

## Opgave 2 Moving average filter

Student: Jesse Denaux, Daan Dekoning

### Opgave:

Maak met een arduino nano en een IR-afstandssensor een afstandsmeter welke de afstand van een voorwerp kan bepalen tussen dit voorwerp en de sensor. De minimum detecteerbare afstand is 5 cm en de maximaal detecteerbare afstand is 30 cm. (opmerking door gebruik van een grafiek spanning IR-sensor in functie van  $1/(\text{afstand sensor ten opzichte voorwerp})$  bekom je een min of meer lineair verband tussen spanning van sensor en afstand. Gebruik een 5-punts moving average filter om stoorinvloeden (lees verkeerde afstandswaarden die door de sensor worden doorgegeven ten gevolge van IR-stoornissen) zoveel mogelijk te verwijderen. Om een meer betrouwbaar afstandcijfer te bekomen neem je het gemiddelde van 20 metingen na elkaar.

Breng de afstanden zichtbaar naar de console toe (zowel de ongefilterde afstandswaarden als de gefilterde zo kan je een idee vormen wat de filter doet met de meetresultaten). Als ongefilterde waarde neem je de waarde telkens aan het begin van de 20 metingen

Ter controle van de (goede) werking van de afstandsmeter kies je vier verschillende afstanden waar je het voorwerp plaatst om de afstand te detecteren. De precieze afstand waar het voorwerp zich bevindt moet gekend zijn zodat je een oordeel kan vormen over de nauwkeurigheid van je schakeling. Kies deze afstanden over het meetbereik van je sensor: 10 cm, 18,5 cm, 20 cm, 25,7 cm.

