

Audio-spectrumanalyzer met dekatrons

nieuw leven voor vintage-buisjes



Charles van den Ouweland (Nederland)

Het is altijd leuk om nieuwe toepassingen te vinden voor vintage-onderdelen, vooral voor displays. Meestal zijn ze te mooi om ze in een la te laten liggen. Dit audio-analyzer-project is een voorbeeld. We geven dekatrons een tweede leven.

Dekatronen zijn gasgevulde telbuizen uit de jaren vijftig van de vorige eeuw [1]. Gedurende een korte periode in de elektronicageschiedenis werden ze gebruikt als tellers en geheugencellen in computers, zoals de Harwell Witch-computer [2]. Elke buis kon één cijfer van 1 tot 10 bewaren, dus om een computer te bouwen, waren heel wat van die buizen nodig. Ik had er een paar liggen en dacht na over een project waarin ik ze zou kunnen gebruiken. In nixie-klokken is soms één dekatron ingebouwd als secondeteller. Dat is leuk, maar ik wilde iets nieuws.

In dit artikel presenteer ik een 7-bands stereo audio-spectrumanalyzer met dekatron-display. Dit is een leuke aanvulling voor audiofielen met een buizenversterker die op zoek zijn naar visueel spektakel!

Het idee is simpel: het audiospectrum wordt

niet weergegeven als een staafdiagram, niet met LED-displays of (nog saaier) met een LCD; in plaats daarvan wordt het volume van het audiosignaal in elke frequentieband weergegeven door het aantal oplichtende punten in één van de dekatrons. In elke dekatron zijn de tien lichtpunten in een cirkel gerangschikt. De linkerhelft daarvan vertegenwoordigt het linker kanaal en de rechterhelft het andere.

De hardware van mijn audio-spectrumanalyzer (**figuur 1**) bestaat uit een besturingseenheid voor de signaalverwerking en zeven identieke dekatron-eenheden voor het display. En natuurlijk is er ook een voeding nodig.

De besturingseenheid

In **figuur 2** zie je het schema van de besturingseenheid en de dubbele voeding voor

de spectrumanalyzer. Ik had aanvankelijk een oude print omgebouwd voor mijn eigen prototype van de schakeling. Inmiddels zijn de ontwerpfiles (Eagle-bestand `msgseq7ctrlr_Elektor.sch`) en print-layout beschikbaar op de GitHub-pagina van dit project [3].

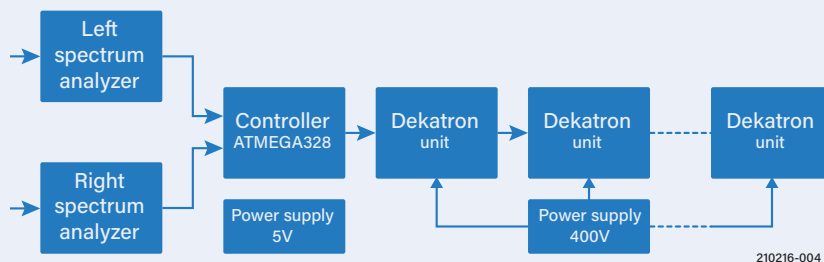
De microcontroller (IC1) is een ATmega328P, bekend van de Arduino. Hij wordt hier echter niet gebruikt als een Arduino. Het kleine programma (300 regels) is met Atmel Studio geschreven in C.

Om het audiospectrum te bepalen, gebruiken we een gespecialiseerde chip: de MSGEQ7 van het Amerikaans bedrijf Mixed Signals. Deze analyseert het audiosignaal en verdeelt het in zeven frequentiebanden. De microcontroller kan de zeven niveaus uitlezen door digitale commando's naar de chip te sturen. De chip reageert daar op met een spanning op zijn uitgang, die naar de A/D-omzetter in de microcontroller wordt doorgegeven. Er zitten twee analyzer-IC's in de schakeling (IC2 en IC3), voor elk audiokanaal één.

