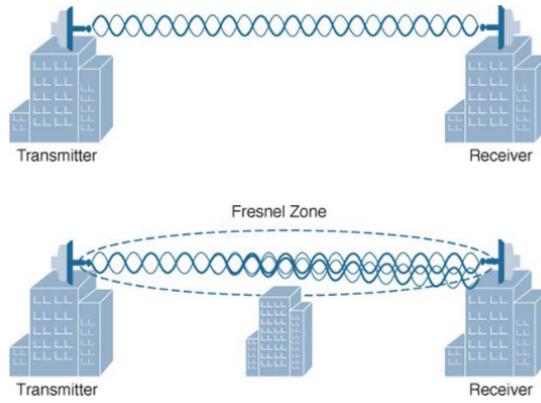
Figuur 77: Fresnel zones

- Afhankelijk van frequentie en afstand



Figuur 78: Fresnel zone radius is afhankelijk van frequentie en afstand

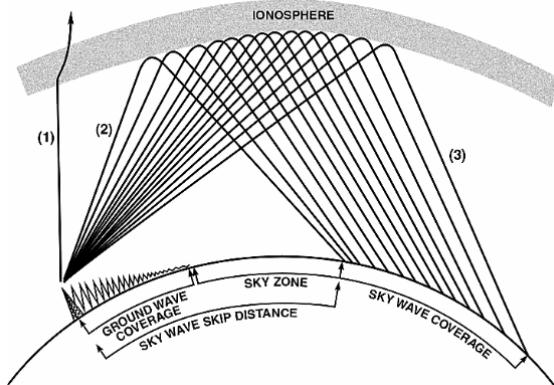
- Communicatie kan verstoord worden, zelfs door een object buiten de line-of-sight



Figuur 79: Verstoring door object in line-of-sight van transmissie

8.8.11 Propagatie van RF-signalen

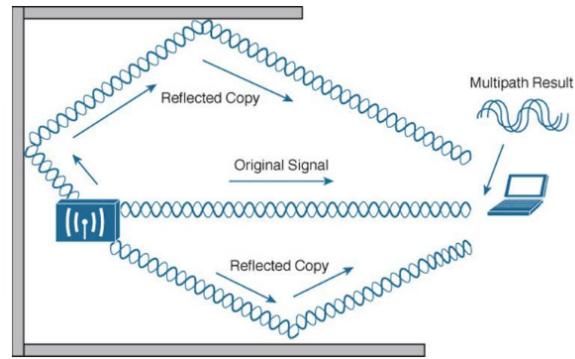
- Bepaalde effecten kunnen worden gebruikt
- Bv: Communicatie mbv reflectie op de ionosfeer



Figuur 80: communicatie mbv reflectie op de ionosfeer

8.8.12 MIMO

- = Multiple In / Multiple Out
- Reflectie en scattering kan positief worden gebruikt
- Beamforming
- Extra processing nodig



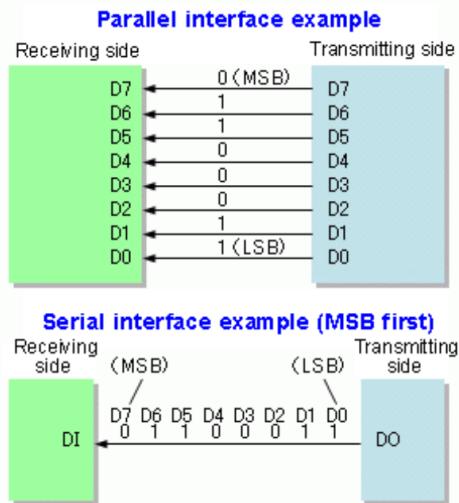
Figuur 81: MIMO

8.8.13 Multiplexing

todo

9 Seriele communicatie

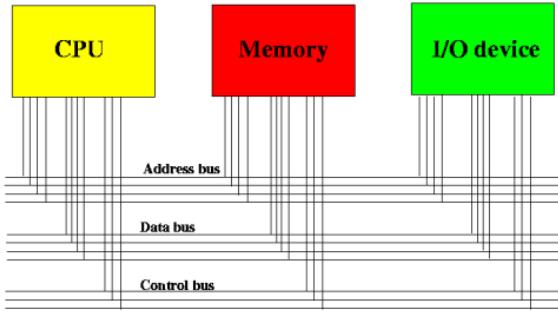
9.1 Serieel vs parallel



Figuur 82: Serieel vs parallel

9.2 Parallelle communicatie

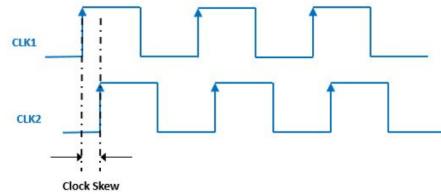
- Eenvoudig
- Sneller (bij eenzelfde kloksnelheid)
- Veel connecties
- Clock skew / jitter



