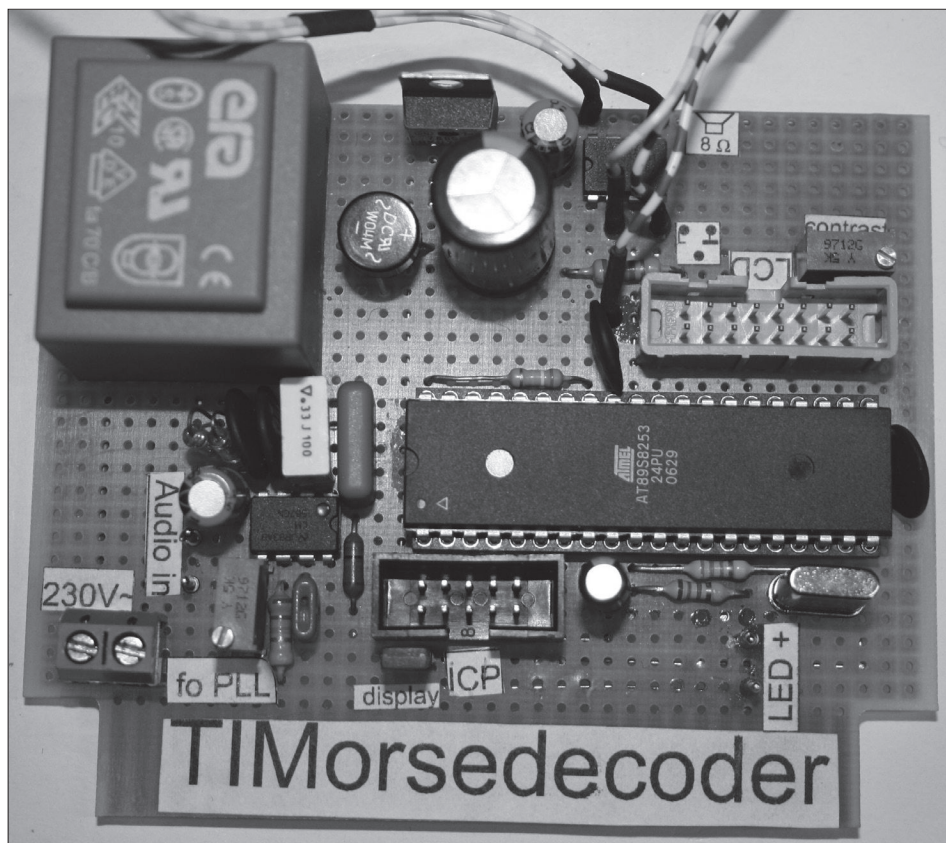


De TIMorsedecoder

door Wim Kruyf PAoWV

Morsedecodeer programma's kunnen handig zijn als ondersteuning bij contesten, DX-ing en Rag Chew QSO's. Maar ook kunnen ze prima gebruikt worden bij het leren van morse.

Wim beschrijft in dit artikel een stand alone morsedecoder, die rechtstreeks op de audiouitgang kan worden aangesloten, zodat we geen PC nodig zijn.



Er zijn zendamateurs, en niet zo weinig, die helemaal geen CW op het gehoor kunnen nemen en toch aan CW contesten en dergelijke meedoen. Vooral met contesten, omdat PC-programma's zoals CWget en Skimmer dan slechts machineschrift krijgen te verwerken, waar die alleen maar goed in zijn.

De reclame voor Skimmer bevat ook stevast decoderingsvoorbeelden tijdens contesten.

Nu zijn er mensen die door een opgelopen handicap uiterst moeilijk morse opnemen kunnen leren, seinen gaat prima, en die kunnen wellicht een stap verder komen en zeker QSO's maken met een decoder.