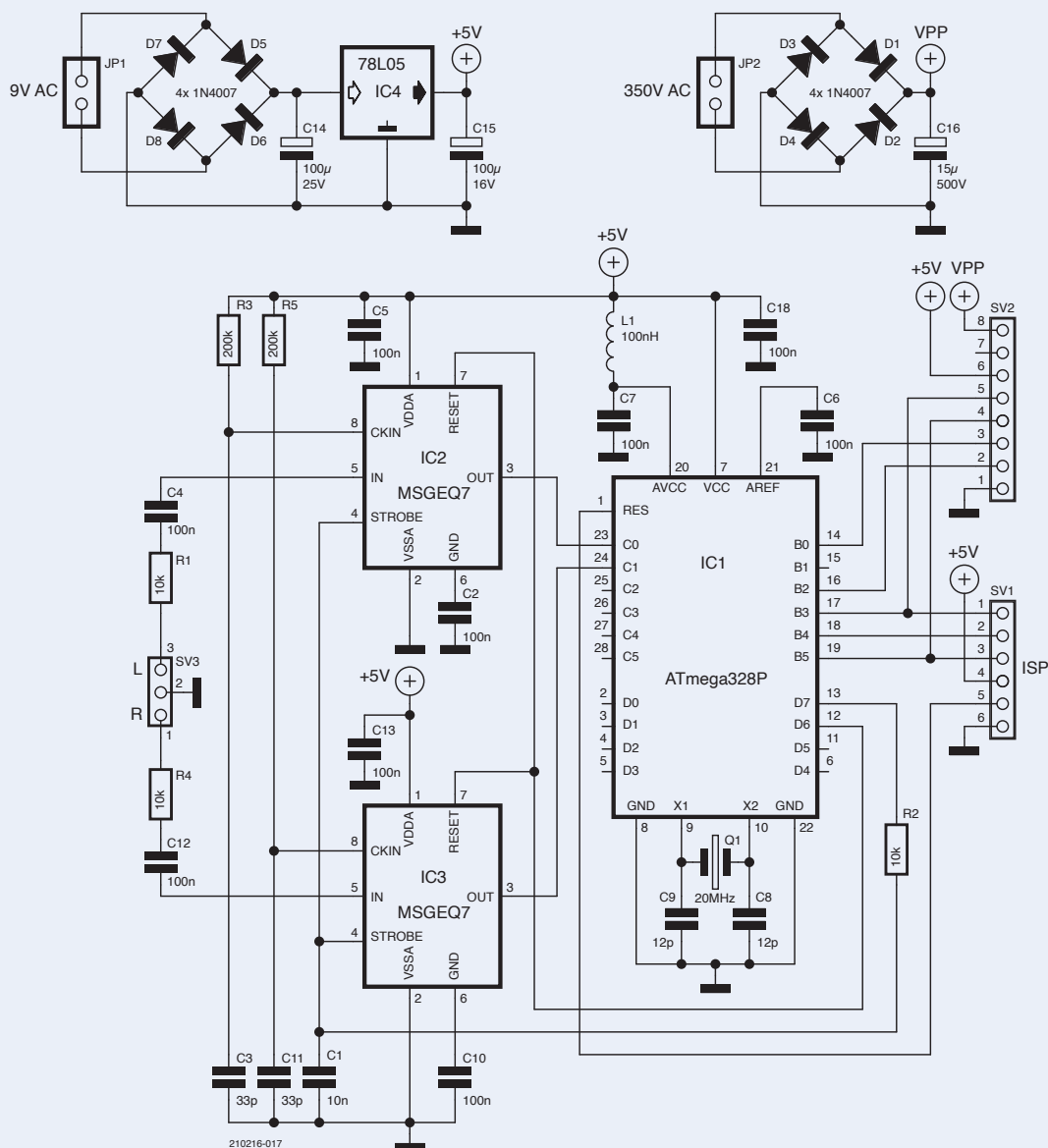
Pas op, hoogspanning!

