**Practicum Analoge Sensoren**

****

Bron: [Sparkfun](https://cdn.sparkfun.com//assets/parts/1/8/4/00242-1.jpg)

Inleiding 2

Code eisen 3

Meetrapport 3

Benodigdheden 3

Meetopstelling 4

Opdracht 1: Uitvoeren van metingen met Analoge Afstandssensor 7

Opdracht 2: Bepalen van Overdrachtsfunctie 8

Opdracht 3: Toepassen van Moving Average Filter 9

Opdracht 4: Toepassen van Lineaire Interpolatie 10

Opdracht 5: Uitwerken in meetrapport 11

## 

## Inleiding

In dit practicum laat je zien dat je:

* De **overdrachtsfuncties** van een afstandssensor kan bepalen.
* Kunt werken met moving average filters.
* Lineaire interpolatie kunt toepassen.
* Je code kan schrijven die gebruik maakt van voorgaande leerdoelen   
  (Bijvoorbeeld functies, pointers en arrays).

Het verloop van het practicum kan worden samengevat in onderstaande flow diagram:

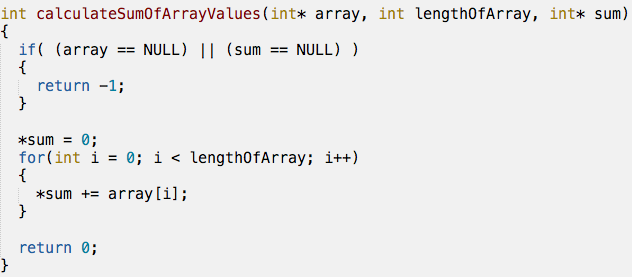
### 

Figuur 0: Flow diagram met een overzicht van dit practicum

### Code eisen

Laat bij het maken van de code zien dat je kunt omgaan met functies, pointers, arrays en dat je het gebruik van globale variabelen zoveel mogelijk vermijdt.

Zie bijvoorbeeld onderstaande functie:



Maak in de code die je schrijft ook gebruik van:

* Checks op mogelijk NULL pointers van input variabelen (‘int\* array’ en ‘int \*sum’).
* Return variabele om aan te geven of het resultaat correct  
   (bijvoorbeeld ‘return -1;’ als er een fout is opgetreden).
* Pointers om output variabelen terug te geven (bijvoorbeeld ‘int\* sum’).

### Meetrapport

De resultaten van alle onderdelen van dit practicum verwerk je in een meetrapport.

Het is **niet voldoende** om alleen maar de antwoorden van alle deelvragen in te leveren.

## Benodigdheden

* Vel A4 wit papier.
* Plakband voor vastzetten van de sensor
* Arduino en toebehoren

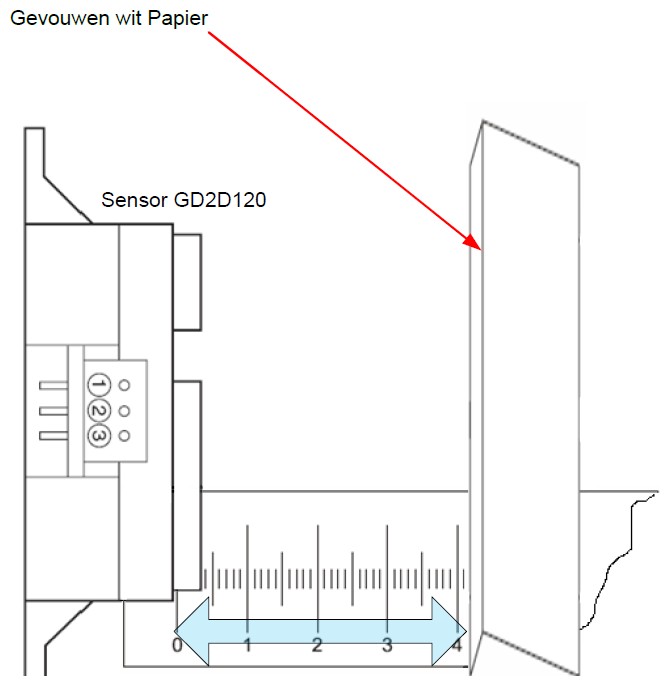
Verkrijgbaar via ISSD:

* Sharp GD2D120 IR of vergelijkbare sensor (onderdeel van ES2-box).
* Een Multimeter
* Een Centimeter

## Meetopstelling

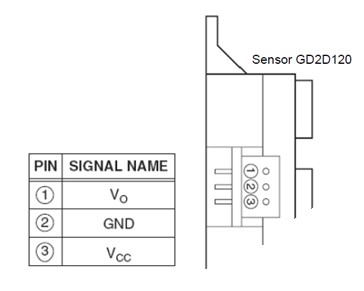
Sluit de afstandssensor op de Arduino (Vcc = 4.5 - 5.5 Volt). Raadpleeg de technische specificaties van de sensor. Plaats een gevouwen witte A4 papier voor de sensor. Meet de uitgang **Vo** (zowel analoog als digitaal) van de afstandssensor als functie van de afstand **L**. (**L**= is de afstand tussen de sensor en het gevouwen wit papier.).

Zie ook onderstaand figuur. Voor aansluiting op de Arduino, zie volgende bladzijde.

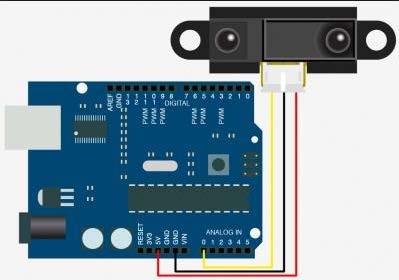


**Afstand L**

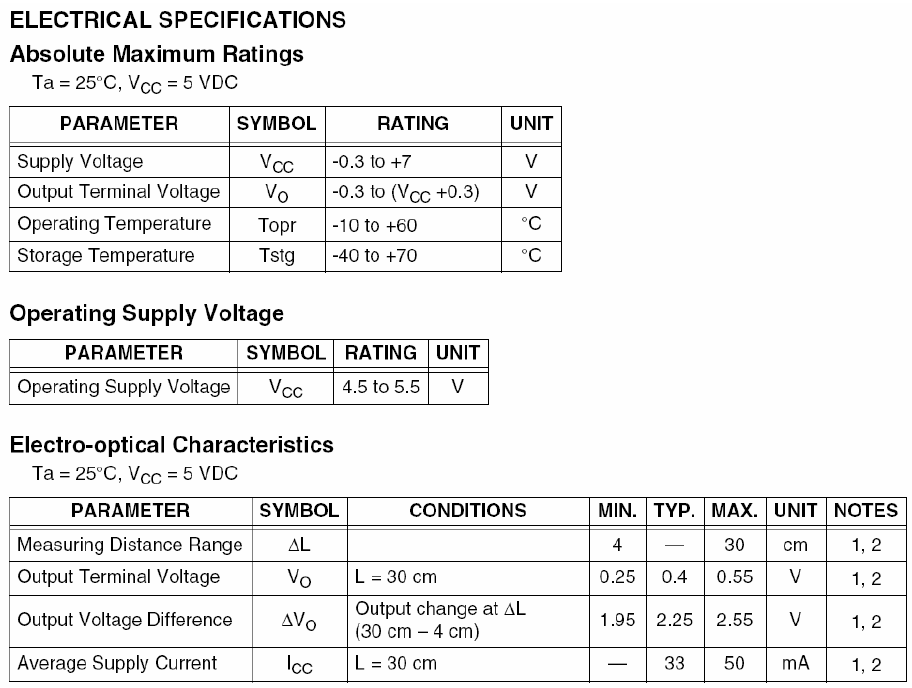
Figuur 1: Meten van afstand met behulp afstandssensor en papier



Figuur 2: Betekenis van pinnen op de afstandssensor.