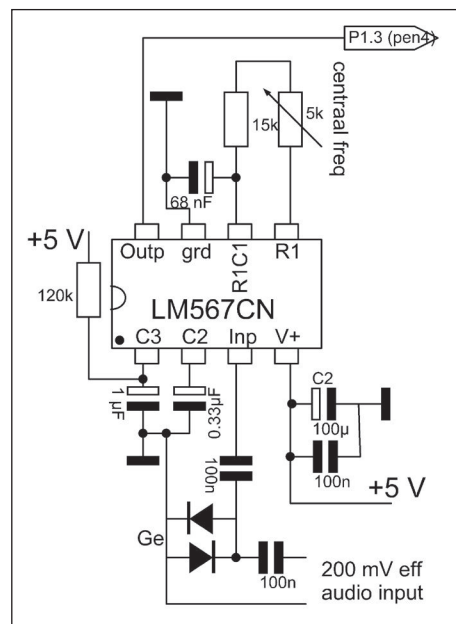
Ik wist niet dat er mensen bestonden die geen morse kunnen leren, anders dan door smoesjes vanwege gebrek aan wilskracht en doorzettingsvermogen (98% van de aanmelders zoals blijkt uit tellingen op

inwww.lcwo.net), maar PA4TIM heeft me duidelijk gemaakt dat dat wel het geval is, ten gevolge van een ongeval dat hem een blijvende hersenbeschadiging opleverde.

Voorts zijn er QRQ CW amateurs die rag chewen in morsecode boven de 70 wpm en die seinen dus met een keyboard, want dat gaat een bug echt te snel, die stellen soms dat ze op hoge snelheid zijn gekomen door tijdens een leerperiode mee te lezen op een decoder.

Zie het verhaal op <http://sites.google.com/site/tomw4bqf/copyingcwover70wpm>.

Paul PA3AQL, de webmaster van www.hellemonster.com bevestigde mij ook dat hij op die wijze het ham high speed domein, dus QRQ vanaf 25 wpm, betreden had en dat nu zonder hulpmiddelen handhaaft.



Het PLL filter.

Deze verhalen staan aan de wieg van de geboorte van de hier gepresenteerde morsedecoder, speciaal ontwikkeld en gebouwd voor en in samenwerking met PA4TIM.

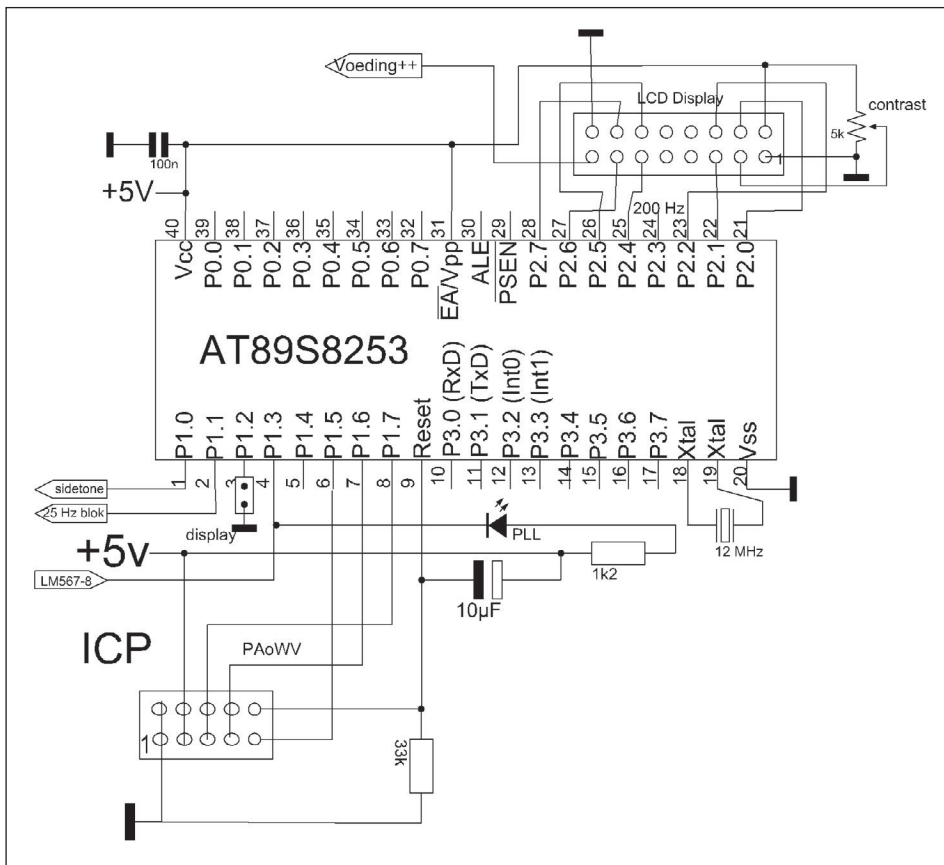
Er is voorzien in een directe sleutelinput, zodat je je eigen schrift kunt controleren als je beter wilt leren seinen. Dan heb je bovendien het voordeel, omdat er ook een gefilterde sleuteloutput is, dat je een QRP zendertje zonder contact bounce van je sleutel kunt gebruiken, dat vermijdt sleu-telklik op de band.