Dekatronen werken met spanningen tot 450 V. Als je niet vertrouwd bent met het werken met hoge spanningen, begin dan niet aan dit project en werk niet met deze buizen. Zelfs als de netspanning is uitgeschakeld, kan er nog steeds een gevaarlijke spanning over de hoogspanningscondensator staan. Als je aan de schakeling gaat werken, vergeet dan niet die condensator eerst te ontladen.

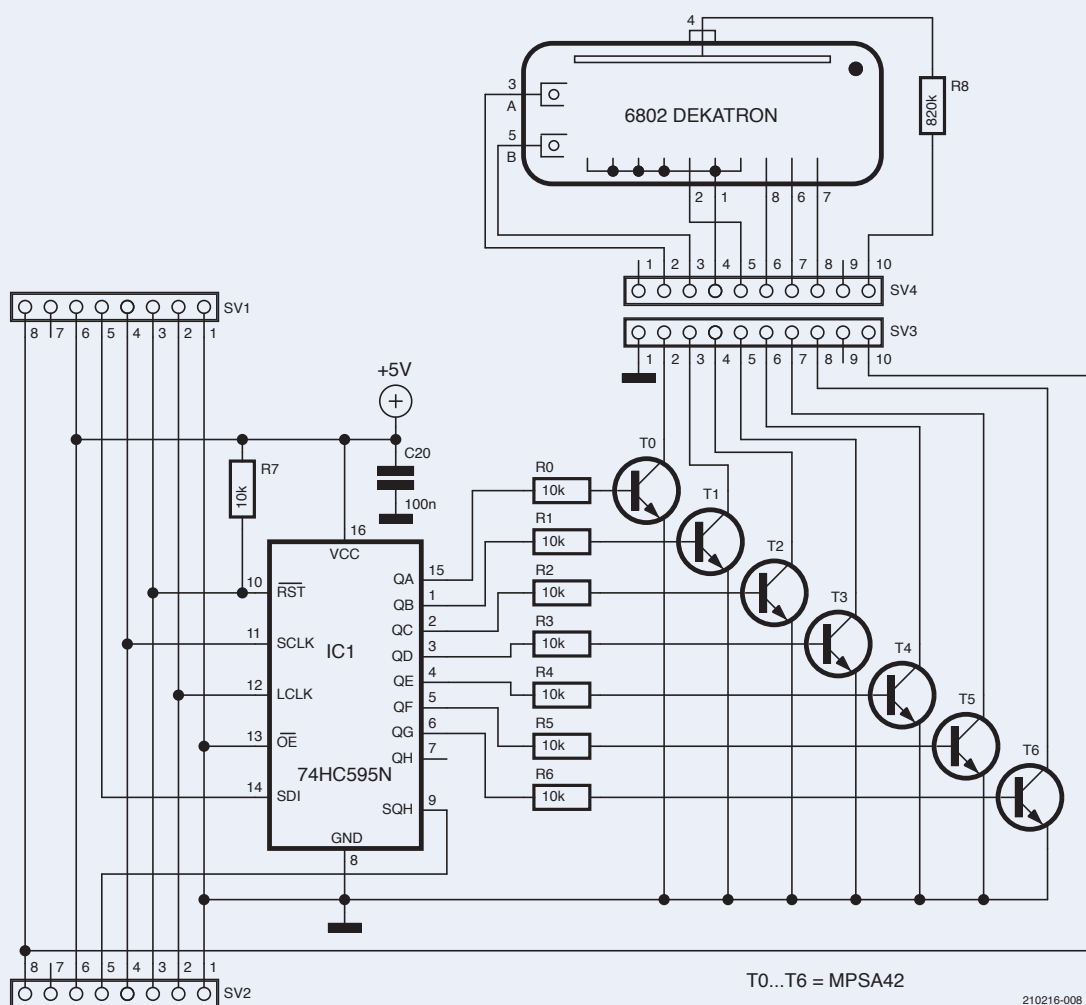
Noch de ontwerper noch Elektor accepteren aansprakelijkheid voor eventuele schade die het gevolg is van het bouwen, testen en gebruiken van (delen van) dit project!



Figuur 1. Blokschema van de audio-analyzer.