Figuur 83: Parallelle communicatie

9.2.1 Clock skew

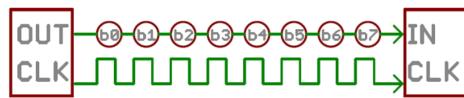
- Verschillende vertraging tussen signalen
- Begrensd maximale klokfrequentie



Figuur 84: Clock skew

9.3 Seriele communicatie

- Minder bekabeling
- Minder problemen met timing
- Minder last van crosstalk
- Complexer / SERDIS



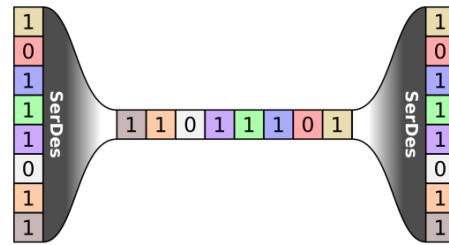
Figuur 85

9.4 Serieel vs Parallel: wanneer gebruiken?

- Lange afstanden: serieel, want minder verbindingen
- Vroeger: lokale korte verbindingen, parallel
- Tegenwoordig: snelle lokale verbindingen, serieel
- ATA in harde schijven vervangen door SATA (Serial ATA)
- PCI vervangen door PCI Express op moederborden, ook serieel

9.5 SERDIS

- Serialiser
- Deserialiser

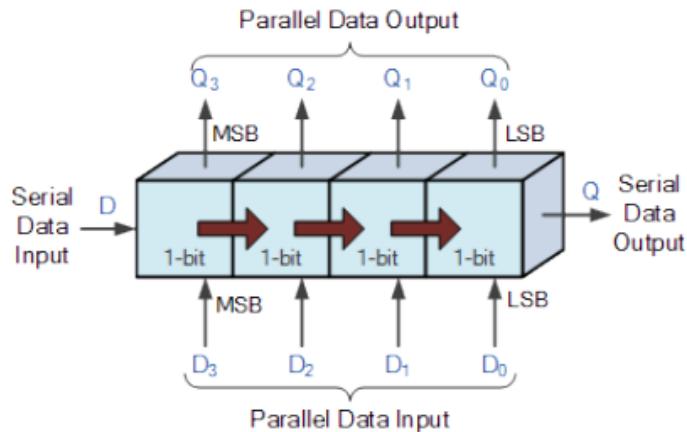


Figuur 86: SERDIS

9.5.1 Implementatie

Met schuifregister

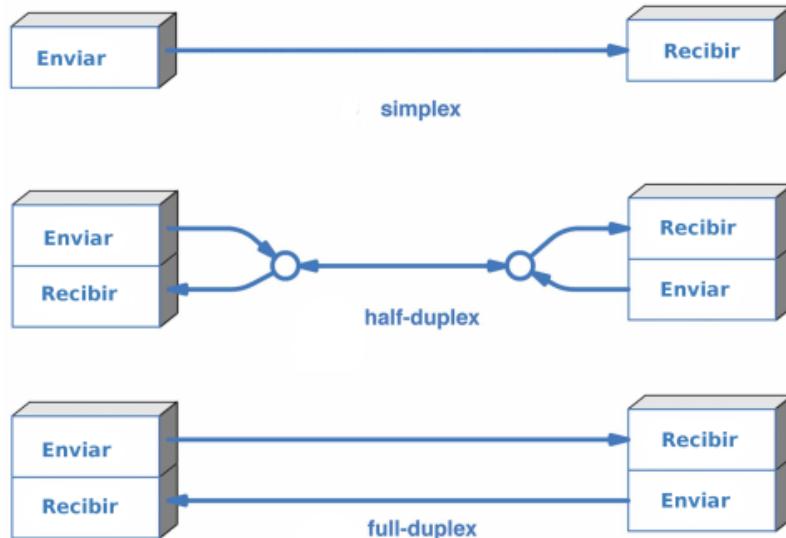
- SIPO (serial in parallel out)
- PISO (parallel in serial out)



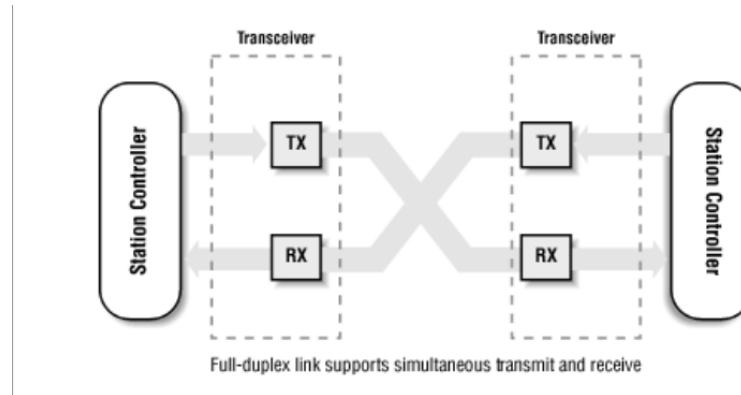
Figuur 87: Schuifregister

9.6 Duplex

- Full-duplex
- Half-duplex
- Simplex



Figuur 88: Verschillende verbindingen



Figuur 89: Full duplex

9.7 Flow Control

Beperken hoeveel de zender kan/mag versturen.

