

# Sensoren, Actuatoren en Filtering

## Qing Scholten

1.  $f_s = 9000\text{Hz}$  en  $N = 256$ . De hoogste frequentie die hier mee te meten is

$$f_{\max} = \frac{f_s}{2} = \frac{9000}{2} = 4500\text{Hz}.$$
 De minimale frequentie die te meten is is

$$f_{\min} = \frac{f_s}{N} = \frac{9000}{256} = 35,15625.$$

2.

```
BeginMicro = micros();
samples[sampleCounter] = (float32_t)analogRead(AUDIO_INPUT_PIN);
EindMicro = micros();
Serial.print(BeginMicro); Serial.print("-"); Serial.print(EindMicro);
Serial.print("="); Serial.println(EindMicro-BeginMicro);
```

COM3 (Teensy) Serial

```
25346495-25346518=23
25346606-25346630=24
25346717-25346741=24
25346828-25346851=23
25346939-25346963=24
25347050-25347074=24
25347161-25347184=23
25347273-25347297=24
25347383-25347407=24
25347494-25347518=24
25347605-25347628=23
25347716-25347739=23
25347827-25347851=24
25347938-25347961=23
25348049-25348073=24
25348160-25348183=23
25348271-25348294=23
```

☒ Autoscroll

Een sample meting duurt 23 tot 24 microseconden.

3. Dit loopt niet knel want  $23/24$  microseconden is kleiner dan de periodetijd van een sampel wat  $\frac{1}{9000} = 111$  microseconden is.

4. Er zit  $T_s = 111$  microseconden tussen twee samples.

COM3 (Teensy) Serial

742023-742052=29

5. Het duurt 29 microseconden = 0.029 miliseconden.

```
BeginMicro = micros();
for (int t=0; t<256; t++) timeDomainRaw[t]=samples[2*t]; // Copy the samples
into TimeDomain as the FFT uses samples as output
EindMicro = micros();
Serial.print(BeginMicro); Serial.print("-"); Serial.print(EindMicro);
Serial.print("="); Serial.println(EindMicro-BeginMicro);
```

6. De frequentie resolutie is  $\Delta f = \frac{f_s}{N} = \frac{9000}{256} = 35.15625 Hz$

7.