Figuur 2. Schema van de besturingseenheid.



Figuur 3. Schema van één dekatron-eenheid.

De dekatron-eenheid

De microcontroller communiceert met de dekatrons via een driedraads seriële interface (SPI, dus) met een snelheid van 5 Mbit/s. Elke dekatron zit op een printje met een 74HC595 (serieel-in, parallel-uit schuifregister) en een aantal MPSA42-transistoren als hoogspanningsdrivers.

Ik heb een printje van $4 \times 4,5$ cm voor deze units ontworpen, of eigenlijk twee printjes want de eenheden bestaan elk uit een sandwich van twee printen. In **figuur 3** zie je het schema. De bovenste print bevat een voetje voor de dekatron en de anodeweerstand, en de onderste bevat alle andere componenten. De gaten voor het dekatronvoetje zijn geschikt voor verschillende types (zowel keramische als plastic voetjes). De onderste print heeft links een male connector SV1 en rechts een female connector SV2, zodat ze gemakkelijk op elkaar geprikt kunnen worden tot een

complete display-eenheid. In **figuur 4** zie je enkele complete dekatron-modules klaar voor een test.

Printfabrikanten bieden vaak een prototype-service (tien printen voor een redelijke prijs) dus je kunt goedkoop printen voor 10 eenheden laten maken. De Gerber- en boorfiles die je nodig hebt voor de bestelling zitten in *Unit6802_Elektor.zip* [3].

De voeding

Er zijn twee voedingsspanningen nodig: 5 V en 400 V. De elektronica voor de voeding is te zien in figuur 2. De 5V-voeding is eenvoudig: een 9V-transformator met een bruggelijkrichter, een 78L05-spanningsregelaar en enkele condensatoren.

In de dekatrons vindt gasontlading plaats. Daarvoor is een vrij hoge V_{PP} -spanning van ongeveer 400 V nodig. Mijn eerste oplossing was een schakelende voeding, opgebouwd

rondom een MC34063 of zelfs een eenvoudige 555. Maar ik ontdekte dat zo'n voeding veel storing veroorzaakt bij de analoge signalen (vooral aan de uitgang van de MSGEQ7). Het lukte me niet om daar goede resultaten mee te bereiken. Toen vond ik een leverancier in China, Yollen Electrical Store, die transformatoren bouwt volgens klant-specificatie. Ik bestelde een aantal kleine 50Hz-transformatoren voor 350 V, 1 VA, voor € 3 per stuk plus € 6,50 verzendkosten (**figuur 5**). Met vier 1N4007-diodes (D1... D4) en een condensator van 10 μ F/500 V (C16) was ik klaar.

Hoewel deze schakeling met buizen werkt, is het vermogensverbruik heel bescheiden. Dekatrons zijn zogenaamde koude kathode-componenten, dus ze hebben geen gloeidraad! Elke buis gebruikt slechts 0,3 mA anodestroom aan de hoogspanningskant, dus een vermogen van ongeveer 1 W. Beide trans-

formatoren hebben een capaciteit van 1 VA en het totale vermogensverbruik is minder dan 2 W.

Software

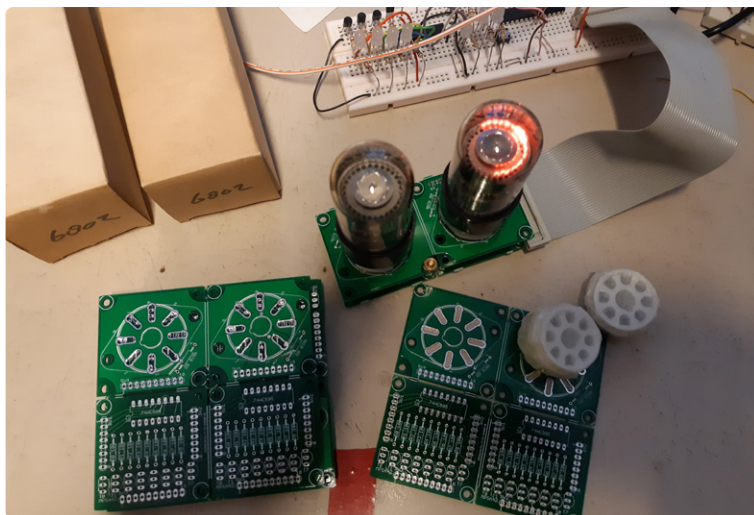
De software voor de ATmega328 is (in Atmel Studio) geschreven in C; de broncode en het HEX (adecimale) bestand voor het programmeren van de microcontroller kunnen worden gedownload van de GitHub-pagina [3]. Het werk wordt gedaan in de hoofdloop en in interruptroutines. Het interfaceren met de analyzerchips (inclusief het aansturen van de ADC) gebeurt in de hoofdloop, het interfaceren met de dekatrons via SPI wordt gedaan door de interruptroutines.