- Kan in hardware ⇒ extra verbindingen
- Kan in software ⇒ controle karakters
- Hardware (CTS / RTS / Enable / ...)

- Xon / Xoff: starten/stoppen van het signaal
- No

9.8 Synchroon vs Asynchroon

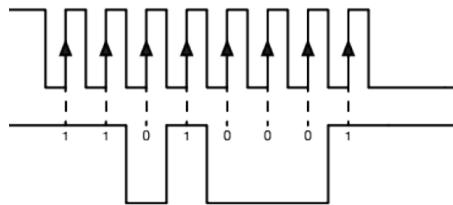
- Synchroon \Rightarrow Gebruikt een aparte kloklijn
- Asynchroon \Rightarrow Geen kloklijn



Figuur 90: Kloklijn

9.9 Synchrone seriele communicatie

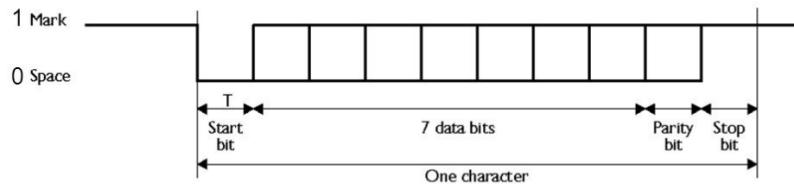
- 1 device genereert de klok
- Alle transmissies synchroniseren op deze klok
- Kan op de stijgende flank of op de dalende flank



Figuur 91: Synchrone seriele communicatie

9.10 Asynchrone seriele communicatie

- Werkt zonder aparte kloklijn
- Dus minder verbindingen
- Syncrhonisatie door ontvanger noodzakelijk
- Start en stop bit (/conditie) noodzakelijk



Figuur 92: Asynchrone seriele communicatie

9.10.1 Parameters

- Baudrate (=bits per seconde)
- Aantal databits
- Pariteitsbit
- Aantal stop bits

9.10.2 Standaarnotering

- Bijvoorbeeld: 9600-8-N-1
- Baudrate = 9600
- 8 = aantal databits
- N = welke pariteit
- 1 = aantal stop-bits

9.10.3 Baudrate

= Snelheid in bits/seconde

- Niet alle bits zijn databits
- Start, stop en pariteitsbit transporteren geen data

Bijvoorbeeld

- 8-N-1 \Rightarrow 80% efficientie
- 8 databits van in totaal $8 + 1 + 1 = 10$ bits = $8/10$ of 80%

9.10.4 Aantal databits

- Typisch 7 of 8 databits
- 7 bit is voldoende voor niet extended ASCII
- 8 bit \Rightarrow noodzakelijk voor binaire transmissie (bv voor firmware)
- XMODEM / ZMODEM protocollen voor binaire transfers (weinig gebruikt want er bestaan nieuwere protocollen zoals USP)

9.10.5 Pariteit

= Foutdetectie

Soorten pariteit

- Even (E)
- Odd (O)
- None (N)

Aantal 1 bits tellen en dit aantal steeds even of oneven maken

9.10.6 Stopbits

- Aantal bits op het einde van een data-bit reeks
- 1/1.5/2 bits

9.11 Snelheid

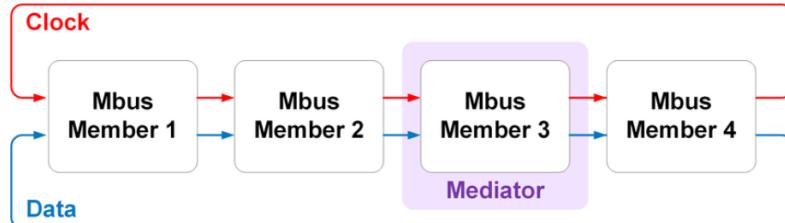
- 1 transmissie = 1 karakter
- Aantal bits voor 1 karakter = som alle bits
- Baudrate in bits/seconde

Voorbeeldoefening

- 300-8-E2
- 8 databits + 1 pariteit + 2 stopbits + 1 startbit = 12bits
- 300 bits/seconde = $300 / 12 = 25$ CPS (=characters per second)

