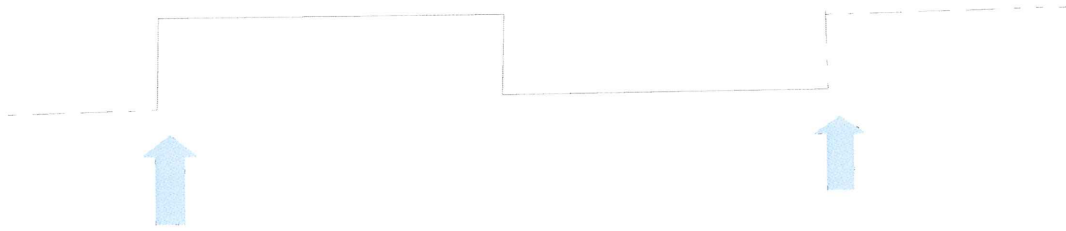


Toelichting over enkele gedachtengangen, dewelke in de CTCSS-routine toegepast zijn.

Bedoeling is, het berekenen van een frequentie, aan de hand van de tijdsduur van een aantal periodes dewelke zich, in een tijdsspanne van 100 milliseconden, voordoen.

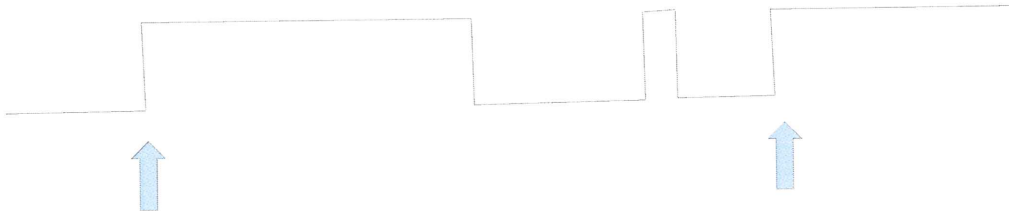
Bekijken we een onverdachte cyclus, onverdacht door afwezigheid van ruis en spikes.



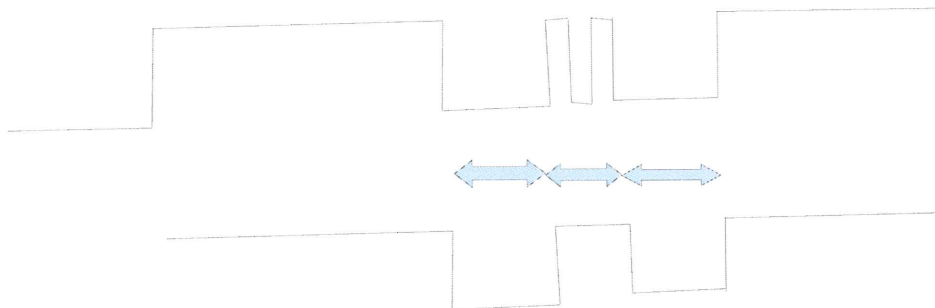
Het begin en einde van een cyclus wordt bepaald door 2 opeenvolgende stijgende flanken (of 2 opeenvolgende dalende flanken). Meten we een tijdsverschil van bv. 12,5 milliseconden, dan is de frequentie dan

$$1000 / 12,5 = 80 \text{ Hz.}$$

Evenwel, indien er zich spikes voordoen, wordt alles enigszins moeilijker. Om alles wat te vereenvoudigen, ga ik ervan uit dat de maximale tijdsduur van een spike niet langer is dan 2 milliseconden. We hebben dan:



Deze spike kan virtueel uitgegond worden doordat er, aan beide zijden van de spike voor een tijdsduur van minstens 2 milliseconden, geen stijgende / dalende flanken te vinden zijn. Dit noem ik een enkelvoudige spike. Dergelijke spikes, hoeveel er zich ook voordoen in een cyclus, kunnen allen genegeerd worden. Meervoudige spikes daarentegen kunnen niet zo maar genegeerd worden.

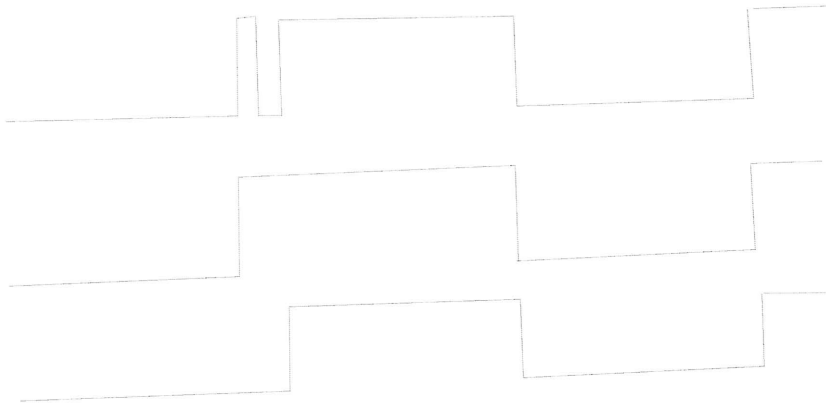


Hier is het niet mogelijk om te bepalen of de spikes uitgegond mogen worden, dan wel of ze behoren tot cycli van een frequentie van

$$80 * 3 = 240 \text{ Hz.}$$

Dus een cyclus dewelke dergelijke meervoudige spikes bevat wordt gekenmerkt als VERDACHT. Een meervoudige spike heeft minstens drie overgangen in een tijdsspanne van 2 milliseconden * 2.

Wat indien indien een enkelvoudige spike zich situeert dicht bij het begin of het einde van een cyclus ?



Dan ontstaat er terug een meervoudige spike. Terug 3 overgangen in een tijdsspanne van 2 millis *2.

Het is dus onmogelijk te bepalen waar de eerste cyclus eindigt en waar de tweede begint. Dus beiden worden als VERDACHT gekenmerkt.

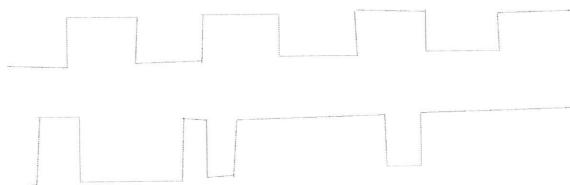
Uiteindelijk worden de duurtijden van alle onverdachte cycli tesamen geteld en op het einde van de sampling wordt de totaal tijd gedeeld door het aantal onverdachte en opgetelde cycli. Dit geeft dan de gemiddelde tijd van de periodes en daarmee wordt dan de frequentie berekend.

Het spreekt voor zichzelf dat, hoe meer ONVERDACHTE cycli er gevonden worden, hoe zekerder het eindresultaat wordt.

Wat indien een spike maximaal maar een halve milliseconde lang is, in weerwil van het tot nu toe aangehouden 2 milliseconde ?

Er zullen dan cycli als VERDACHT gekenmerkt zijn, die evengoed ONVERDACHT konden zijn. Een goede strategie is dus : bepalen van de maximale lengte van een spike en dit als norm te nemen voor het al of niet verwerpen van cycli. Merk ook op dat hoe kleiner (in milliseconden) deze norm is, hoe hoger de frequenties zullen zijn die gemeten kunnen worden.

Op deze manier opgesteld, heeft men reeds een routine die correct een aanwezige toon detecteerd, zelfs al zit die toon diep in de ruis. Bij afwezigheid van een toon, zal er random, echter frequent (+/- per seconde), toch een niet bestaande toon gevonden worden. Daartegen kan men maatregelen treffen. Bekijken we eerst een random blok golf, in vergelijking met een effectieve blok golf.



Beide blok golven hebben dezelfde lengte en hetzelfde aantal cycli, namelijk 4 cycli en bv een totale tijdsduur van 50 milliseconden. De gemiddelde tijdsduur van 1 cyclus wordt voor beide 12,5 millis. Bijgevolg, de frequentie voor beide is $1000 / 12,5 = 80$ Hz. Nochtans is er een blok golf, dewelke random gegenereerd werd, waarbij onze routine deze golf een frequentie toekent van een niet

random golf.

Een voor de hand liggende maatregel is, om slechts ONVERDACHTE cycli te aanvaarden die min of meer eenzelfde lengte hebben. Random blokgolven worden dan genegeerd.

Men zou kunnen zorgen dat deze, onverdachte cycli, opgeslagen worden in een tabel, waaruit men, op het einde van de sampling-periode, deze cycli uitzoekt welke men nodig heeft.

Om de routine eenvoudig te houden is er voor een andere aanpak gekozen.

Namelijk, de lengte van de allereerste ONVERDACHTE cyclus wordt als maatstaf genomen, waaraan alle volgende moeten voldoen (mits een marge in + en -).

Dit, tesamen met de voorwaarde dat er een minimum aantal cycli geteld moeten zijn, zorgt ervoor dat de routine nog steeds de juiste aanwezige tonen (zelfs diep in de ruis) ontdekt, maar dat er, bij afwezigheid van een toon er nu geen sprake meer is van valse aanduidingen.

Men kan opmerken dat de lengte van de allereerste cyclus een povere maatstaf is, gezien er geen garantie is dat deze maatgevend is.

Voor de uitschakeling van random gevallen speelt dit hoegenaamd geen rol en voor het zoeken naar aanwezige tonen heeft de praktijk uitgewezen dat, ook daar, deze gedachte geen rol van betekenis speelt, gezien er bij proeven gebleken is dat er uiterst zelden, een aanwezige toon, niet gevonden wordt of verkeerd benoemd wordt.

Het moet zijn dat, louter de aanwezigheid van een toon, al voldoende is om spikes en andere fenomenen uit te schakelen.

Een woordje over het begrip MINIMUM AANTAL getelde cycli.

Dit kan op velerlei manieren geïmplementeerd worden. Mij schijnt de volgende wijze als optimaal te zijn.

Er kan simpelweg een vast aantal genomen worden, of zoals nu, een percentage van het aantal CHANGES tijdens de sampling-periode. Die CHANGES kunnen nogal oplopen, zeker in het geval er per cyclus, er een enkelvoudige spike aanwezig is. Dan kan het gebeuren dat, bv. bij 80 Hz, het normale aantal CHANGES (+/- 16) verdubbeld wordt naar 32.

Men kan ook opteren voor een percentage van het maximale aantal te tellen millis.

Bij een telling van een correcte blok golf kan men verwachten dat alle cycli geteld worden en dat men (bij een samplingTime van 100 millis) men zeker 90 a 95 millis geteld heeft. Men zou kunnen stellen dat men het sample slechts aanvaard op het moment dat er 30 percent (of een ander getal) van de sampling-time opgeteld is.

De instructie wordt dan

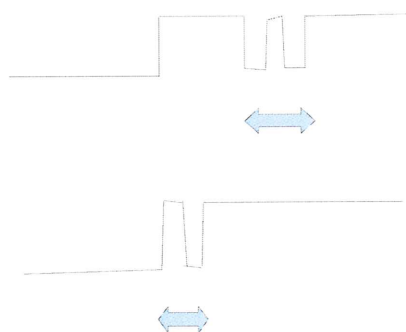
```
if ((totalTimeOk) < (sampleTime*30/10010/30))      {
    result = 0,0;                                     }
else                                                       {
    result = „„„„„„„„„„„„„„„„„„„„„„„„„„„„„„„„„„ ;   }
```

NIET GEIMPLEMENTEERD.

De routine, zoals ze nu is, is voldoende robust en geeft totale voldoening.

Wenst men nog dieper te gaan, op straffe van het verliezen van de eenvoud, zijn er nog wel mogelijkheden.

1 ervan is het volgende : om het aantal ONVERDACHTE cycli te maximaliseren is het aangeraden zich te concentreren op meervoudige spikes, die zich bevinden tussen 2 cycli in. Daar zal een dergelijke spike direct 2 cycli als verdacht kenmerken. Bekijken we nogmaals meervoudige spikes.



Een meervoudige spike, midden in een cyclus, heeft een EVEN aantal CHANGES, startend met 4, 6, 8, , , , , .

Een meervoudige spike, bij begin en einde van een cyclus, heeft een ONEVEN aantal CHANGES, startend met 3, 5, 7 , , , , , .

Hierdoor kunnen we differentiëren en bv. meervoudige spikes opzoeken met 3 Changes.

We weten dat 1 van de 2 stijgende flanken, van een dergelijke spike, het begin en einde moet zijn tussen de twee cycli. Weliswaar weten we niet dewelke dit is. Is het de eerste of is het de tweede ?

Maar het zal diegene zijn die de lengte, van de 2 cycli tesamen, verdeelt in twee, bijna gelijke cycli, van elk den helft.

We proberen eerst met de eerste stijgende flank. Lukt dat, dan worden beide helften, ieder op zijn eigen, geteld en behandeld zoals een normale cyclus.

Lukt het niet, dan doen we de oefening over met de tweede stijgende flank. Lukt het, dan zie hierboven.

Lukt het niet dan is het waarschijnlijk een random geval en doen we geen inspanningen meer.

Dit kunnen we, als het moet, ook doen voor 5, 7, 9 CHANGES, maar de routine zal exponentieel moeilijker en ingewikkelder worden en meer en meer processing time opslokken.

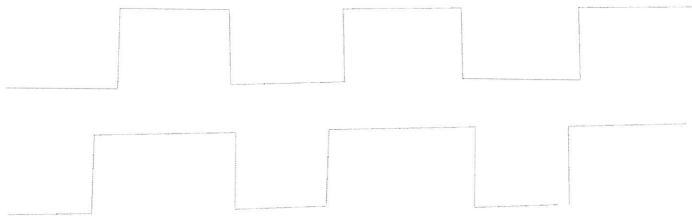
NIET BEKEKEN noch over NAGEDACHT.

Het decoderen van de tonen, wanneer er geen modulatie is, loopt anders dan het decoderen met modulatie.

In ieder geval kan de marge van, 1,5 procent meer of minder, niet aangehouden worden bij het decoderen van tonen bij modulatie. Wordt de marge van 1,5 naar 5 procent gebracht, lukt het dan toch weer.

Dat duidt erop dat de totale lengtes van de blokgolven op zich niet wijzigen, de onderlinge lengtes van individuele cycli dan weer wel.

Voorbeeld :



Beide blokgolven zijn even lang en hebben hetzelfde aantal cycli.

De bovenste golf is dan een subtone waar er geen modulatie boven op is. De cycli zijn praktisch even lang, in ieder geval verschillen ze minder dan 1,5 procent (plus of min) van elkaar.

De onderste golf is dan een subtone met modulatie er boven op. Deze veroorzaakt dan een onderlinge verschuiving, waardoor de individuele cycli meer dan 1,5 procent van elkaar verschillen, maar die dan toch, minder dan 5 procent, blijven verschillen van elkaar.

Tot nader order is dit dan mijn verklaring.

So, that's it ! Paul. *ONKAD!*

11/01/2016