Het linker en rechter audiosignaal gaan naar de beide analyzerchips. Die bevatten elk zeven banddoorlaatfilters (60 Hz, 150 Hz enzovoort, zie **figuur 6**), en het signaalniveau in elke band wordt bemonsterd. De AVR krijgt de audio-sigitaalniveaus als analoge signalen, digitaliseert ze en gebruikt de dekatrons als display.

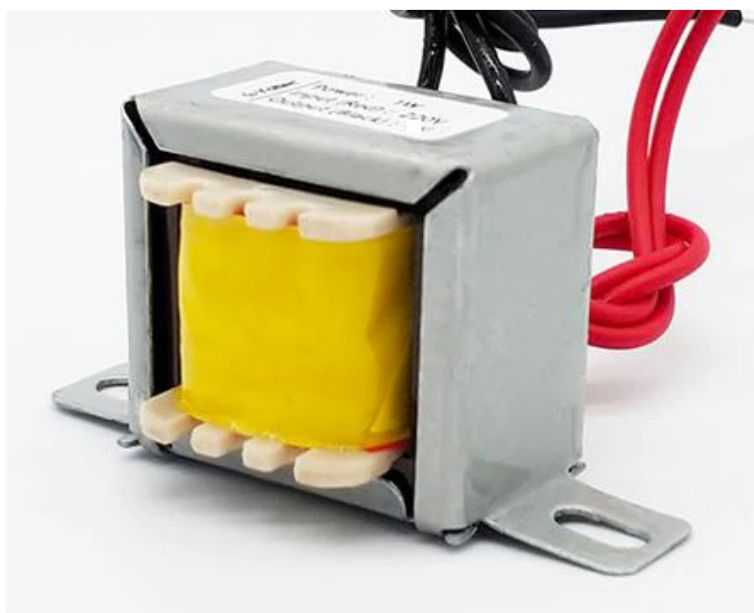
Data-acquisitie

Om de invloed van stoorsignalen te onderdrukken, worden de waarden 15 keer gemeten en worden de beste waarden gebruikt. Omdat er zeven frequentiebanden en twee kanalen zijn (L en R), vinden in elke cyclus 210 A/D-omzettingen plaats, voordat het display wordt ververscht!

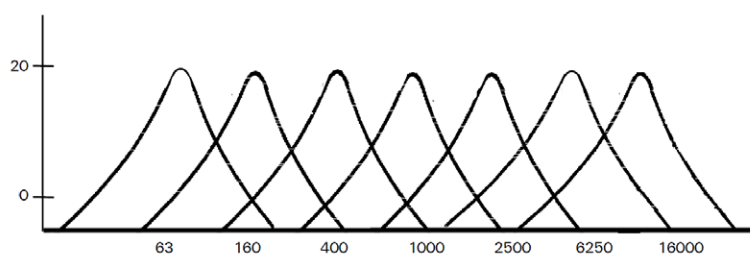
In detail gaat dat als volgt. Eerst worden beide MSGEQ7-analyzerchips gereset. Dan krijgen beide chips een strobe-puls. Het niveau van de eerste frequentieband verschijnt nu op pin 3 van de chips. Er wordt een A/D-omzetting uitgevoerd op het linker en op het rechter kanaal. Beide omzettingen worden tweemaal herhaald, zodat we drie waarden voor links en drie waarden voor rechts krijgen. De hoogste en de laagste waarde worden weggegooid, en alleen de middelste waarden worden bewaard. Dan krijgen beide IC's een tweede strobe-puls, zodat ze overgaan naar de tweede frequentieband en het bemonsteringsproces wordt herhaald. De derde tot en met zevende frequentieband krijgen dezelfde behandeling. Deze data-acquisitie cyclus wordt vijf keer herhaald en alleen de laagste waarden worden gebruikt omdat ik heb