9.12 Meerdere deelnemers

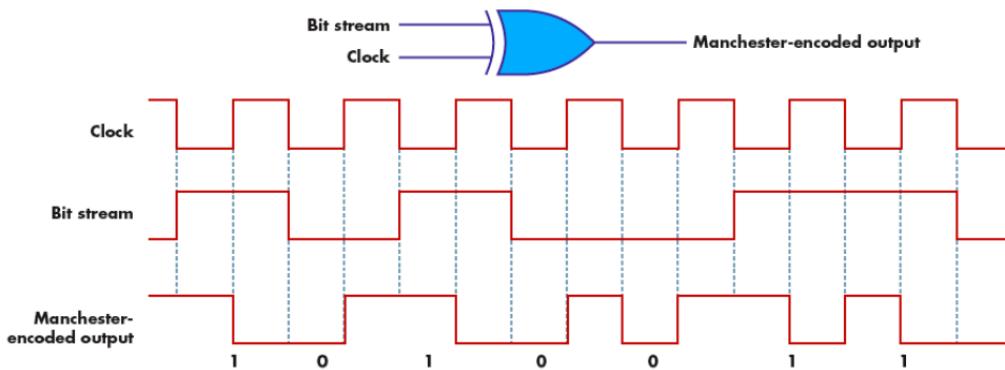
- Multi device bus
- Daisy chain (=de ene deelnemer hangt aan de andere tot we aan de laatste deelnemer hangen, dan hangt de laatste ook aan de eerste)
- ⇒ afspraken/arbitrage is noodzakelijk



Figuur 93: Meerdere deelnemers op een bus

9.13 Manchester encoding

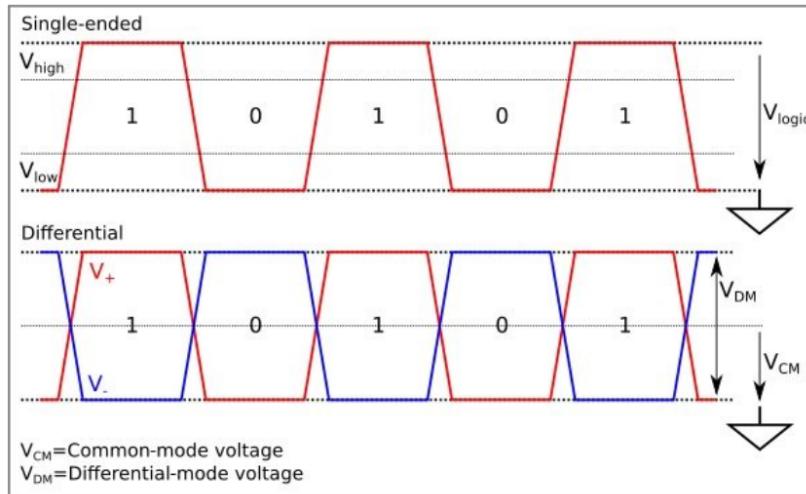
- Manier om asynchroon te werken, maar de klok en datasignaal in 1 signaal te stoppen
- Altijd voldoende omschakeling (bv voor RF transmissie)
- Met AND-operatie



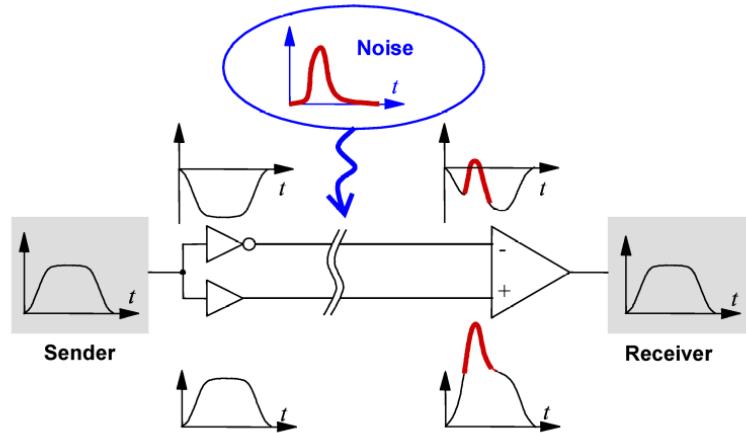
Figuur 94: Manchester encoding: Clock & Data = OUTPUT

9.14 Differentiële communicatie

- Differential vs single-ended
- Zowel synchroon als asynchroon kan differentieel of single-ended werken
- Minder gevoelig aan storingen



Figuur 95: Single-ended (boven) vs Differentieel (onder)