Het ontwerp

De decoder bestaat uit twee stukken, een ingangscircuit dat de signalen uit de ontvanger met een phase lock loop in de vorm van een touch tone decoder IC LM567CN (Conrad, kost 95 cent) decodeert tot blokgolven, en een schakeling met een controller die de morse snelheidsonafhankelijk decodeert en presenteert op een 2 maal 16 karakter LCD display. Andere lengten displays kunnen ook gebruikt worden, de displaylengte is als een constante in het programma wijzigbaar voor assembleren. Ik heb een tweeregelig 40 karakter per regel type geprobeerd en dat werkt prima, maar dat eist een bredere kast dan ik beschikbaar heb.

PA4TIM heeft uitgebreid geëxperimenteerd met het ingangscircuit en deelde mee tot resultaten te zijn gekomen die nauwelijks onderdoen voor menselijke waarneming in QRM. Zijn resultaten zijn te vinden op zijn website www.pa4tim.nl/?p=2076.

De ingangsschakeling met de tone decoder is zover ik kan nagaan eerder gebouwd door IK3OIL en gepubliceerd in QST 1999, zie ook www.ik3oil.it, en daarna zonder bronvermelding een aantal keren door anderen gepresenteerd, indachtig het adagium dat succes vele vaders heeft.

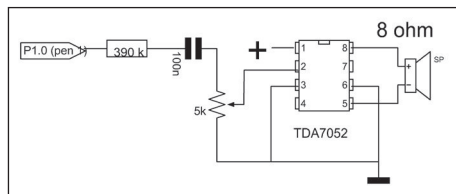


Het schema van de TIMorsedecoder.

Voor het testen heb ik met een daarvoor geschreven programma in C een aantal geluids.wav bestanden gemaakt met 4 kHz brede witte ruis en een 1 kHz CW signaal met 20 of 40 wpm en verschillende signaal ruisverhoudingen, die met de PC herhaald kunnen worden afgespeeld en dan ruis met signaal als audio aan het ingangscircuit wordt toegevoerd. Doordat dat signaal zich steeds herhaalt, het bestand heeft een speelduur van slechts 7 seconden, is de prestatiewijziging bij experimenteren met de parameters goed te beoordelen.

Het bestand timorse4.wav decodeert in mijn exemplaar foutloos, daaronder (meer ruis) treden fouten op. De testbestanden staan beschikbaar op het forum www.forum.vrza.nl, onder de rubriek CQ-PA.

PA4TIM heeft de schakeling uitgebreid met externe analoge filters en is gekomen tot foutloze decodering van timorse3.wav. Timorse2 lukte bijna, schreef hij, maar was niet foutloos te krijgen.