Figuur 3: Aansluiten van Arduino afstandssensor   
 (voor de sensoren met een andere kleur bedrading: blauw =Vo, zwart = Gnd, rood = Vcc)



Figuur 4: Specificaties van afstandssensor (De waarden in het grijze vlak kan variëren afhankelijk van het type sensor.)

## Opdracht 1: Uitvoeren van metingen met Analoge Afstandssensor

Meet bij verschillende afstandswaarden zowel de analoge als de digital output van een afstandssensor.

Gebruik voor de analoge waarde een multimeter om het voltage van de afstandssensor te meten. De digitale meting voer je uit met hulp van een eenvoudige Arduino programma.

1. Verwerk de meetresultaten in een **tabel**. Zorg voor voldoende meetgegevens. Minimaal 25 meetpunten afhankelijk van het meetbereik van je sensor.

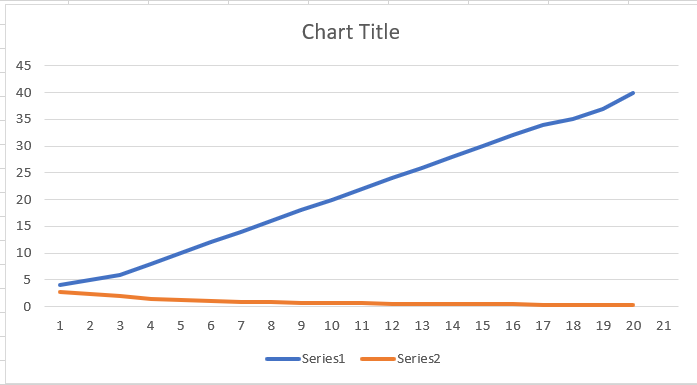
|  |  |  |
| --- | --- | --- |
| Afstand L [cm] | Analog Voltage [Volt] | Analoge Count [Count] |
| 4 | 2,76 | 565,248 |
| 5 | 2,44 | 499,712 |
| 6 | 2,04 | 417,792 |
| 8 | 1,53 | 313,334 |
| 10 | 1,27 | 260,096 |
| 12 | 1,05 | 215,04 |
| 14 | 0,93 | 190,464 |
| 16 | 0,83 | 169,984 |
| 18 | 0,72 | 147,456 |
| 20 | 0,66 | 135,168 |
| 22 | 0,64 | 131,072 |
| 24 | 0,56 | 114,688 |
| 26 | 0,52 | 106,496 |
| 28 | 0,49 | 100,352 |
| 30 | 0,46 | 94,208 |
| 32 | 0,43 | 88,064 |
| 34 | 0,39 | 79,872 |
| 35 | 0,41 | 83,968 |
| 37 | 0,39 | 79,872 |
| 40 | 0,35 | 71,68 |

(\*\* Het kan zijn dat jouw sensor een groter meetbereik heeft.)

1. Zet de volgende gegevens uit in een **grafiek**:

* X-as: het Analoge Voltage (afkortingssymbool AV)
* Y-as: de Afstand (afkortingssymbool L)

Gebruik hier een spreadsheet programma voor (Excel, Google Sheets, Numeric, SciDavis).



## Opdracht 2: Bepalen van Overdrachtsfunctie

1. Bepaal voor **de grafiek** de **overdrachtsfunctie**. Probeer voor de overdrachtsfunctie in ieder geval de volgende functies:

|  |  |
| --- | --- |
| **Type functie** | **Functie vergelijking** |
| Lineaire | y = a . x + b |
| Polynoom – 3de graads | y = a . x^3 + b . x^2 + c . x^1 + d |
| Power | y = a . power(x, b) |
| Exponentieel | y = a . exp( b . x) |

Tabel 1: Overzicht van mogelijke overdrachtsfuncties.

Hierbij komt x overeen met de ‘analoge count’, en y komt overeen met de ‘afstand L [cm]’. De fit-constanten (a,b,c,d) zijn verschillend voor de verschillende overdrachtsfuncties.