Figuur 96: Wat er gebeurt bij storing bij een differentieel signaal

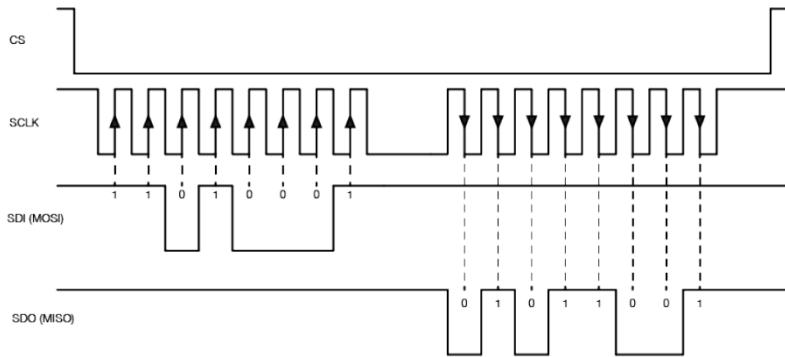
9.15 Benoeming van signalen

Deze namen vind je vaak op protocollen en devices zoals de Arduino, RPi, ...

- RX \Rightarrow receiver
- TX \Rightarrow transmitter
- Dit is mogelijk ambigu zijn. Oplossing: nieuwe naamgeving:
 - MOSI \Rightarrow Master Out Slave In
 - MISO \Rightarrow Master In Slave Out
- CS \Rightarrow Chip Select
- EN \Rightarrow Enable
- R/W \Rightarrow Read/Write
- SCL \Rightarrow Serial Clock
- SDA \Rightarrow Serial Data
- Minder gebruikte signalen (Niet expliciet te kennen)
 - CTS
 - DTR
- ...

9.16 Protocollen

= Standaarden voor communicatieafspraken



Figuur 97: Schema dat toont hoe het protocol werkt

10 Bussystemen

- Enkel directe punt-tot-punt verbindingen naar centrale eenheid
 - Meerdere devices delen een bus
- ⇒ Bussystemen zijn praktischer en (vaak) goedkoper

10.1 Communicatie op een bus

- Gelijktijdig gebruik van een bus is onmogelijk bij meerdere devices op deze bus
- Gebruik van:
 - TDM
 - Tokens (token ring)
 - CSMA (/CD)
 - Master / Slave
 - ...

⇒ Dit zijn allemaal systemen voor arbitrage op een bus

10.2 Master / Slave communicatie

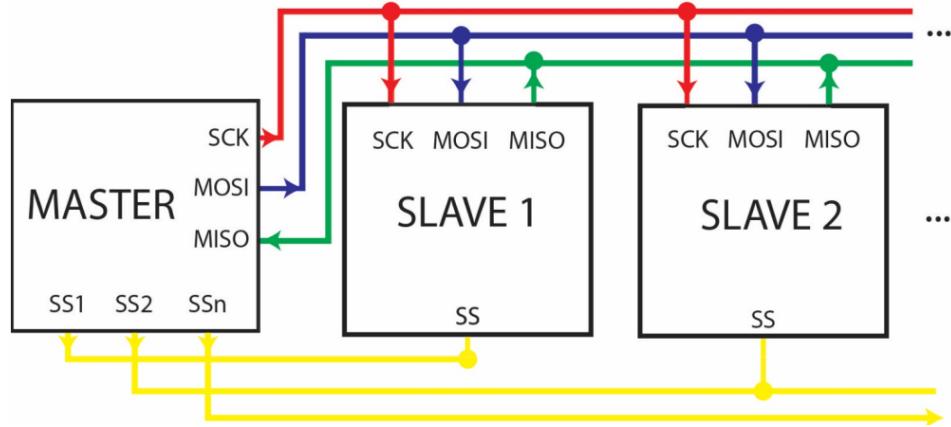
- Alle communicatie start vanuit de master
- Slaves antwoorden op vraag van de master
- Polling van slaves voor informatie

10.3 Serial Peripheral Interface (SPI) communicatie

- Synchrone seriele bus (=er is een aparte kloklijn aanwezig)
- Full duplex (=communicatie in 2 richtingen op hetzelfde moment mogelijk)
- 1 master
- 1 of meerdere slaves

10.3.1 Werking

- MOSI (Master Out Slave In)
- MISO (Master In Slave Out)
- Gemeenschappelijke klok
- Chip(/Slave) Select (CS/SS) per slave