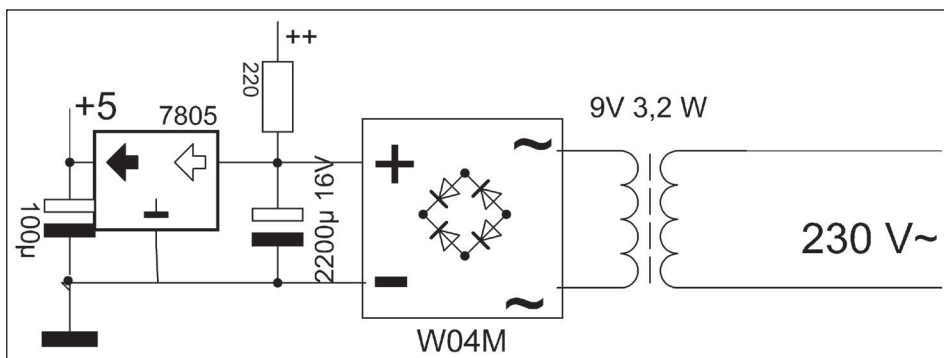


De optionele sidetone.

De werking van de digitale schakeling

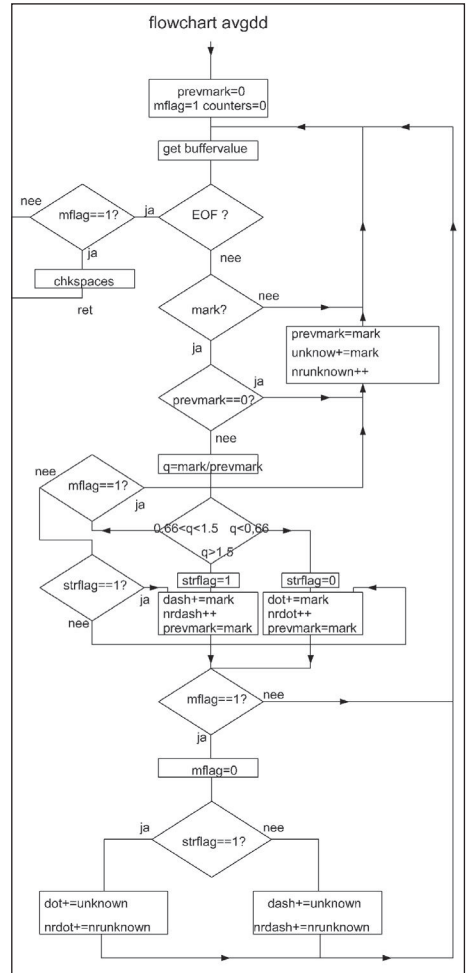
Van de ontvangen basisband CW signalen uit de PLL wordt de tijdsduur gemeten, zowel van de marks als de spaces. De bemonsterfrequentie van de tijdmeting is gekozen op 7812 keer per seconde. Dat is de minimale frequentie van de overflow interrupt van een 8 bits timer_0, diens interrupt afhandeling heeft tot taak een 16 bits tijdmetinggetal te verhogen. Een tweede timer_1 staat in een dertien bits mode, die levert een 25 Hz testsignaal in de vorm van een symmetrische blok op pen 2 van de controller.

Tevens een sidetoon van 1000 Hz op pen 1. Tot slot is een derde timer_2 in gebruik



die 1000 keer per seconde een 16 bits schuifregister doorschuift, het aantal enen in dat register bepaalt of er een 1 of een 0 als gefilterde input naar de signaalverwerking gaat.

Aldus wordt met een dergelijk FIR filter de haren die optreden ten gevolge van de ruis en het locken van de PLL op het signaal weg gefilterd. Dat is nodig omdat de haarpulsen anders als korte signaalelementen worden gezien en de decodering dan dus fout gaat.

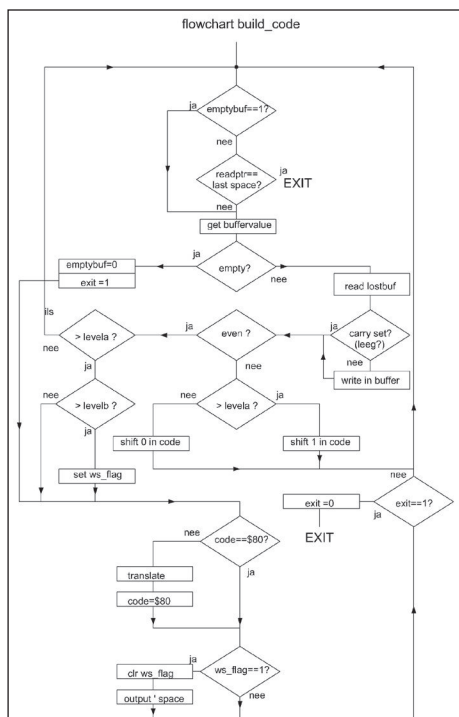


De tijdsduur van het signaal en rustelementen werden aanvankelijk via een externe interrupt_0 afhandeling bepaald en opgeborgen in een circular buffer van 16 posities van 2 bytes elk, die als hij vol is de tijdsduur van de laatst ontvangen 16 tekenelementen bevat, waarbij het meest significante bit 1 is gemaakt om aan te geven dat het tekenelement een mark was en 0 voor een space.

De maximale tijdsduur die gemeten kan worden treedt dus op bij het bereiken van 32767 op de tellerstand.

Dat betekent dat de minimale morsesnelheid 2 wpm is. De maximale snelheid wordt bepaald door de debouncetijd, en die is daardoor gelimiteerd op een snelheid van 98 wpm. Dat bouncen kan echter ook met het voorgeschakelde filter, zodat als die eruit gelaten wordt geldt "The sky is the limit". Dat is echter niet zinvol bruikbaar.

De processor herkent slechts externe interrupts die van hoog naar laag gaan, daarom was een ex-or gate nodig, waarbij de processor zorgt dat die buffert of invertiert. In een later stadium van de ontwikkeling is de ex-or 74LS86 eruit gelaten. De hele zaak met de ex-or wordt nu intern in de FIR-filter interrupt als nevenactiviteit afgewerkt. De uitgang van het digitale filter bepaalt of de sidetoon aan is, en vergeleken met de waarde die optrad bij de vorige interrupt wordt bepaald of er sprake moet zijn van een externe interrupt, die dan intern door het zetten van een vlag als software interrupt wordt geïnitieerd.



Uit de gemiddelde lengte van de strepen en punten worden de beslissingsniveaus bepaald om vast te stellen of er sprake is van punt, streep, interletterspatie, letterspatie of woordspatie, aan de hand van de gevonden beslissingsniveaus wordt de morse gedecodeerd en op de display gezet.

Dat beslissen gebeurt door elke ontvangen mark in de buffer te delen door de voorafgaande mark en als die een verhouding hebben kleiner dan 2/3 is het nieuwe element een punt en het vorige een streep. Is de verhouding groter dan anderhalf, dan is het nieuwe een streep en het vorige een punt, en tot slot tussen 2/3 en 1,5 wordt het element als hetzelfde gekwalificeerd als het vorige.

Dat lijkt een bewerkelijke zaak om twee 16 bits getallen te delen met een 8 bits processor, het delen is doorgaans het probleem niet, maar wel kan de ermee gemoeide tijd dat zijn, maar omdat we slechts willen weten of de uitkomst van de deling groter of kleiner dan de genoemde grensgetallen 2/3 en 1,5 is, is het een kwestie van een keer aftrekken een keer schuiven en nog een keer aftrekken om dat vast te kun-

nen stellen. Dat scheelt aanzienlijk in de tijdsduur en dat kan belangrijk zijn. In de paragraaf Metingen wordt dat nog nader bekeken.

Iedere keer moet je dus een buffer vol hebben om te kunnen decoderen, dat zijn dus totaal 16 punten, strepen en soorten spaties. De langste morsetekens bevatten 10 punten, strepen en spaties samen, en de kortste slechts 2, zodat er 1 tot 8 letters in de buffer passen.

Als de buffer vol is wordt de berekening en de decoding gestart tot de laatste letterspatie in de buffer, omdat de rest van de laatste letter niet bekend is. Dat kost tijd, inmiddels komen nieuwe tekens binnen en daarom is er een tweede buffer die de overflow tijdelijk opvangt.

Door deze methode is de decoding instantaan, je kunt dus de morsesnelheid sterk variëren, de decoding gaat gewoon correct door. Slordig seinen in een banana-boat ritme is door deze tactiek ook geen probleem.

Je zult echter lastig tegen een telegrafist op kunnen, want totaal vervormde morsetekens worden in de context goed gedecodeerd, bijvoorbeeld de uitnodiging om de sleutel te wisselen, de letter K, wordt vaak geseind als TET, de mens weet uit het verband van het bericht dat het K betekent, de machine niet, die legt slechts koude beslissingscriteria aan op grond van in het recente verleden ontvangen morsetekens.

Prosigns SK AR KN en AS en KA zijn aparte fonts voor op de display gemaakt. Niet gedefinieerde morsetekens, en dat zijn er nogal wat, worden als een * afgedrukt.

Om de PLL en de digitale schakeling te kunnen testen tijdens de ontwikkeling is een 60 wpm puntenriedel uit de chip te halen door 25 Hz output blok op P1.1 (pen 2) van de processor door te verbinden met de sleutelinput op P1.3 (pen 4).

De audio komt in de vorm van een reeks punten op 1000 Hz uit pen P1.0 (pen 1) van de processor, waar als optie ook een sidetoon audioversterker kan worden aangesloten, die in het schema is te vinden.

De phaselock decodeert het ingangssignaal dat via een C-tje wordt aangeboden, vanaf 100 mV, maar geeft voor signalen in de ruis en QRM veel spikes af, die worden door de processor geïnterpreteerd als korte punten. Dat stuurt dan natuurlijk de decoding in de war.

Daarom is er een digitaal laagdoorlatend post-detectiefilter geplaatst op de ingangspen 4 van de controller, de output daarvan gaat door een in de software geïmplementeerde schmidt-trigger met hysteresis en is voor metingen of sleutelen van een zender beschikbaar op pen 5 van de controller.

De grensfrequentie van het filter is te vergelijken met de klokfrequentie van timer_2. PA4TIM doet dat filteren met een tweetal op-amps, die als analoog tweede orde laagdoorlatend Besselfilter zijn geschakeld en tussen de PLL output en de eerste filterloze versie van het programma in de controller is geplaatst.

Een LED staat op de uitgang van de PLL, die geeft een indicatie of de gedecodeerde morse klopt met het geluid uit de ontvanger.

Bij teveel ruis is dat niet meer het geval. De sidetoon is de morse zoals die na het digitale filter beschikbaar is en daarna gedecodeerd wordt.

Er zijn twee display modi mogelijk, naar gelang de voorkeur van de operator. De eerste is een lichtkrant op de bovenste regel van de display, de laatst ontvangen letter staat dan altijd rechts boven, en op de onderste lijn staat de ontvangen morsesnelheid.

Bij de tweede display mode, is de snelheid niet zichtbaar maar wordt de onderste lijn van links naar rechts gevuld, en wipt een regel naar boven als die vol is, dat heb ik de printer mode genoemd, die leest wat rustiger omdat de letters op hun plaats blijven staan en je ziet dan gemiddeld ook meer letters op de display, namelijk 24 in plaats van 16, met een maximum van 32 letters. De keuze kan worden gemaakt met een jumper op pen 1.3 van de processor. De jumper geplaatst levert de printer mode.

De verwerking van de signalen geschiedt met 16 bits, er zijn bij sommeringen, voor bepaling van de gemiddelde tijdsduur van de signaalelementen, 24 bits getallen mogelijk, maar als blijkt dat het derde byte niet 0 is wordt een fatal error met een nummer op het scherm gemeld. Nooit gezien, maar mocht het voorkomen, bij extreem lage snelheden kan dat verwacht worden, dan kan ik de software erop alsnog aanpassen.

Ook overflow van de hulpbuffer zou een fatal error geven, ook dat gebeurt niet in de praktijk. Dus de zaak blijkt voldoende ruim in haar jasje te zitten.

Metingen en afregelen

Met een teller heb ik de output beatnote van mijn ontvanger gemeten als het CW filter in de ontvanger in staat. De bandbreedte heb ik bepaald door de ontvanger te verstemmen tot de S meter net een tikje begint te dalen. Dat levert een bandbreedte van 870 tot 1130 Hz, zodat ik de PLL zijn centraalfrequentie op 1000 Hz afregel.

Dat kun je doen door een teller met een scope probe te hangen aan pen 5 van de LM567, als er geen inputsignaal is (audioingang kortsluiten) en de trimpot dan te verdraaien tot de teller de centraalfrequentie van je CW filter aanwijst. Bij mij is dat dus 1000 Hz.

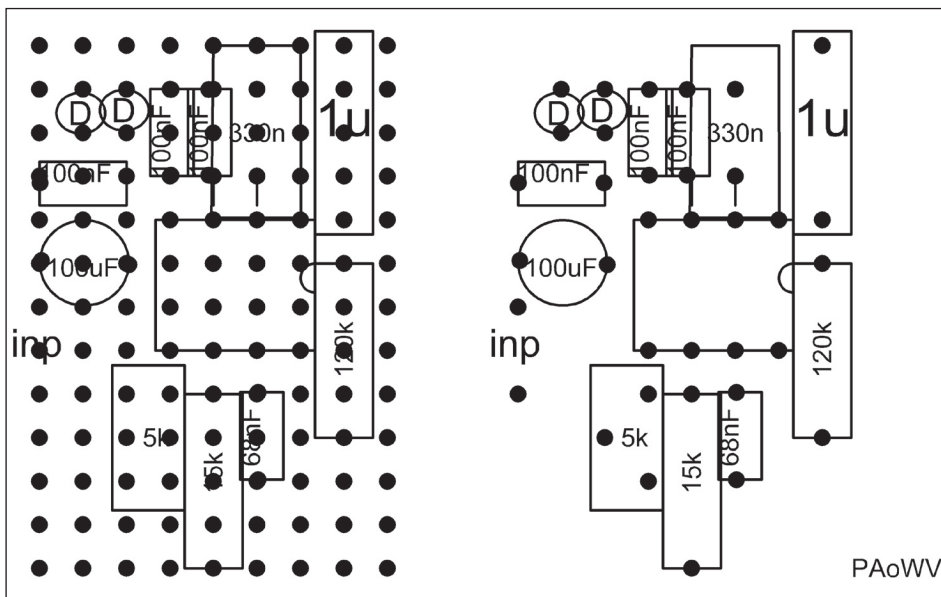
Voor het bekijken van de detectie van de PLL heb ik daarom een 1000 Hz sidetoon uit de controller gekozen op P1.0, die door een testdraadje te spannen tussen de 25 Hz blokoutput op pen 2 en de morseinputpen 4 een morse signaal, bestaande uit uitsluitend dots, van 1000 Hz toonhoogte genereert met een snelheid van 60 wpm.

De tijdsduur van de hele decodering van de buffer is ook gemeten inclusief een startdeling voor het berekenen van de snelheid, en dat blijkt alles bij elkaar 2,5 ms te duren. Een fractie dus van de kortste morsetekens. Dat meten gebeurt door bij het begin een outputpen op de controller hoog te maken en aan het eind weer laag, dan kun je op een aangesloten scope de tijdsduur meten dat hij in die routine zit inclusief de onderbrekingstijden door de vier interruptroutines.

De processorbelasting van de timer_0 interrupt voor het ophogen van de elementduurteller; de externe interruptroutine die door het zetten van de flag door de FIR filter interruptafhandeling intern wordt gestart, en de timer_1 interrupt die de 25 Hz testsignaal en 1000 Hz sidetoonlevert kunnen gemeten worden door aan het begin van de interrupt afhandeling routine een processorpen laag te maken en aan het einde weer hoog.