Figuur 4. Printen en opgebouwde eenheden.



Figuur 5- De 350V-nettransformator.

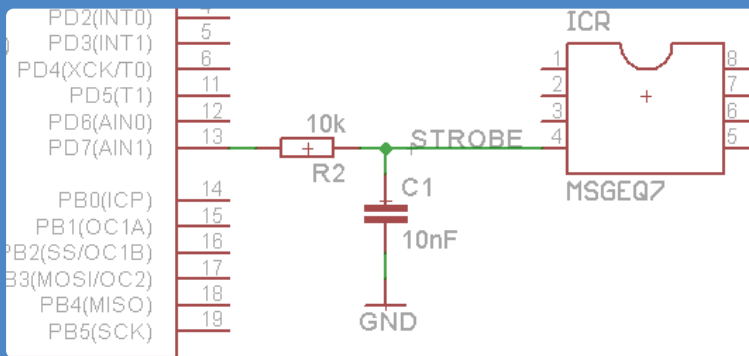


Figuur 6. De frequentiebanden van de MSGEQ7 (bron: Mixed Signal Integration).

MSGEQ7... echt of namaak?

Pas op voor namaakchips! Ik heb enkele MSGEQ7's gekocht van een verkoper op AliExpress en van een verkoper op EBay en die bleken allebei namaak te zijn. De exemplaren die ik bij Sparkfun heb besteld voor \$ 4,95 per stuk waren echt. Het is niet zo moeilijk om te bepalen of je een namaak- of een echte chip hebt. Sluit 5 V aan tussen pen 1 (VDD = +5 V) en pen 2 (VSS = 0 V) en meet de stroom. Een echte MSGEQ7 gebruikt minder dan 1 mA. De namaakchips die ik had, trokken meer dan 10 mA.

Uitgangsruijs van de MSGEQ7



Deze chip is al besproken in Elektor mei/juni 2019 [5], maar wat er niet bij stond is hoe je moet omgaan met de ruis op de uitgang.

Mijn eerste experimenten met de chip waren niet erg geslaagd vanwege die ruis. Ongeveer één van elke tien gemeten waarden was gewoon helemaal fout. Op het internet zie je veel anderen die worstelen met hetzelfde probleem. Ik heb een aantal maatregelen genomen om dit te verhelpen.

Eerst heb ik geprobeerd het softwarematig op te lossen, door dezelfde waarde meerdere keren te bemonsteren en zo meerdere metingen van de chip te krijgen. Ik gooide de hoogste en laagste waarde weg en nam het gemiddelde van de rest van de samples. Dat hielp een beetje. Ik heb ook verschillende hardwaremaatregelen geprobeerd.

Om te beginnen heb ik geprobeerd de afstand tussen de microcontroller en de MSGEQ7 te verkleinen. Dat hielp helemaal niet. Ik heb ook geprobeerd de voeding voor de microcontroller en voor de MSGEQ7 van elkaar te scheiden, maar het was tevergeefs. Toen ben ik afgestapt van de schakelende voeding en ik ben overgegaan op een zeer eenvoudige voeding met ouderwetse transformatoren, diodes, condensatoren en een 7805. Dat hielp een beetje. Vertwijfeld probeerde ik een RC-filter in het digitale pad van het STROBE-sigitaal. En tot mijn grote verrassing was het probleem toen opgelost! Je kunt het vinden in het schema: C1 en R2 vormen een eenvoudig RC-filter dat natuurlijk van STROBE een afschuwelijk digitaal signaal maakt, maar de MSGEQ7 accepteert het en de ruis is (vrijwel) verdwenen!

gemerkt dat de hogere waarden doorgaans alleen maar storing zijn.