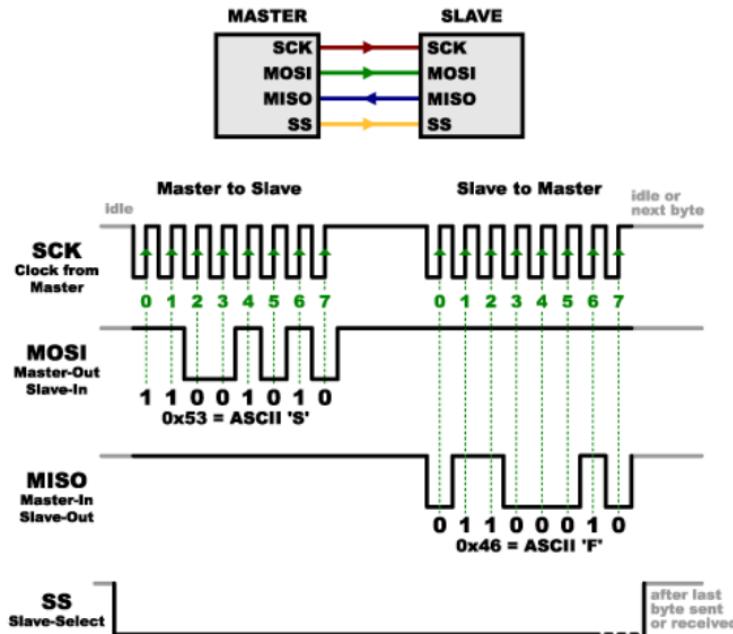


Figuur 98: Werking SPI

10.3.2 Eigenschappen

- Elke lijn heeft slechts 1 driving source (aansturingsbron)
- Clock gegenereerd door master
- Collisions zijn onmogelijk
- Slechts een enkele SPI Slave actief per moment
- Actieve Slave door Master gekozen (via Chip Select (CS) lijn)
- Een SPI Slave kan geen communicatie aanvragen, behalve via een eventueel aparte lijn (IRQ) los van de SPI bus

10.3.3 Timing Diagram



Figuur 99: SPI Timing Diagram

- SS laag om communicatie tot stand te brengen
- Master genereert een klok na SS
- Master stuurt verzoek naar slave: byte wordt doorgeklokt
- Slave stuurt op kloksignaal data terug
- Master-To-Slave en Slave-To-Master tegelijk is mogelijk want full-duplex

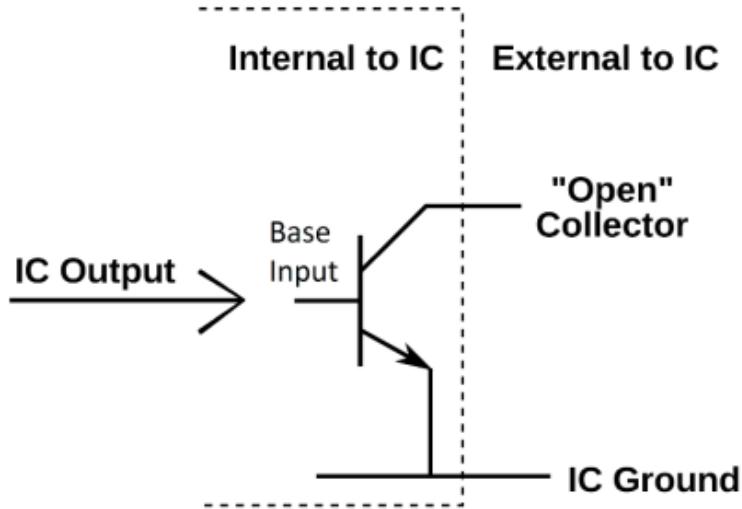
10.4 I2C communicatie

- Inter-IC Circuit ($I^2C \Rightarrow I^2C$)
- Synchrone seriele interface (=aparte kloklijn)
- 2 open collector lijnen: Clock & Data (SCL & SDA)
- Half duplex
- 1 of meerdere masters (!) ("multi master bussysteem")
- 1 of meerdere slaves
- Slaves hebben elk een adres (beslist door fabrikant)
- Typisch lagere communicatiesnelheden dan SPI