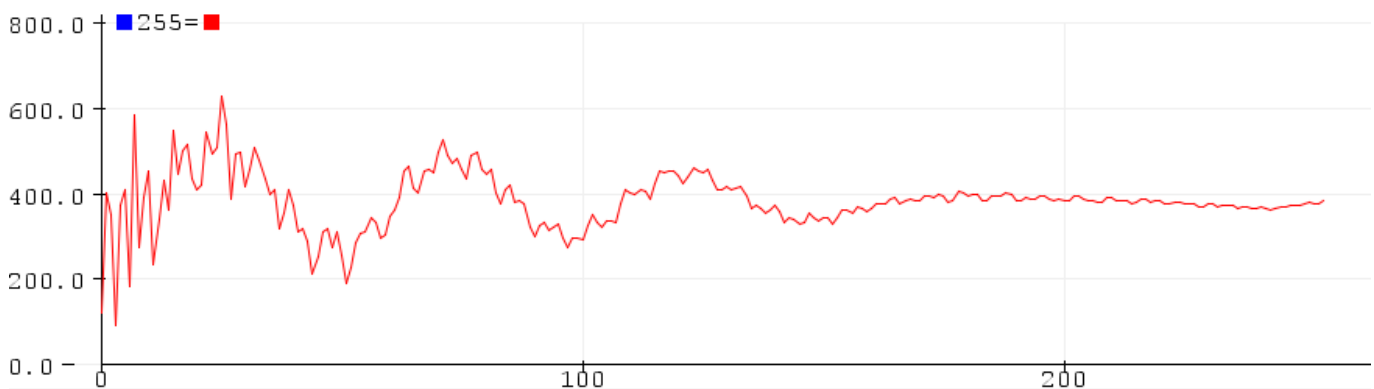
TijdsdomeinRaw

COM3 (Teensy) Serial

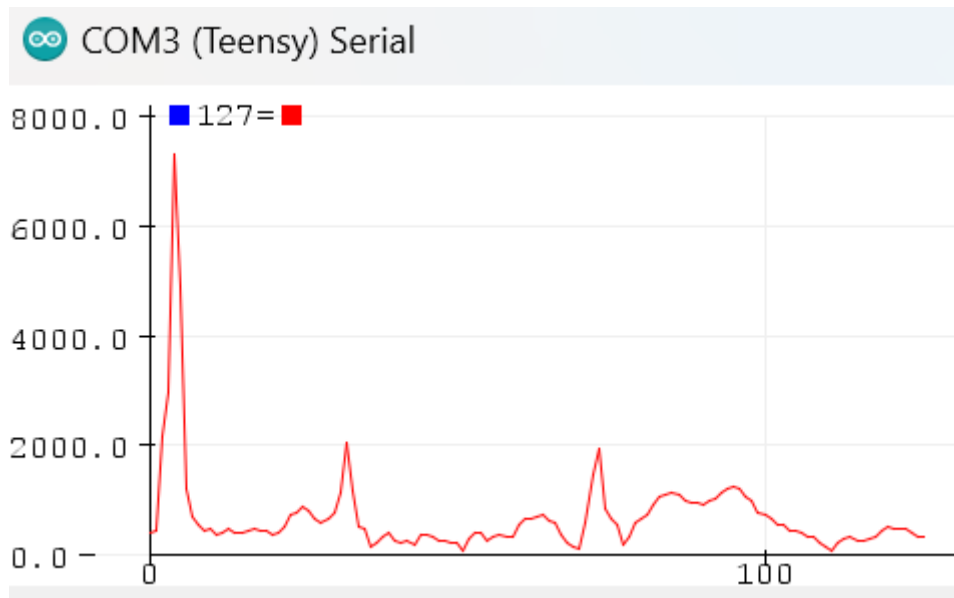


TijdsdomeinRev

COM3 (Teensy) Serial



Verzenden



Frequentiedomein

De eerste twee plaatjes zijn bijna hetzelfde. De TijdsdomeinRaw is de ruwe informatie die opgehaald is door de sensor afgezet tegen het tijdsdomein. De TijdsdomeinRev is gemaakt door middel van berekeningen met de data uit het frequentiedomein. Hierdoor is de faseverschuiving in het ruwe beeld er uit. De eerste piek in de grafiek van het frequentiedomein is de draaggolf waar alle andere frequenties aan toegevoegd zijn om een signaal te krijgen.

8.

```
COM6 (Teensy) Serial
2344473-2350687=6214
0= 373.00
1= 337.00
2= 440.00
3= 378.00
4= 54.00
5= 368.00
6= 592.00
7= 428.00
8= 448.00
9= 294.00
10= 364.00
11= 328.00
12= 310.00
13= 372.00
14= 411.00
Autoscroll
Nieuwe regel
Uitvoer wissen
```

Het duurt 6214 microseconden.

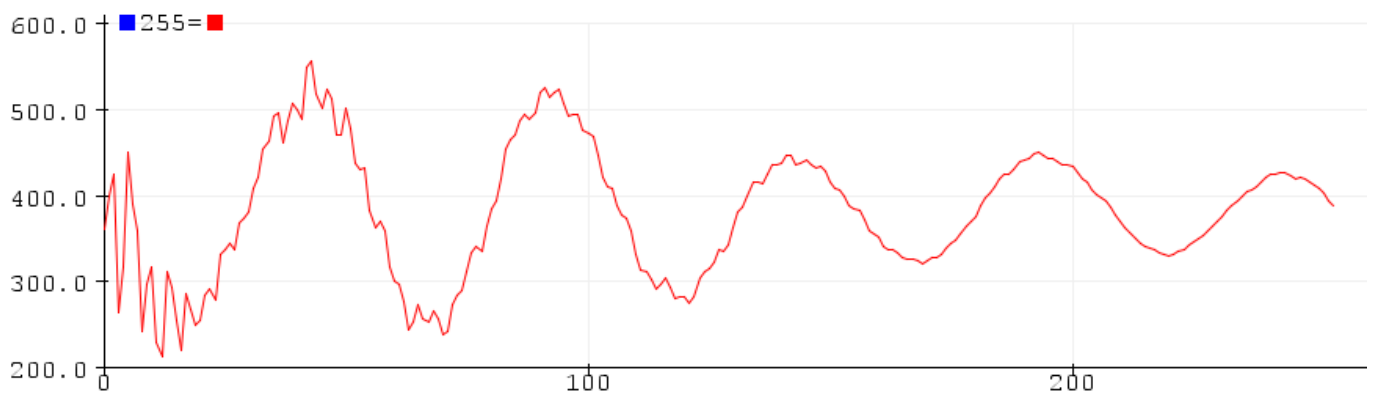
```
BeginMicro = micros();
arm_cfft_radix4_instance_f32 fft_inst;
arm_cfft_radix4_init_f32(&fft_inst, FFT_SIZE, 0, 1);
arm_cfft_radix4_f32(&fft_inst, samples); // samples now contains the real and
imaginary part of the FFT
EindMicro = micros();
Serial.print(BeginMicro); Serial.print("-"); Serial.print(EindMicro);
Serial.print("="); Serial.println(EindMicro-BeginMicro);
```

9.  $T_2 + T_3 = 29 + 6214 = 6243$  microseconden dus dat is 6.243 miliseconden. Dus de vertragingstijd is 6.243 ms. In het artikel staat dat een vertraging tot 10 ms acceptabel is. Aangezien  $10 > 6.243$  miliseconden, is deze vertraging acceptabel voor bijvoorbeeld een gehoorapparaat.

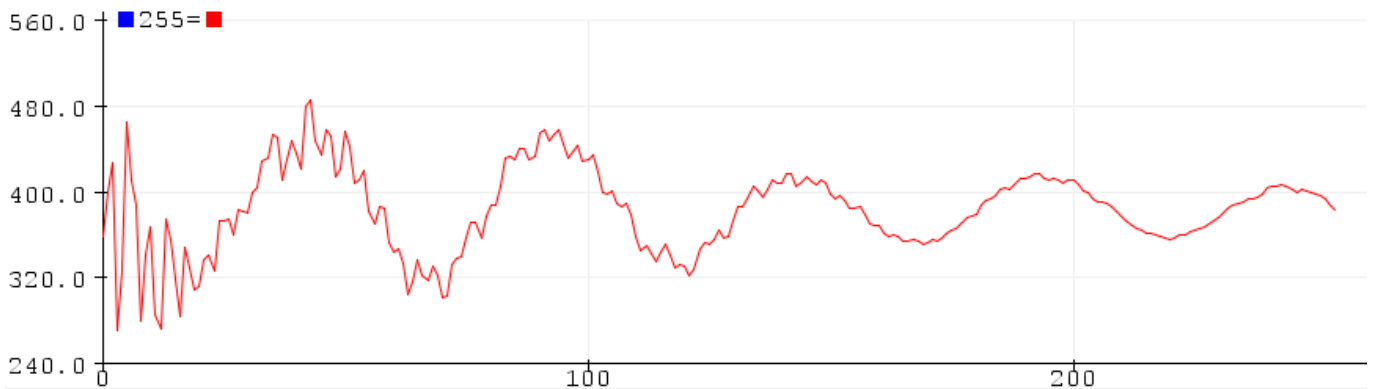
10.

```
//filter low-frequency noise
for (int i = 1; i < 50; i++) {
    samples[i] = 0;
}
for (int i = 511; i < 511 - 50; i--) {
    samples[i] = 0;
}
```