**Onderzoek** hoe je met jouw spreadsheet programma zelf een functie in een grafiek kan laten fitten én de gevonden functie vergelijking kan laten zien om zo de constanten (a,b,c,d) per overdrachtsfunctie te bepalen.

In Excel doe je dit via het toevoegen van een ‘Trend Line’. Via SciDAVis door analysis -> fit wizard te kiezen.)

1. Implementeer naast de lineaire overdrachtsfunctie ook nog de 2 ander meest geschikte overdrachtsfuncties op de Arduino. Onderzoek welke van de 3 algoritmen nauwkeuriger is en welke het snelste. Je meet de executie snelheid m.b.v. je programma (hint: micros() ).   
     
   Het programma herhaalt steeds een afstandsmeting en laat via een terminal de volgende gegevens zien:   
    + de gemeten analoge count van de afstandssensor  
    + per overdrachtsfunctie wordt getoond:  
    + de berekende afstand in [cm].  
    + de executie-tijd in micro-seconds voor het bereken van de afstand.

### linear y = -13,729x + 34,017

### Polynoom – 3de graads y = -9,1607x3 + 51,907x2 - 96,036x + 65,009

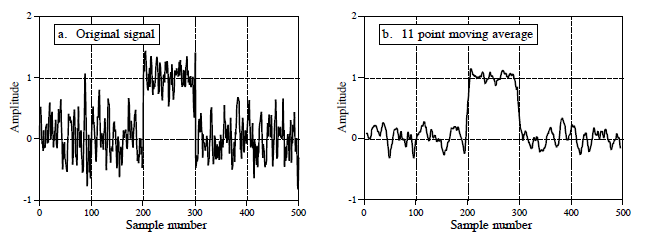
### Power: y = 12,849x-1,098

### Exponentieel y = 42,541e-0,95x

1. Welke overdrachtsfunctie kies je voor een uiteindelijke implementatie? Motiveer je antwoord.

*De Polynoom – 3de graads want die komt het meest overeen met de grafiek.*

## Opdracht 3: Toepassen van Moving Average Filter

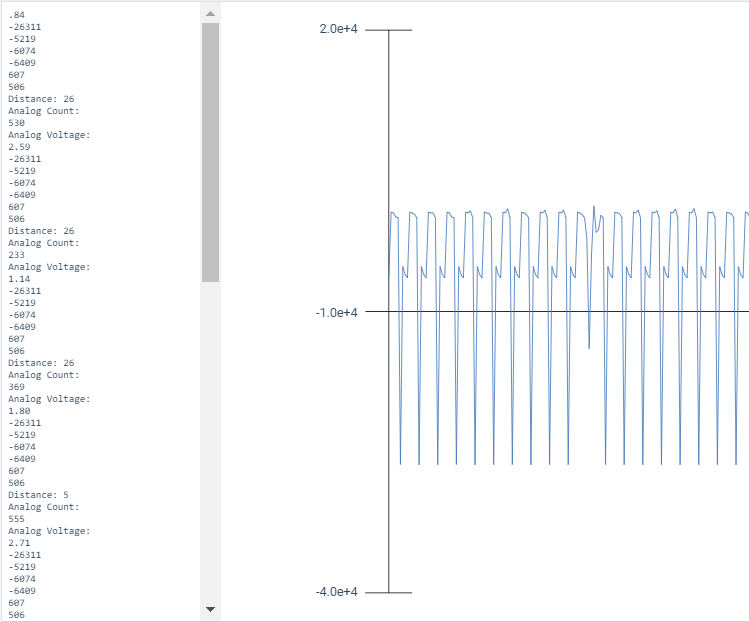
  
Figuur 5: Een voorbeeld van een moving average filter

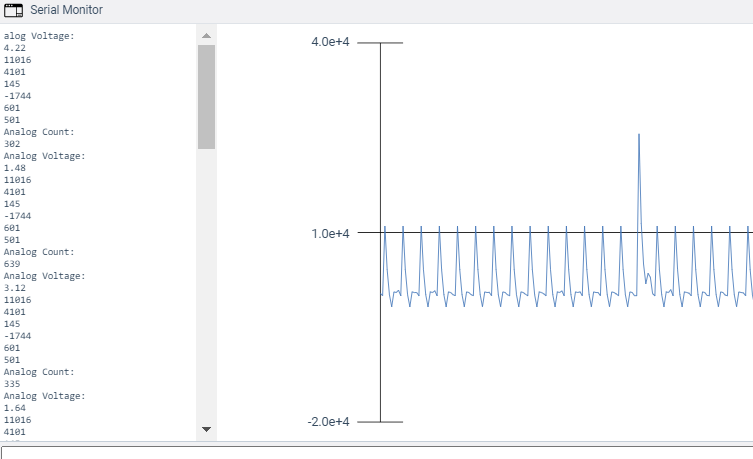
Het doel is om op basis van continu herhalende afstandsmetingen het moving average filter toe te passen. Bij een moving average filter van 11 punten laat je bijvoorbeeld steeds het gemiddelde zien van de laatste 11 metingen. Je kiest zelf met welke frequentie je de metingen uitvoert. Bijvoorbeeld 4 keer per seconden (4 Hz).

1. Schrijf code met functie(s) om het moving average te berekenen. Het aantal meetwaarden waarover het moving average wordt berekend, moet makkelijk aan te passen zijn. Bedenk een bruikbare manier om de laatste metingen te onthouden waarover je het gemiddelde wilt bepalen.