### Code:

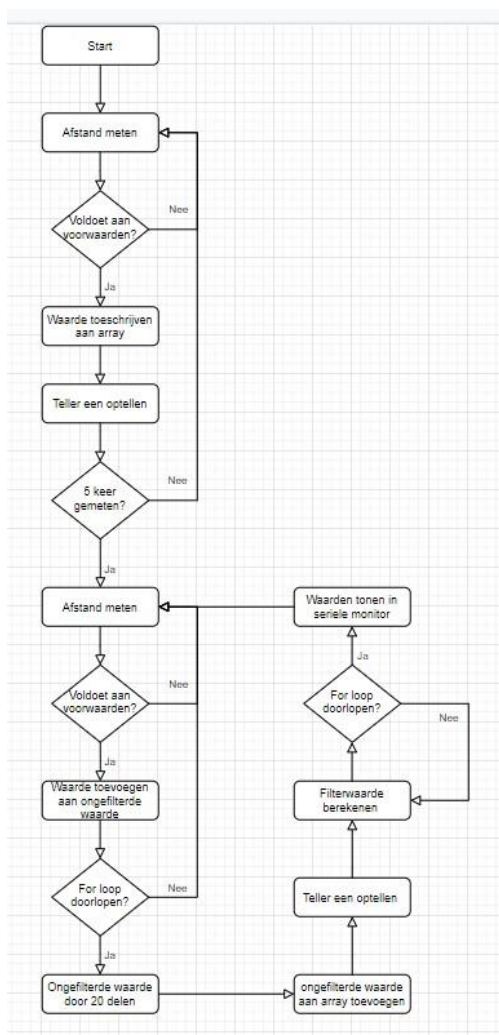
Momenteel gebruiken we in de code de Sharp IR bibliotheek om de metingen te maken. Echter zou het beter zijn om een lookup table te maken van de responsekarakteristiek van de sensor uit de datasheet. Een andere optie is om de sensor te testen en zo een lookup table op te stellen, zo is de lookup table specifiek en correct voor de specifieke sensor.

```

1 #include <SharpIR.h>
2
3 #define IRSensor A0 //We connecten de ir sensor met pin A0
4 #define model 1080 //We geven het model mee van de ir sensor
5
6 int afstand_cm = 5; //We maken een variabele aan om de afstands waarde in op te slaan
7 const int MIN_AFSTAND = 5; //We steken de minimum afstand in een variabele
8 const int MAX_AFSTAND = 30; //We steken de maximum afstand in een variabele
9 int teller = 0; //We maken een variabele teller aan om te tellen voor de 5-puntes moving average
10 int ongefilterde_waarde;
11 int gefilterde_waarde;
12
13 SharpIR mySensor = SharpIR(IRSensor, model); //We maken een nieuwe instantie aan van de SharpIR klasse
14
15 void setup() {
16   Serial.begin(9600); //We starten de seriële monitor
17   while(while_teller < 5){ //We gaan eerst al 5 waarden in de array steken zodat deze niet 0 zijn als we de berekening gaan uitvoeren
18     if (mySensor.distance() > MIN_AFSTAND && mySensor.distance() < MAX_AFSTAND){ //De afstand moet groter zijn dan de min waarde en kleiner dan de max waarde anders zal deze niet worden toegevoegd
19       afstand_cm[teller] = mySensor.distance(); //We steken de gemeten waarde in de array
20     }
21   }
22 }
23
24 void loop() {
25   for (int i=0; i<20; i++){ //We maken een for loop om 20 ongefilterde waarden te verkrijgen die voldoen aan de voorwaarden van het if statement
26     if (mySensor.distance() > MIN_AFSTAND && mySensor.distance() < MAX_AFSTAND){
27       ongefilterde_waarde += mySensor.distance(); //We tellen de afstands waarde op bij de rest
28     }
29   }
30   ongefilterde_waarde /= 20; //Na 20 waarden delen we het totaal door 20 om het gemiddelde te krijgen
31   afstand_cm[teller]= ongefilterde_waarde; //We voegen de gemiddelde waarde toe aan de array en gebruiken %5 om telkens de laatste waarde te overschrijven indien de array vol is
32   teller++; //We tellen 1 bij onze teller zodat we steeds de juiste waarde in onze array kunnen plaatsen
33   BerekenFilterWaarde();
34   ToonWaarden();
35 }
36
37 void BerekenFilterWaarde(){ //We berekenen de gefilterde waarde
38   for (int i=0; i<5; i++){
39     gefilterde_waarde += 0.2*afstand_cm[i]; //We lopen door onze array en vermenigvuldigen iedere waarde met 0.2 waarna we dit optellen bij elkaar
40   }
41 }
42
43 void ToonWaarden(){ //We tonen eerst de ongefilterde waarde en daarna de gefilterde waarde
44   Serial.print("De ongefilterde waarde = "); Serial.print(ongefilterde_waarde); Serial.println("cm.");
45   Serial.print("De gefilterde waarde = "); Serial.print(gefilterde_waarde); Serial.println("cm.");
46 }

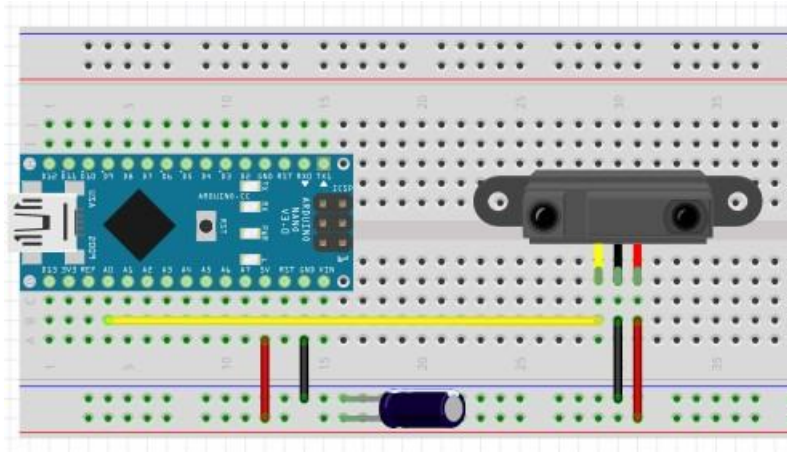
```

Stroomdiagram:



## Hardwareschakeling:

We plaatsen een condensator in de schakeling omdat dit soort van sensoren vaak ruis hebben en omdat de sensor op korte intervallen even veel stroom kan gebruiken bij een meting.



## Meetresultaten:

## Besluit: