

door Wim Kruif PAoWV

De MSIA is een 'Multi Sensory Injection Aid', was da nou weer, bedoeld voor HST (high speed telegraphy) training. Ondanks, of misschien wel dankzij, het vervallen van de morse-eis geniet deze tak van de hobby de laatste jaren van een toenemende belangstelling en aandacht.



In morsetelegrafie heb je als zintuigen je oren, en soms je ogen (Aldis lamp) en inmiddels antiek - op persbureaus - papierstroken met punten en strepen erop geïnt die in hoge snelheid langs de ogen van een tiepmiep m/v gleden die ze dan tegenregering blind met een noodgang uittypte.

HST is tegenwoordig een sport, met regelmatig georganiseerde wereldkampioenschappen. Kijk eens op www.rufzxp.net onder toplist en je ziet wat die mensen presteren in onderlinge competitie. Leuk om dat ook eens te proberen.

Bij het trainen om je snelheid te verhogen rijst de vraag of je dat niet succesvoller kunt doen als het morse-signaal aan meer dan een zintuig gelijktijdig wordt aangeboden. Aan die vraagstelling is recent nog

wetenschappelijk onderzoek gewijd, zie [1] en bevestigend beantwoord.

We hebben vijf zintuigen, sommige mensen zelfs zes, en het hier beschreven zelfbouwapparaat stelt je in staat om het morse-signaal aan drie zintuigen gelijktijdig aan te bieden, namelijk oog, oor en gevoel.

Het apparaat is door mij ontwikkeld in 2010, naar aanleiding van een vraag van een ham op het forum van www.lcwo.net, die daar onder het pseudoniem Digibeet morse oefende, met zoals hij schreef QTH in Elbonia.

Ik ben inmiddels 60 jaar gelicenseerd, maar dat is nog steeds een van mijn ruweweg 300 niet gewerkte landen. Digibeet schreef op de lcwo website op 31 maart 2010, dat hij heel dankbaar was voor mijn

hulp, want hij bereikte veel sneller zijn verlangde snelheid en hij liet dat daar zien aan de hand van zijn grafieken, die een scherpe knik vertonen toen hij met dit apparaat begon te trainen.

Ik zond hem behalve het apparaat ook alle ontwerpgegevens. Door een hard disk crash later had ik uiteindelijk niks over.

Naar aanleiding van die website correspondentie, vroeg SV2KBS mij recent om nadere gegevens. Ik heb van her en der weer de gegevens bij elkaar gesprokkeld, verloren gegane delen aangevuld en de resultaten worden hier nu gepresenteerd.

Ontwerp beschouwingen

De bandbreedte van je oor is normaal breder dan 3 kHz. De hoogst hoorbare geluidsfrequentie zakt in de loop van je leven van 20 kHz naar beneden 3 kHz, en dat is dan de tijd dat mensen hoorhulpmiddelen nodig hebben en geen spraak meer kunnen verstaan in een rumoerige omgeving.

Hoge snelheid morsetelegrafie, zeg 1000 karakters per minuut PARIS standaard, heeft een baudrate van 167 baud. De baudrate is het aantal kortste signaalelementen dat in een seconde past. Een symmetrische blok golf van 83 Hz heeft dezelfde baudrate en kan een riedel punten met die snelheid vertegenwoordigen.

Het oor is dus in principe geschikt om die snelheid te volgen, ook als het signaal met on/off keying op een toon gemoduleerd is, kan het oor de volledige signaalbandbreedte met gemak omvatten.

Kijken we naar het oog, dan weten we dat het oog geschikt is om Morse lichtsignalen op te nemen die met een Aldis lamp worden verzonden, zoals bij de marine gebruikelijk was. Dat gebeurde door openen en sluiten van een jaloezie voor een sterke gerichte lamp. Maar het aantal aan uit bewegingen dat met de hand gebeurde zal wel nooit 10 per seconde overschreden hebben; mannen konden dat overigens sneller dan vrouwelijke militairen, maar dat terzijde.

Gebruiken we LED verlichting, die snel in- en uitgeschakeld kan worden en sturen we die aan/uit met 83 Hz, dan zorgt de retentie van je netvlies in je oog ervoor dat je dat aan/uit niet waarneemt, maar je ziet een constante verlichting met de sterkte die hoort bij de gemiddelde LED-stroom.

Om toch hoge snelheid morse te kunnen waarnemen met het oog moet je dus iedere aan periode een ander punt op je netvlies kiezen, dat kan als je een hele rij LEDs in plaats van een neemt, die worden dan als functie van de tijd van links naar rechts gebruikt, als een soort one shot oscilloscopen display per morse karakter. De display is dus alleen geldig, zonder na-

lichttijd, gedurende de ontvangst van een karakter.

In feite is de display dus te vergelijken met de met punten en strepen beïnkte papierband, die door de typist in leesbaar schrift werd omgezet met hoge snelheid.

In beide gevallen van de LED moet het morsesignaal basisband worden aangeboden, dat wil zeggen niet gemoduleerd als omhullende op een audiotoon.

Tot slot heb je de trillerfunctie, die je met je gevoel waarneemt. Die heeft een zoemer nodig die aan een vinger of een andere plek op het lichaam wordt aangelegd. De zoemer moet snel genoeg kunnen aan- en uitschakelen, bijvoorbeeld met de genoemde 83 keer per seconde. Voor een zoemer lijkt dat tamelijk breedbandig, maar een luidspreker is als zoemer te gebruiken en die kan de bandbreedte, veroorzaakt door het aan/uitschakelen makkelijk aan.

Je kunt dan bijvoorbeeld een of meer vingers op de conus houden. Het is te verwachten dat de gebruiker de zoemertoonhoogte wil kunnen kiezen, terwijl ook basisband gekozen moet kunnen worden. Dat wil zeggen dat de conus van de luidspreker net zo op- en neergaat als de seinsleutel. Mark is laag en space is hoog. In omgekeerd geval kunnen de aansluitdraden van de speaker worden verwisseld. Bij dit ontwerp kan de buzzer 600 Hz afgeven en gehele deeltallen daarvan tot 12 toe (dus 50 Hz) en tevens basisband.

Vereiste audio input

Oefensignalen met als enige doel snelheid te trainen worden gegenereerd zonder QRM, met een hoge signaal ruisverhouding en zonder bounce (contactdender) bij in- en uitschakelen van de dots en dashes en tenminste per berichtdeel (woord, codegroep, call) een constante toonhoogte en snelheid. De bruikbaarheid van dit trainingsapparaat wordt vergroot als er per berichtdeel geringe verhogingen en verlagingen van snelheid zijn toegelaten en er dus een adaptieve snelheidsaanpassing plaatsvindt.

Voorlopig ontwerp

De input is een zuiver AM gemoduleerd aan/uit signaal bijvoorbeeld uit een computer de koptelefoon of line output. Er is dus geen contact bounce aanwezig zoals optreedt bij een gewone seinsleutel of een relais. Dat signaal wordt geklipt door een limiter LM339N. De output van de limiter is laag als de inputtone in het positieve deel van de sinus zit en de output is hoog als er geen toon aanwezig is, of de toon zit in de negatieve helft van de sinus.

Hoog uit de limiter output kan dus twee dingen betekenen: geen signaal (space) of de markttoon is bezig aan de negatieve deel van de audiosinus. Daarom hebben

we een houd-voorziening hold nodig, die de output laaghoudt gedurende de negatieve helft van de audiosinus. De hold-tijd moet tenminste de tijdsduur hebben van een halve periode van de audiotoon. De toonhoogte is onbekend, maar als ondergrens 200 Hz aanhouden is een veilige waarde, zodat de holdtijd minstens 2,5 ms moet zijn.

Als je de morsemark tenminste verlengt met 2,5 ms door die holdtijd, verleng je de marks en verkort je de spaces van het signaal. Dat kan gecompenseerd worden door de start van een mark met de holdtime, dus 2,5 ms, te vertragen.

Er wordt ook een audio sidetoon gegenereerd met een vaste frequentie van 500 Hz. Die is zeker nuttig om te beluisteren of de ontvangen morse gelijk is aan de gegenereerde morse, zodat het volume correct kan worden ingesteld en eenmalig het beslissingsniveau van de limiter. De output is echter een symmetrische blokgolf, en voor HST wordt vaak een andere golfvorm gewenst met soepel verlopende in- en uitschakelvorm. In dat geval moet geluisterd worden naar de morsebron en niet naar de sidetone.

We hebben 4 outputs: een 600 Hz sidetoon, een multiLED display, een losse LED die als Aldislamp werkt en een buzzer output. De AldisLED gaat aan en uit met de marks en spaces van het morsesignaal,

dus met de omhullende van het input-signaal. Die Aldis LED is alleen geschikt voor lagere snelheden, en om de gebruiker te overtuigen dat je snelle wisselingen niet meer kunt waarnemen.

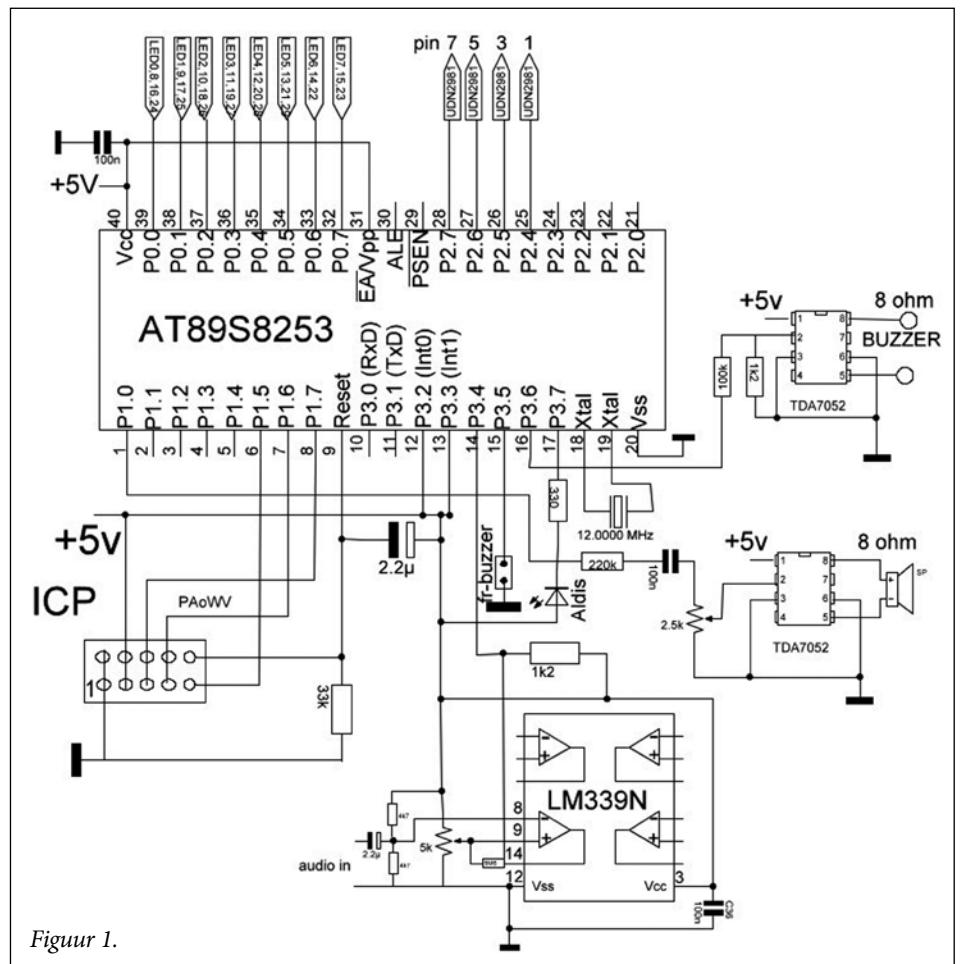
De buzzer output is uitgevoerd op twee entrees met 8 ohm impedantie, zodat de gebruiker kan experimenteren met andere buzzer typen van het type dat het beste bevalt, zoals de gebruiker van het eerste prototype ook deed, zoals hij op lcwo.net in het forum publiceerde.

Interface hardware

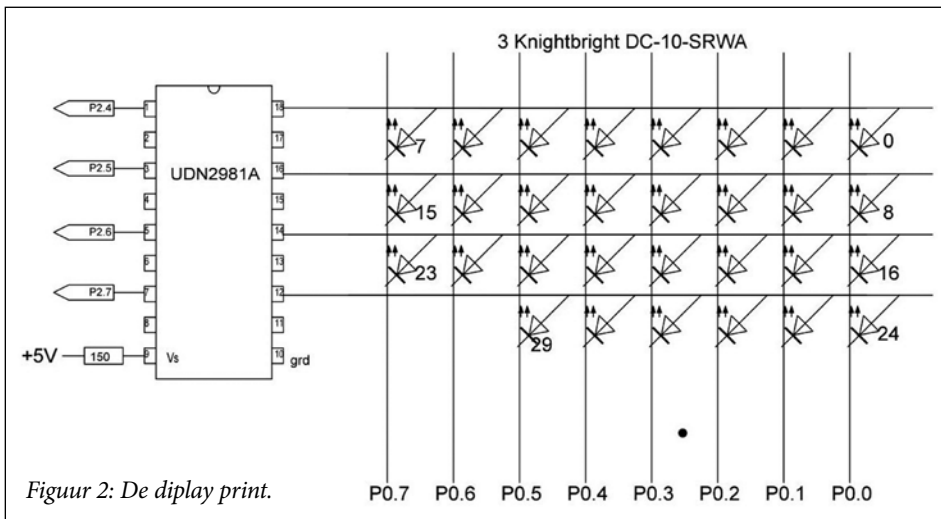
Het ingangssignaal wordt (fig. 1) via een 2,2 μ F elco om het DC signaal te blokken aan de limiter LM339N aangeboden. Als de + input daarvan een hogere spanning heeft dan de - input is de output van de limiter hoog, anders laag. Om oscillatie tijdens niveauwisseling te voorkomen is positieve terugkoppeling toegevoegd, zodat een + die net iets hoger is dan de - en de output naar hoog laat gaan de + extra omhoog wordt getrokken, zodat het verschil niet marginaal meer is. Dit leidt tot een hysteresis in het beslissingsniveau. Een instel multituernpot moet zo ingesteld worden dat zonder signaal de output van de limiter net hoog is. Niet veel meer, want dat bepaalt het punt op de sinus waar de limiteroutput van niveau wisselt.

De multi-LED display

De display (fig. 2) bestaat uit 30 LED's in



Figuur 1.



Figuur 2: De display print.

totaal 3 IC's van Knightbright. Van links naar rechts zijn ze genummerd 0 tot 29. De LED's zijn onderverdeeld in 4 groepen van 8 stuks, waarbij de laatste groep dus niet compleet is. Een portpin van de controller 89S8253 mag maximaal 10 mA sinken wanneer de pen laag is, en niet meer dan 1 pen van de port. Van port P2 worden 4 pennen gebruikt om een van de vier groepen te kiezen. Nooit is meer dan een van de vier pennen hoog. Een hoge pen van P2 kan geen 10 mA source, daarom is een driver UDN2981A tussengeschaald. De LED's hebben een serieweerstand nodig om de stroom te beperken tot 10 mA.

Totaal is slechts een serieweerstand nodig, als die wordt opgenomen tussen de Vs van de driver en +5V voeding, omdat er nooit meer dan een LED gelijktijdig aan is.

Oorspronkelijk zijn dump IC's van Baco afkomstig gemonteerd type DC-7G3H-WA, die hebben echter zeer storend 3 rode en 7 groene LED's en de intensiteit is te laag. Conrad verkoopt echter de superheldere DC-10SRWA voor € 4,20 per drie stuks en die voldoen prima. Je kunt de stroom niet opvoeren zoals normaal bij multiplexen van LED's wel zou mogen, omdat de ditfrequentie van de morse daarvoor te laag is.

Met deze IC's geeft de ingestelde 10 mA echter ruim genoeg licht, zonder dat anders een inverterende 8 bit driver en pull up weerstanden op controllerport P0 nodig zou zijn geworden. De Aldis LED zou in principe in serie met de drivervoeding kunnen staan, maar die gaf teveel spanningverlies, zodat zelfs zonder serieweerstand niet meer dan 5 mA liep. Daarom is de Aldis LED gevoed uit een aparte processorpen op een andere port (P3).

De morse bestaat uit dots en dashes (punten en strepen) en de tijdsduur van een dot waarin het morsepatroon verdeeld is wordt hier en verder 'dit' genoemd.

De software van de controller genereert een klok met een periodeduur die gelijk is

aan de dit-duur van de ontvangen morse. In het midden van elk ontvangen dit zorgt die klok ervoor dat de volgende LED aan gaat of niet, afhankelijk van of dat element een mark of een space is wordt betreffende LED aan- of uitgezet en aldus het karakter op de display gepresenteerd. Nooit is er meer dan een LED aan dus vorige LED's worden altijd gedoofd. De display is dus te vergelijken met een one shot oscilloscope trace zonder nalichting.

Realisatie vereist een synchroon met de baudsnelheid van de morse lopende klok, en na 30 dits, de volle breedte van de display, mag de klok niet meer dan een halve dit-tijd afwijken. De start van het display is de ontvangst van een letterspace plus een

halve dit om er voor te zorgen dat in het midden van de dit wordt bepaald of het mark of space is. Een letterspace resets de positieteller die bepaalt welke LEDn aan de beurt is naar LED0 en zet de ditklok stil tot de eerste markflank van het volgende karakter wordt ontvangen.

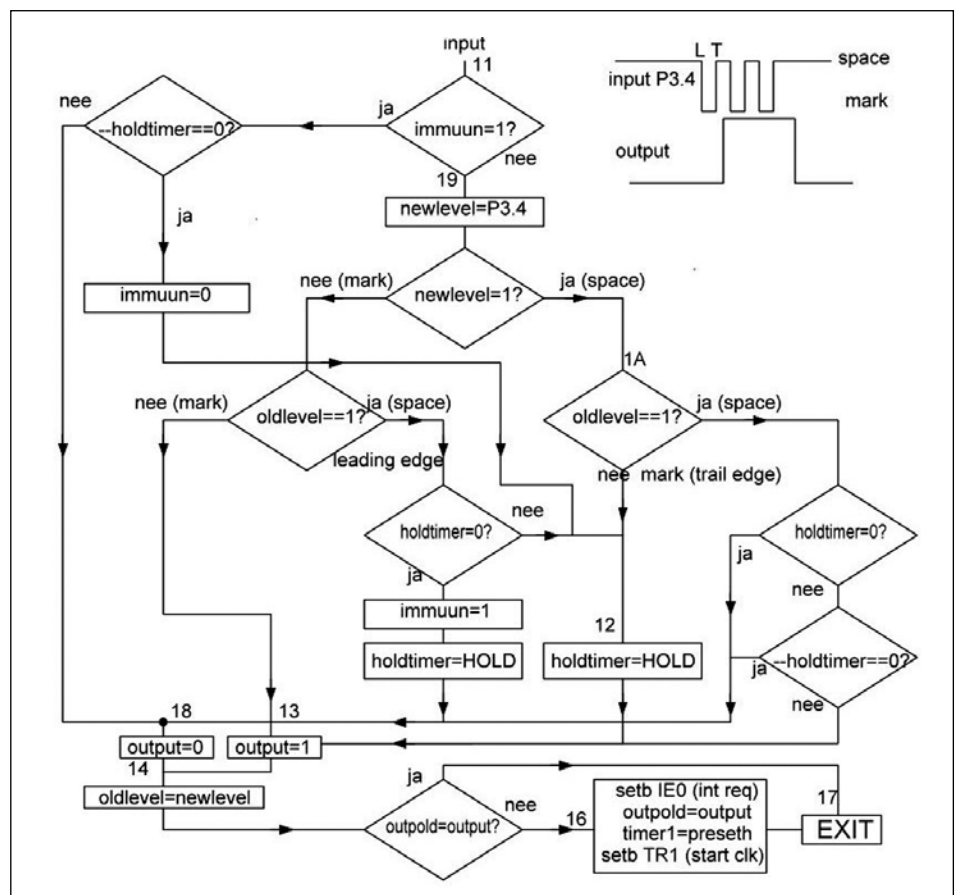
Als de klok te langzaam loopt kan een mark dit juist voor het begin en na het einde worden bekeken en wordt het dus gemist, evenzo bij een te snelle klok zou een mark dot twee keer kunnen worden bekeken en het dit aldus verdubbelen op de display. Voor een ongestoorde display mag de klok dus maximaal een half dit afwijken op de volle displaybreedte van 30 bits wat overeenkomt met 1,6% in frequentie.

Dat is nogal een strenge eis en daarom wordt bij elke flank in de omhullende van het morsesignaal re-synchronisatie van de klok toegepast, dat wil zeggen dat bij een flank de klok voor- of achteruit wordt gezet zodanig dat na een halve dit tijd gekeken wordt naar de waarde van het morsesignaal.

In dat geval, nu nog zonder adaptieve klok dus, mag de klok een halve dit tijd afwijken op 3 dits, de toegestane tolerantie is dus al daardoor ruim 16% plus en minus de nominale frequentie.

Werking van de softwaredetectie

Timer0 is ingesteld op de bemonsteringsnelheid, bij iedere overflow wordt de interruptafhandelingsroutine int1 aangeroepen.



Figuur 3. Rectifier in int1, start mark and mark space delayed over HOLD counts.

In fig. 3 staat de flowchart van deze interrupt afhandelingsroutine die de detectie voor zijn rekening neemt van het signaal dat door de LM339N limiter als geklipte gemoduleerde CW wordt afgegeven. Dat signaal staat ter verduidelijking schematisch in de rechterbovenhoek van de tekening. Een leading en trailing edge zijn daar met L en T aangegeven.

Het gedetecteerde signaal, genaamd 'output' de omhullende dus, of basisband noise, staat eronder getekend.

Er worden SPS samples per seconde genomen. SPS is ingesteld op 15625 monsters per seconde. Dat is ook de herhalingsfrequentie van de interruptafhandeling. Ingeval de morse gemoduleerd is op een toon van 500 Hz bestaat de ingang uit blokken met een mark en space van beide 1 ms en zijn er dus ongeveer 15 bemonsteringen per blokje dat door L en T begrensd wordt.

Het idee is om de output hoog te houden, na de start van een trailing edge T, gedurende een holdtime. De holdtime wordt geïntroduceerd door een holdtimer te resetten op 0 bij elke trailing edge van het input signaal. De holdtimer moet een max count HOLD bereiken voor de output daadwerkelijk laag gaat. Die timer moet dus minimaal een halve sinusperiode van de morsetoon overbruggen en de frequentie daarvan weten we niet, maar wel een veilige schatting is dat die nooit beneden $200 \text{ Hz} = \text{FCLOW}$ zal zijn, zodat een holdtime die overeenkomt met $2,5 \text{ ms}$ volstaat. In formulevorm: $\text{HOLD} > \text{SPS} / (2 * \text{FCLOW})$.

De SPS monsters kunnen verdeeld worden in 4 soorten paren van buurmonsters. De buurmonsters worden previous level en present level genoemd. De 4 groepen zijn: beide laag, beide hoog en (twee groepen) een van beide hoog en de andere laag. De laatste twee representeren dus de detectie van een leading en een trailing edge van het gemoduleerde inputsignaal, zoals aangegeven met L en T in de tekening.

Het resultaat is dat het outputsignaal hoog is gedurende de aanwezigheid van de morsetoon en aanvullend hoog gedurende de volle holdtime na de laatste trailing edge van een mark. Dat betekent dat de mark in de output dus de volle holdtime te lang zou zijn. Dat moet gecorrigeerd door de voorzijde van het outputsignaal met de holdtijd in te korten.

Derhalve moeten we verschil kunnen maken tussen de eerste leading edge van een morse-mark en de volgende leading edges. Dat kan, want bij de eerste leading edge heeft de holdtimer zijn volle eindwaarde bereikt en bij alle andere leading edges binnen een mark niet.

Een bit vlag genaamd 'immune' wordt ge-

bruikt om de state van de holdtimer delay gedurende de holdtijd aan de voorzijde te bepalen zodat daar de output laag blijft onafhankelijk van de leading en trailing edges gedurende die aanlooptijd. Dat is een aantal $\text{SPS} / (2 * \text{FCLOW})$.

Aldus werkend krijg je een outputsignaal dat een correcte representatie is van de omhullende van het binnenkomende morsesignaal, verschoven over de gekozen holdtijd.

Elke sampling interrupt (int1) vergelijkt de gegenereerde outputwaarde met die gegenereerd bij het vorige sample. Als er verschil is, dus een flank van de morse-omhullende, dan wordt een vlag IE0 geset, die in de software aldus de externe int0 afhandelingsroutine aanroept. Die meet de voortgang van de sampleteller, legt die vast en reset die teller op 0. Die doet dus de daadwerkelijke lengtemeting van het morseelement uitgedrukt in samples voor alle morse marks en spaces. Het is een 15 bits softwarecounter, die omhoogtelt met de sample rate, en daarom onderdeel is van de sampling interruptroutine int1 die dus ook de zojuist besproken taken van omhullende detectie vervult.

Alleen de duur van de marks worden op-

geborgen in een twee elements 16 bits buffer. Deze bevat dan steeds de tijdsduur van de laatste twee ontvangen marks, uitgedrukt in samples. Deze data is in gebruik om de klok adaptief te maken aan de inkomende morsesnelheid, op een wijze die verderop in de tekst wordt besproken.

De detectieklok

De detectieklok (dit klok) is gerealiseerd met timer1. Flow chart van de overflow interrupt afhandelingsroutine int3 is getekend in fig. 4. Elke overflow van de 16 bits timer triggert int3. De verstreken tijd tussen 2 interrupts bepaalt de repetitietijd van de gedetecteerde signaal bemonstering. Het doel is het output signaal, dat geleverd wordt door int1, in het midden van elk dit te bemonsteren. De timer wordt herladen tijdens de interruptafhandeling met een waarde uit RAM die 'preset' is genoemd. Die bepaalt dus de herhalingsfrequentie van de klok.

Door preset te wijzigen kan de klok adaptief, dus zich aanpassend aan de inkomende morsesnelheid, worden gemaakt.

In de reeds deels besproken detectie interruptafhandelingsroutine int1 wordt op elke up- en downflank van het gedetecteerde morsesignaal (signaal 'output') de stand van de dit-timer1 die bij overflow int3 triggert in waarde gezet halverwege zijn doorlooptinterval, zodat hij na een halve dittijd de interrupt geeft. Tevens wordt de klok van de dittimer dan ingezet, in het geval die uitstond begint hij dan dus te lopen.

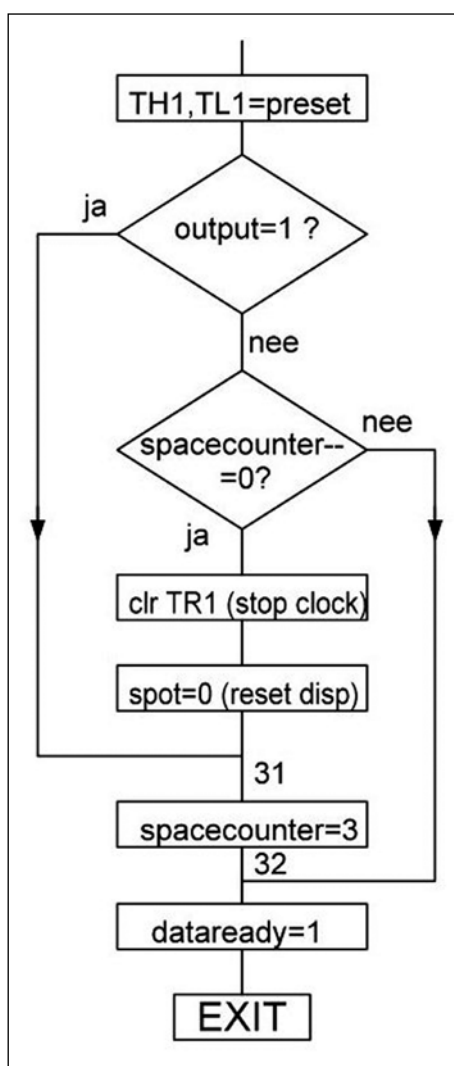
Elke flank van de gedetecteerde morse zet op die manier de klok weer gelijk, wat de eerder besproken tolerantievergroting tot ruim 16% levert, omdat twee flanken van het gedetecteerde 'output' signaal van int1 maximaal 3 dits uit elkaar liggen.

Verder wordt de ditklok timer herladen met de waarde uit 'preset' bij overflow. Int3 bevat voorts een spatieteller 'spacecounter'. Als 'output' hoog is wordt die gepreset op 3 en als 'output' laag is (space dit) naar beneden geteld. Zijn er dus 3 achtereenvolgende dits 0 (letterspatie) dan komt die spacecounter op 0, wat betekent dat hij de ditklok stilzet en tevens de positieteller op de display terugzet naar links op LED0. Er wordt altijd een vlag dataready geset, zodat het hoofdprogramma dat daarop wacht, weet dat er een geldig signaal 'output' is, de vlag dataready wordt door het hoofdprogramma gereset en de waarde van 'output' op dat moment op de volgende LED gezet.

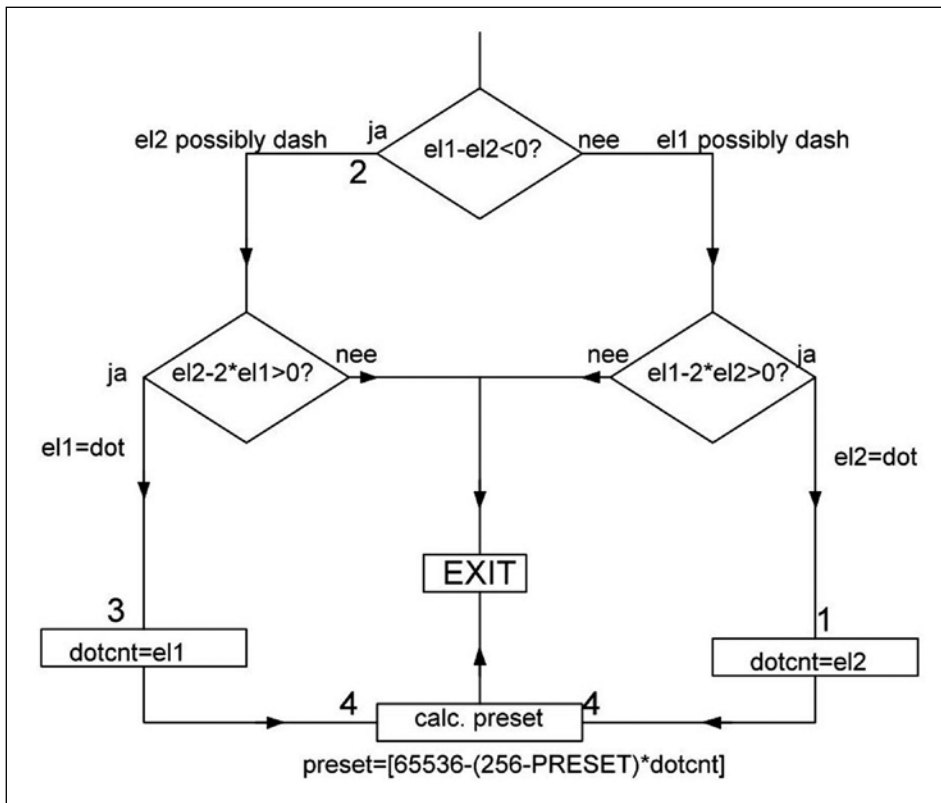
De adaptieve uitbreiding van de ditklok

De basisfrequentie van deze klok kan gewijzigd worden door de variabele 'preset' in RAM te wijzigen.

In int0 de door de software opgeroepen externe interruptroutine wordt de waarde van een marklengte opgeborgen in een



Figuur 4: Dit klok in int3.



Figuur 5: De adaptieve uitbreiding van de ditklok.

twee positiebuffer, die de laatste twee ontvangen marks bevat. Een vergelijking wordt in het hoofdprogramma gemaakt als die buffer gewijzigd is, door te bepalen welke van de twee de grootste is, en vervolgens of die minstens twee keer groter is dan de andere.

Is dat niet het geval dan is er geen verdere actie, omdat er twee ongeveer evenlange marks zijn ontvangen, is er wel verschil van minstens een factor 2, dan is bekend welke van de twee een dot was en welke een dash. De dot wordt gebruikt om de snelheid van de morse te bepalen, en daaruit de nieuwe presetwaarde in RAM voor de ditcounter te berekenen.

De afgeleide formule daarvoor staat in de flowchart van fig. 5.

Sidetone, morse berichten en de buzzer driver

De interrupt afhandelingsroutine van timer2 is int5. Flowchart van deze routine staat in fig. 6. Timer2 wordt geïnitieerd in de auto reload mode, met een repetitiefrequentie van 1200 Hz. Dat betekent dat in die routine omkeren van portpen P1.0, dat op die pen een blokgolf geeft van 600 Hz. Dat is de sidetone.

De pin P1.0 wordt laag gehouden als het gedetecteerde morsesignaal 'output' laag is, en als dat hoog is gecomplementeerd. De sidetone wordt dubbel gebruikt, er is namelijk ook een bit 'on' dat bepaalt of intern opgewekte morse uit het sidetone speakertje komt of uit staat.