Vervolgens pak je een 10k potmeter en een 1 mA draaispoelmeter. Schakel de potmeter, die als regelbare weerstand is geschakeld door de looper met een der uiteinden te verbinden op maximale weerstand, in serie tussen +5V en massa, en regel de potmeter zo af dat de meter net vol uitslaat.

Daarna de massakant losmaken en verbinden met de processorpin die alleen laag is tijdens interruptafhandeling, en zie daar de meter wijst de processorbelasting door de interruptafhandelingen aan, gemeten in procenten met volle schaaluitslag=100%.



De onderdelenopstelling van de PLL.

Nabouw

Nabouw is simpel. De zaak is gemonteerd op gaatjesboard 10 cm breed 8 cm lang (half euroformaat) zodat een standaard kastje eromheen past. De tekeningen van frontje en achterzijde, zijn in een pdf file te vinden op www.forum.vrza.nl onder de rubriek CQ-PA, als die op een printer worden afgedrukt en uitgeknipt kunnen die dienen om de maten van de onderdelen over te nemen op het aluminium met een kraspenpunt; vervolgens de gemaakte putjes centeren, voorboren met 2 mm en naboren met de gewenste afmetingen.

Voor de print geldt: Eerst de voeding maken en de spanning controleren. Wat smeltijm op de soldeeraansluitingen van de netspanning deponeren is nooit weg. Dan de processor ICvoet monteren en kijken of de spanning op pen 31 en 40 +5 is en pen 20 grd. Versterkertje TDA7052 monteren, en de PLL, de opstelling van de onderdelen blijkt voldoende duidelijk uit de foto en van de PLL is die apart op rasterformaat getekend want die is nogal compact door het woekeren met de beschikbare ruimte. De processor vervolgens plaatsen, heel goed opletten dat er geen poten dubbel gaan en dat pen 1 op de juiste plek zit. Niet een halve slag draaien dus. De display eraan. De contrastregelaar trimpot verdraaien tot de vlakjes van de letters te onderscheiden zijn.

Dan kan voordat de PLL is aangesloten al geprobeerd worden of de zaak werkt door 25 Hz uit pen 2 van de processor tijdelijk te verbinden met de morseinputpen op pen 4. Je moet dan een snelle riedel punten horen, en de tekst verschijnt op de display terwijl de snelheid op de onderste regel staat (59 à 60 wpm) als de jumper op pen 3 niet geplaatst is.

De ICP connector is nodig om de controller te programmeren, heb je een gepro-

grammeerde controller dan kan die worden weggelaten.

Het display is een normaal 2 regelig 16 letter HD44780 compatible geval, zoals verkocht door Dijken in Groningen en Baco in IJmuiden. Heb je een ander merk dan de boortekeningen aanpassen. Indien een andere regellengete gewent wordt, dat opgeven bij het programmeren van de controller.

Je kunt natuurlijk zelf de connector voor audio input kiezen, ik had geen cinch (tulp-) connector liggen, en daarom heb ik twee stekkerbussen gemonteerd. De radiowinkel op de hoek en op bijna alle andere hoeken is reeds lang verdwenen. Desalniettemin blijven de evangelisten met holle stem verkondigen dat zelfbouw niet dood is.

Gebruik met een normale seinsleutel om het eigen schrift te controleren kan door de seinsleutel aan te sluiten op pen 8 van de PLL, volgens het wired or principe en de open collector output van de PLL kan dat.

Eventueel kun je dan een zender sleutelen door die met een transistor interface te bedienen vanuit pen 5 van de controller waar de bounce vrije output van de sleutel beschikbaar is, 0 is key down.

Onderdelen die ik heb gebruikt en die bij Conrad verkrijgbaar zijn:

181544	tda 7052 1W audio
177202	LM567
335371	Mini ldspr 8 ohm
523232	Al kaartkastje euro

Een geprogrammeerde processor kan ik leveren voor 12 euro excl. verzendkosten. Neem dan contact op, ook voor eventuele vragen, via email met mijnCALL@amsat.org.

73, PAoWV