Weergeven van de spectra

De besturing van de dekatrons gaat als volgt. Wat het programma betreft, kent elke dekatron 60 toestanden. Waarom 60? Een dekatron moet toch 10 toestanden hebben (vandaar 'deka')? Inderdaad, de dekatron heeft 10 stabiele toestanden. Maar tussen elke twee stabiele toestanden zijn er nog twee overgangstoestanden. In totaal heeft de dekatron 30 kathodes, 10 voor de stabiele toestanden en 20 voor de overgangstoestanden. De dekatron was oorspronkelijk bedoeld als een telbuis. Door een verschoven puls op de eerste en dan op de tweede transitiekathode te zetten, 'telde de buis één stap'. Dat betekent dat het bij een dekatron gemakkelijk is om het lichtpuntje één positie op te schuiven. Maar het is niet zo gemakkelijk (en meestal zelfs onmogelijk) om het lichtpuntje naar een willekeurige andere

positie te brengen. Dat is alleen mogelijk voor enkele van de stabiele posities. Bij sommige dekatrons kunnen alle tien stabiele posities direct worden aangestuurd maar bij andere zijn alleen de posities 0, 5, 8 en 9 beschikbaar op de aansluitpennen. In dit ontwerp wordt de 0-positie gebruikt als beginpunt en daarna beweegt de punt zich alleen naar aangrenzende posities, steeds één stap naar links of één stap naar rechts.

We gebruiken bij deze toepassing alle 30 kathodes. Het is ook mogelijk om twee naburige kathodes tegelijk te activeren en dat wordt gebruikt als een overgang tussen de ene kathode en de volgende. Zo komen we uit op 60 toestanden.

We schakelen bij deze toepassing heel snel om tussen verschillende toestanden, zo snel dat het oog wordt gefopt. De huidige toestand van dekatron *i* (een getal tussen 0 en 59) is opgeslagen in *values[i]*. Deze wordt snel gevarieerd tussen twee uiterste

waarden, zo ongeveer als een ruitenwisser (**figuur 7**). Deze extreme waarden zijn opgeslagen in *minima[i]* en *maxima[i]*. De huidige bewegingsrichting (naar minima of naar maxima) staat in de Boolean *orient[i]*. De toestand wordt elke 150 µs veranderd. Als 150 µs zijn verstreken, wordt interrupt-serviceroutine *TIMER0_COMPA_vect()* aangeroepen. Op dat moment staan de nieuwe waarden die naar de dekatrons moeten worden verzonden al klaar in *buffer[]*. De eerste waarde uit *buffer[]* wordt via de SPI-bus verzonden met een snelheid van 5 Mbit/s. Als deze SPI-overdracht klaar is, wordt de interrupt-serviceroutine *SPI_STC_vect()* aangeroepen, die het volgende byte over de SPI-bus stuurt. Als alle zeven bytes zijn verzonden, berekent *SPI_STC_vect()* de nieuwe *values[]*. De waarde van *buffer[]* wordt berekend; deze moet het bitpatroon bevatten dat over de SPI-bus wordt verzonden. Het array *map[]* bevat de vertaling van *values[i]* naar *buffer[i]*. Indien nodig (als *values[i]* gelijk is aan *minima[i]* of *maxima[i]*), wordt *orient[i]* geïnverteerd. De hoofdloop verandert *minima[i]* en *maxima[i]* op basis van de waarden van de A/D-omzetter. De 'ruitenwisser' moet tussen deze twee waarden heen en weer gaan. Maar soms is de huidige positie van de wisser (*values[i]*) lager dan de nieuwe *minima[i]* of hoger dan de nieuwe *maxima[i]*. Als dat gebeurt, detecteert de functie *checkRange()* dat en verandert de richting (*orient[i]*) van die wisser. Daardoor zal de wisser vanzelf terugkeren naar een positie tussen *minima[i]* en *maxima[i]*.