TimeDomeinRaw

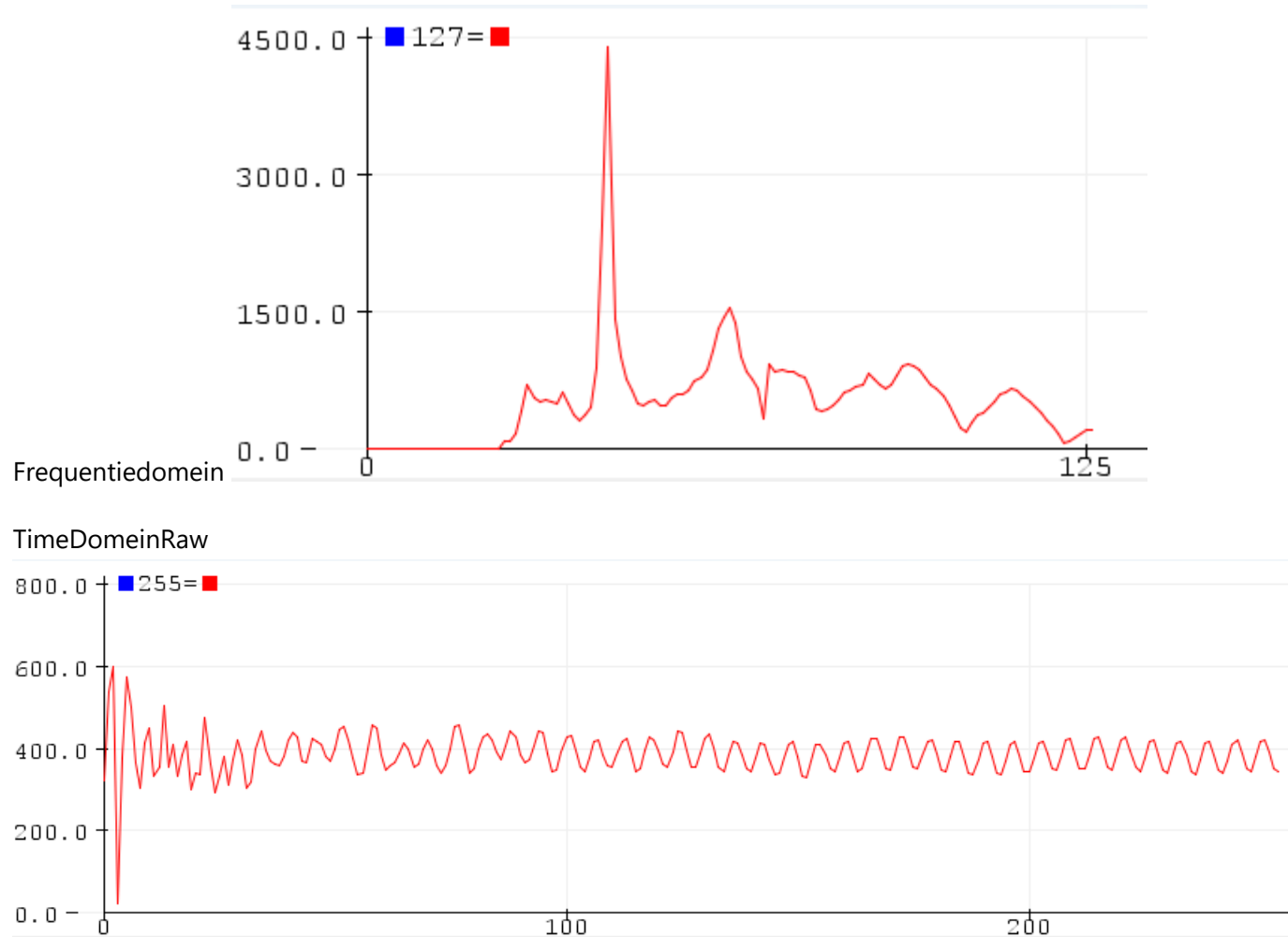


TimeDomeinRev



De noise is er uit gefiltert waardoor het signaal een mooiere sinus lijn laat zien.

11.



Er is een sinusvormig signaal te zien met de frequentie van de gegenereerde toon.

12.

De piek zit op waarde 43.  $f_s = 9000Hz$ , samples is 256 dus  $\Delta f = \frac{9000}{256} = 35.16Hz$ . Met de piek op 43 is  $f_i = 43 * 35.16Hz = 1511.88Hz$  Dit klopt met de uitgekozen frequentie van ongeveer 1500Hz.

13.

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

14. Done