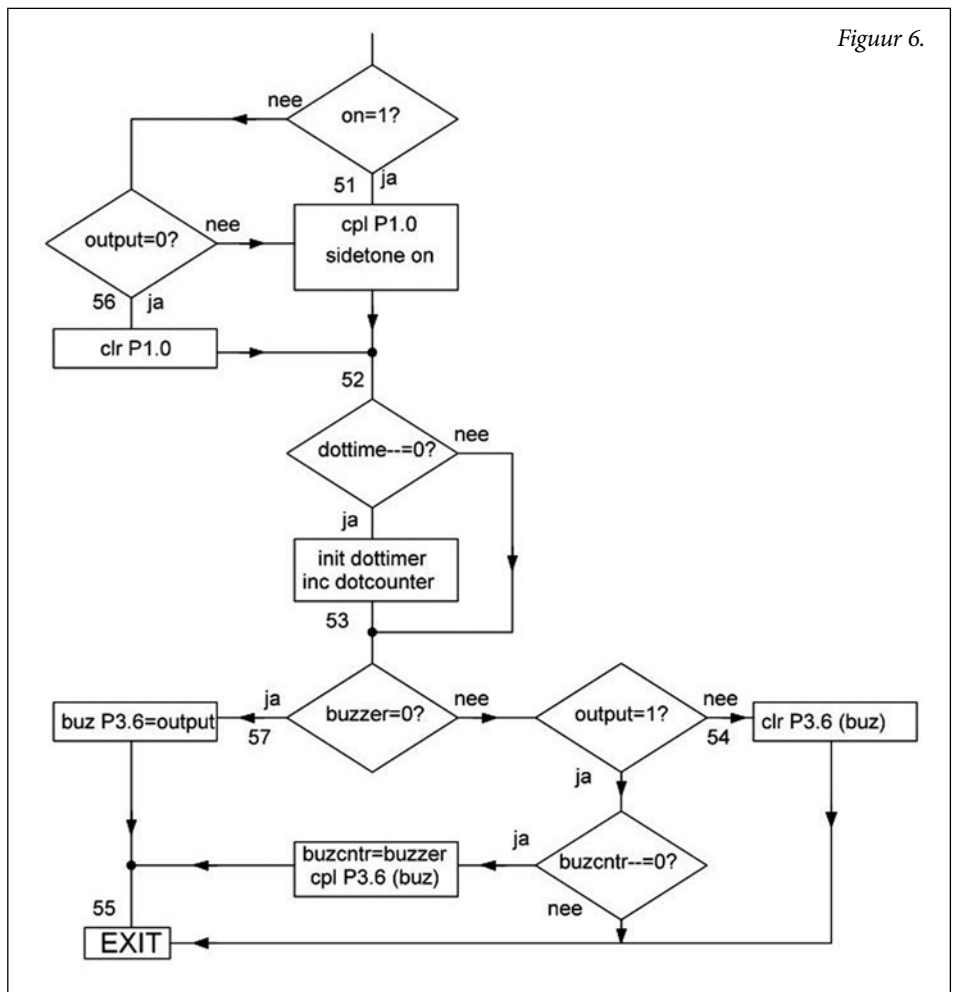
De intern opgewekte morse heeft een baudrate die bepaald wordt door een deler die het verhogen van een dotcounter bepaalt. Die dotcounter bepaalt de timing

alsmede tijdens de ontwikkeling van de software als diagnostisch gereedschap om de inhoud van geheugenposities te laten doorseinen.

Voorts is er een andere deler in int5 die de sidetonefrequentie deelt door een getal naar keuze tussen 1 en 12, en die bepaalt de buzzerfrequentie. De buzzerfrequentie kan ook 0 zijn, dat wil zeggen dat de buzzer beweegt zoals de arm van een seinsleutel doet, dus mark is laag en space is hoog. De buzzeroutput wordt via een aparte audioversterker naar de entree op het frontpaneel gevoerd.

EEPROM

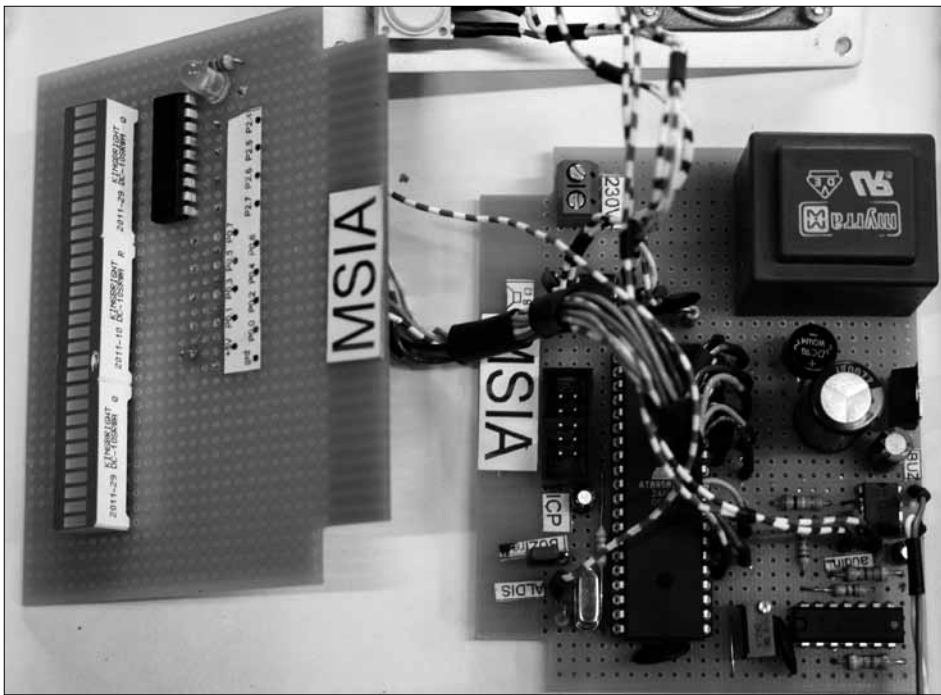
De controller bevat ook een EEPROM, dat wordt gebruikt om de gekozen buzzerfrequentie te bewaren. Als een jumper is geplaatst op P3.4 VOORDAT de netspanning wordt ingeschakeld wordt de buzzerdeeltal 1 verhoogd modulo 13. Dus altijd omhoog t/m 12 en dan terug naar 0. Er wordt een checksum berekend die ook in EEPROM wordt gezet. Zou die niet kloppen bij inschakelen van het apparaat, dan wordt een foutbericht in morse gegenereerd en worden er defaultwaarden



Figuur 6.

van de in het hoofdprogramma gegenereerde morse. Voor de snelheid is 25 wpm gekozen. Die wordt gebruikt voor foutberichten en om de frequentiekeuze van de buzzer op te geven aan de gebruiker,

in EEPROM gezet. In beide gevallen is het daarna 'system halted'. Je kunt dan opnieuw de waarde verhogen door de stekker uit en in het stopcontact te steken, of je kunt de jumper openen waardoor het



apparaat gebruiksgereed is met de laatst ingestelde buzzerfrequentie.

Morse generator

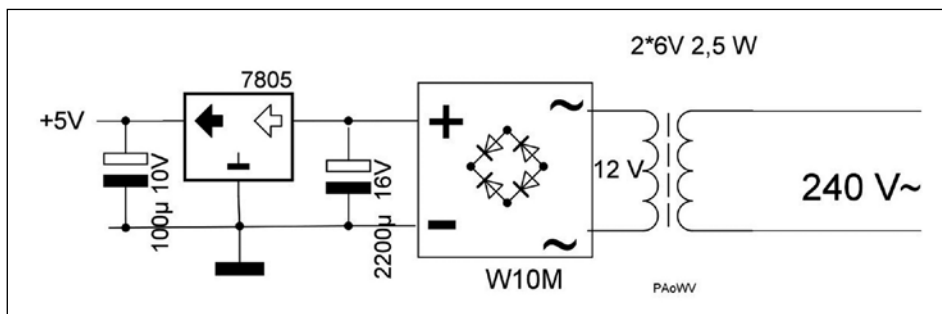
In het geheugen is een tabel opgenomen die iedere ASCII waarde vertaalt in de bijbehorende morsecode. De codering is in een byte met een 0 een dot een 1 een dash; en omdat de morseletters verschillende lengte hebben, daarachter een sluit_1, die aangeeft dat het karakter klaar is. De code wordt bit voor bit naar links geschoven, tot de bytewaarde 0x80 is, omdat de sluit1 link is aangeland.

Twee routines worden gebruikt, een voor het verzenden van een ASCII bericht zoals aangegeven door de datapointer, en een die een waarde van een byte omzet in een decimaal getal met onderdrukte voorloopnullen. Voor diagnostiek zijn er ook routines die geheugenwaarden van een en twee bytes in hex uitzenden.

De software is geprogrammeerd in assembler, en bestaat ongeveer 900 regels.

de schakeling. De opstelling van de onderdelen blijkt uit de foto's. Eerst wordt de voeding gemonteerd en gemeten of daar de gewenste spanning van 5 volt uitkomt. De soldeerpunten van de netaansluiting worden uit voorzorg van een klodder smeltlijm voorzien, om een opdonder te vermijden. 12 MHz kristal, controllervoet worden vervolgens gemonteerd. Meet tussen pen 20 (bij het kristal) (-) en pen 31 en 40 + of daar 5V staat. Dan de processor in de voet, niet een halve slag draaien en heel goed controleren of er geen pootjes zijn dubbel geklapt. De ISP connector uit het schema kan worden weggelaten, die is om ingesoldeerde processoren te kunnen (her)programmeren en is alleen tijdens de ontwikkeling van de software van belang geweest. De 3 LED IC's worden op de displayprint gezet, met de tekst in de richting van de edgeconnectorzijde van het bordje. Dat is de + zijde van de LED's.

De meest linkse LED is LED0 en ze moeten bedraad worden, zoals in het schema



Constructie

De schakeling is gebouwd op 2 experimenteer bordjes, formaat 8 maal 10 cm. De keuze voor 1 bordje van 10 bij 16 cm is opzettelijk niet gemaakt omdat de display dan wegvalt tussen de hogere onderdelen. Een printje bevat de display met driver en de Aldis LED, en het andere de rest van

in fig. 2, waarin de LED's genummerd zijn, staat aangegeven. Een kabeltje van 15 verschillende kleuren geïsoleerd soepel draad wordt gemaakt, om de bovenste print aan de onderzijde te verbinden met de onderste print aan de bovenzijde.

De bedrading van de LED's is gedaan met 0,3 mm geïsoleerd POSYN wikkeldraad,

en omdat dat nogal springerig is na montage gebundeld met ijzergaren.

Sommige onderdelen heb ik van Conrad betrokken www.conrad.nl, de bestelnummers daarvan zijn:

181544	TDA7052
335371	mini luidspreker 8 ohm
523232	alu kaartkastje
189677	40 pens IC voet
179205	7805ST
168726	kwartskristal 12 MHz
441682	elco 2200uF 16V
730986	aansluitklem 2 polig

Een geprogrammeerde processor inclusief verpakking en porto van 15 euro kan bij mij worden besteld per e-mail op mijn-call@amsat.org.

Afregelen

Zonder ingangssignaal kun je door aan de multivoltmeter te draaien een punt vinden dat de sidetoon net niet begint. Dan een stukje terugdraaien, 1/8 slag ongeveer, zodat het stil is.

Testen

Testen kan het beste gebeuren door te luisteren naar de sidetoon, omdat die het gedetecteerde morse representeert. Het volume op de ingang moet zodanig zijn dat dat foutloos klinkt.

Voorts kan een uit een computerprogramma zoals JustlearnMorsecode een serie van dezelfde letter worden aangeboden met diverse snelheden. Dat levert een periodiek signaal op de one shot LED display en is dus makkelijk te herkennen of dat conform de ontvanger letter foutloos gebeurt.

Literatuur:

[1] Tom G. Philippi "Benefits of multisensory presentation in perception, memory and navigation" Dissertatie Universiteit Utrecht fac. Computer science and Informatics Maart 2012.