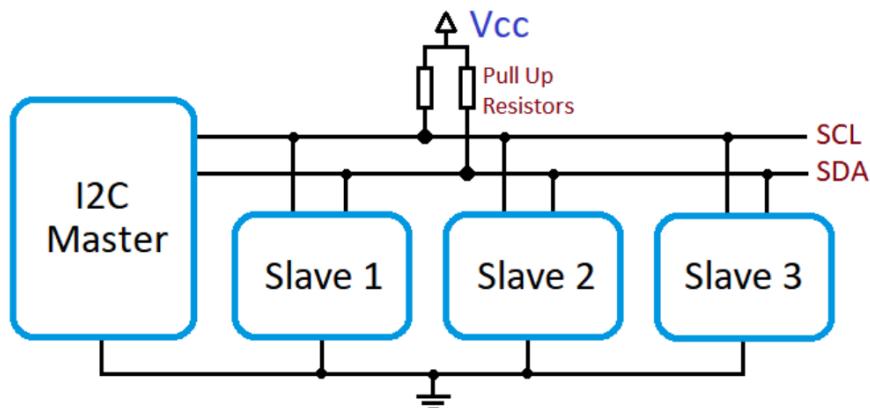
10.4.1 I2C schema

- SCL = Serial Clock

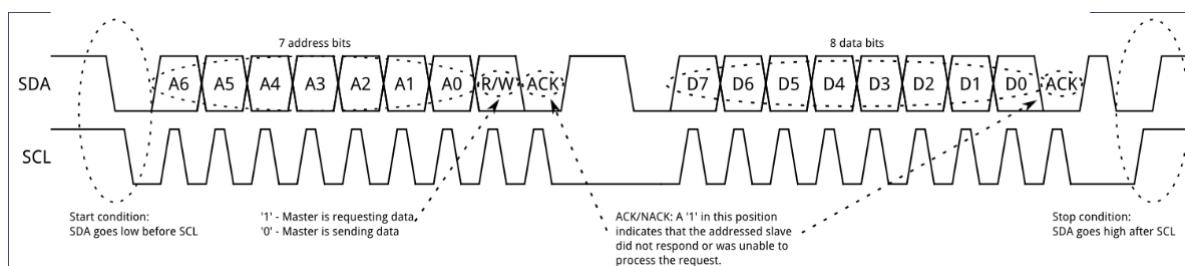
- SDA = Serial Data
- Beide zijn "Open Collector" \Rightarrow Pull-up weerstanden



Figuur 100: I2C schema



Figuur 101: I2C schema



Figuur 102: I2C Timing Diagram

- Startconditie
- 7 adresbits + 1 R/W bit: master leest (1) of schrijft (0)?
- Acknowledge-bit
- 8 data bits + Acknowledge-bit, herhaaldelijk tot alle data gelezen of gestuurd is
- Stopconditie

10.4.2 Eigenschappen

- Slechts 2 lijnen, onafhankelijk van hoeveelheid slaves
- Hoger stroomverbruik door pull-up weerstanden
- Multi-master mogelijk (complexer, niet vaak gebruikt)
- Elke slave heeft een adres (fabriek)
- Adresconflicten beperken vaak het gebruik van meerdere dezelfde Slave devices

10.5 Flow Control

10.5.1 Problemen

- Kan de ontvanger de data op tijd verwerken?
- Is de data klaar om verzonden te worden?
- Is er voldoende plaats in een FIFO-buffer (First In First Out, buffer kan overflowen)?

Mogelijke oplossingen:

- Hopen op succes en fouten opvangen (retransmission)
- Flow control

10.5.2 Hardware flow control

- RTS/CTS lijnen bij RS232 (wordt gebruikt bij Arduino en RPi)
- Clock stretching bij I^2C : slave maakt de clock trager zodat er meer tijd is om de data te verwerken
- ...

10.5.3 Software flow control

- XON/XOFF: stoppen met zenden (XOFF) tot de data verwerkt is (XON)
 - Kan soms voor fouten zorgen
- ...

10.5.4 Geen flow control

- Deterministische timing gebruiken
- Retransmisie van ontbrekende data

10.6 Belangrijke bus-interfaces

- UART Logic-Level / RS232
 - Variant: RS485 / RS422 (langere afstanden, differentieel)
- SPI
- PSP (Parallel Slave Port, dus niet serieel, bij snellere devices gebruikt)
- CAN / LIN (zoals I^2C , maar voor in voertuigen)
- I^2C
- I^2S (S = sound, voor geluid te transfereren)
- 1-wire (Dallas)

11 Foutdetectie en correctie

11.1 Integriteitscontrole

- Is de ontvangen data correct?
- Data kan 'verminkt' worden
 - Door ruis
 - Door transmissiefouten zoals collisions
 - Door moedwillige aanpassing (!)
 - ...

11.2 Detectie van fouten

- Pariteitscontrole
- 1 parity bit toevoegen
 - Even parity
 - Odd parity

⇒ totale aantal 1-bits even of oneven maken ⇒ afspraak tussen zender en ontvanger moet correct zijn

11.2.1 Parity check

- Voorbeeld even pariteit: de zender voegt een pariteitsbit toe zodat het aantal 1-bits altijd even is. Als de ontvanger een oneven aantal 1-bits ontvangt, weet hij dat er een fout gebeurd is.

Character	Sender	Parity Bit	Receiver	Parity
"E"	1000101	1	10001011	Even
"A"	1000001	0	10000010	Even
"C"	1000011	1	11100111	Even
"q"	1110001	0	11100000	Odd Error!