Pas het moving average filter toe voor het volgende aantal punten waarover het moving average   
wordt berekend: 5, 10 , 25 en 50.

1. Laat grafisch de resultaten zien van de door jouw geïmplementeerde filters.   
   In de grafiek moeten zowel het originele signaal als het moving average signaal te zien zijn.
2. Wat kun je zeggen over de respons van de sensor ná het filter? Licht je antwoord toe aan de hand van een grafiek.





## Opdracht 4: Toepassen van Lineaire Interpolatie

In plaats van het gebruik maken van een gefitte functie voor de overdrachtsfunctie, kunnen we ook lineaire interpolatie gebruiken.

1. Implementeer een lineair interpolatie algoritme voor de afstandssensor. Zorg dat je op het display gelijktijdig de afstand in cm kunt afleveren volgens de curve fit en de lineaire interpolatie.

We zien bijvoorbeeld graag een functie om lineaire interpolatie toe te passen. Deze heeft bijvoorbeeld de   
volgende inputs:

* array(s) van meetsamples (x- én y-waarden, x is het Voltage van de afstandssensor, y is de afstand)
* het aantal meetsamples
* het gemeten Voltage

De output van deze functies is dan bijvoorbeeld:

* de berekende afstand

|  |  |
| --- | --- |
| *x* | *Uitkomst y* |
| *0* | *33,999* |
| *10* | *33,329* |
| *50* | *30,647* |
| *100* | *27,299* |
| *250* | *17,249* |
| *500* | *0,499* |

|  |  |
| --- | --- |
| *y* | *Uitkomst x* |
| *33,999* | *0* |
| *30* | *≈ 59,69* |
| *25* | *≈ 134,31* |
| *20* | *≈ 208,94* |
| *15* | *≈ 283,57* |
| *10* | *≈ 358,19* |

1. Wat zijn de voor en nadelen van lineaire interpolatie t.o.v. van een gefitte curve?

*Met lineaire interpolatie kun je het gemiddelde van de lijn berekenen en wat ongeveer de uitkomst kan zijn. Met een curve kom je het dichts bij de gemiddelde waardes en krijg je de meer correcte waardes in plaats van ongeveer het gemiddelde van een rechte lijn aangezien wij geen perfecte rechte lijn hebben.*