Figuur 103: Voorbeeld even pariteit

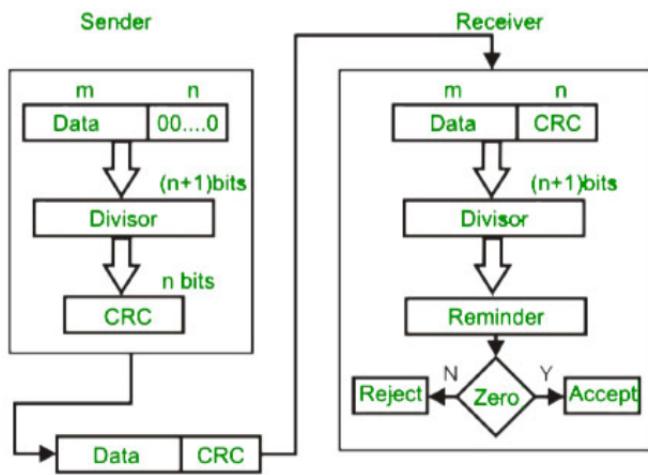
- 1-bit fouten worden gedetecteerd (zie character q)
- 2-bit fouten worden niet opgemerkt (zie character C)

11.3 Foutcorrectie

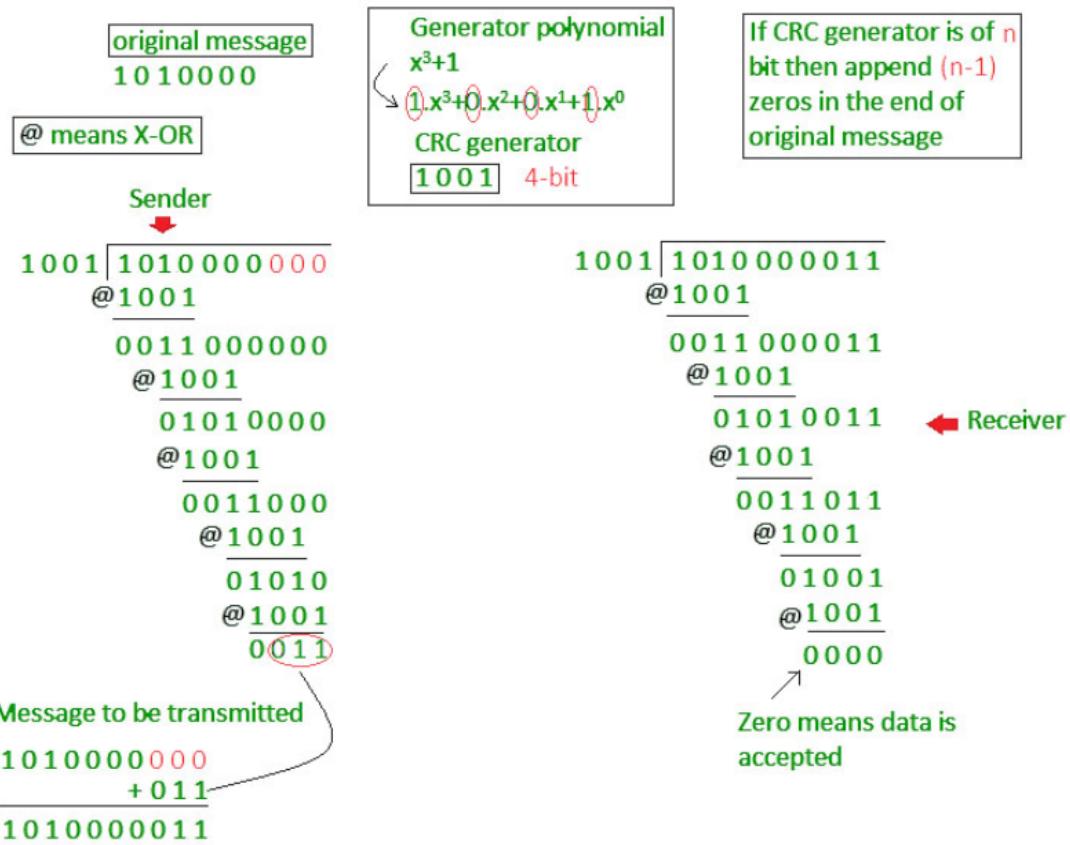
- Error correction code (ECC)
- Detectie en soms correctie van fouten
- Voorbeelden:
 - Reed-solomon code
 - Hamming-code
 - Turbo code
 - CRC's

11.3.1 Cyclic Redundancy Check (CRC)

- Kan steeds alle 1-bit fouten detecteren
- Kan fouten tot aantal CRC-bits detecteren
- Kan sommige andere fouten detecteren, afhankelijk van het gekozen polynoom



Figuur 104: Voorbeeld CRC



Figuur 105: Voorbeeld CRC-4 met polynoom $x^3 + 1$

11.4 Integriteitscontrole

- Foutdetectiecodes zijn geen cryptografische functies:
 - ECC beschermt tegen toevallige fouten
 - ECC beschermt niet tegen bewuste manipulatie
- Gebruik een hash-functie bij integriteitscontrole
 - Bijvoorbeeld SHA2 of SHA3