Mechanische constructie

Er waren eigenlijk twee opties: de buizen verticaal monteren, zoals in een buizenversterker, of ze horizontaal monteren. Ik heb voor dat laatste gekozen, omdat we de bovenkant van de buizen willen zien. Ik wilde een behuizing die geschikt was om in een audiorack te plaatsen. Daarom heb ik de metalen behuizing van een oude kabel-TV-decoder gebruikt. Ik heb deze omgedraaid en een nieuwe frontplaat gemaakt van een metaalplaat uit een oude PC, met nauwkeurig (met een trapboor) geboorde gaten. Boor nooit een gat in metaalplaat met een normale boor, want dan wordt het gat

niet rond maar min of meer driehoekig! Ik heb de gaten eerst geboord met een 3018 CNC-router, die ik normaal gebruik voor het maken van printen.

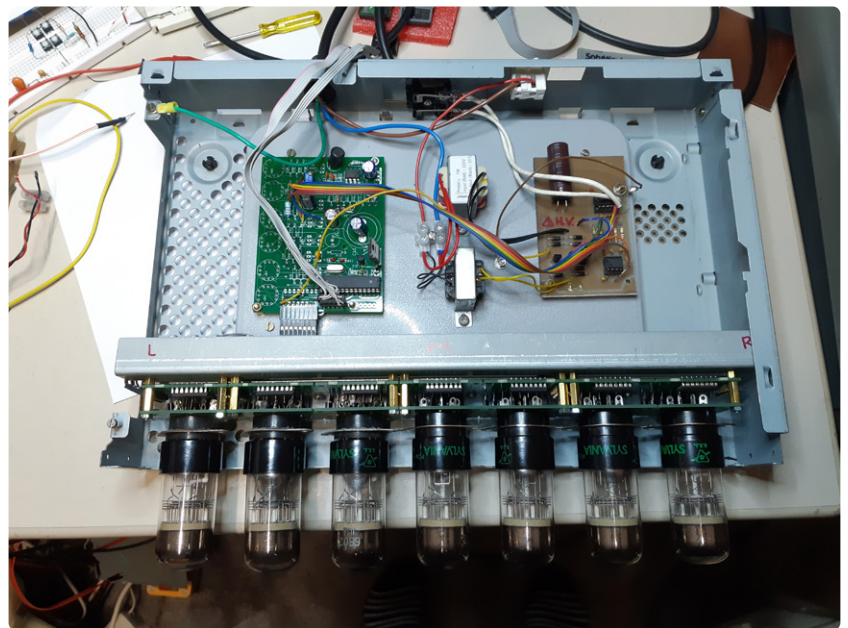
In **figuur 8** zie je een bovenaanzicht van de elektronica in de geopende behuizing. De transformatoren en de printen zijn gemonteerd op een stuk van een plastic snijplank, die ik in de behuizing heb vastgeschroefd. Alle dekatron-eenheden heb ik op een stevige metaalstrip geschroefd, die ook aan de behuizing vastzit. Voor de veiligheid heb ik de metalen behuizing verbonden met randaarde.

Mogelijke modificaties

Je kunt het ontwerp vereenvoudigen door een mono-uitvoering te bouwen. Dat spaart één MSGEQ7. Je kunt ook kiezen voor minder frequentiebanden en zo het aantal van dekatron-buizen reduceren. Daartoe kun je van zeven banden vijf banden maken door de twee hoogste en de twee laagste bereiken samen te nemen. Je kunt ook andere dekatrons dan de 6802 gebruiken; de 6802 heeft geen speciale eigenschappen die hem meer geschikt maken voor deze toepassing dan andere types.



Figuur 7. Het 'ruitenswisserdisplay' van nabij bekeken.



Figuur 8. De audio-spectrumanalyser van binnen.



ONDERDELENLIJST

Besturingseenheid

Weerstanden:

R1,R2,R4 = 10 k
R3,R5 = 200 k

Condensatoren:

C1 = 10 n
C2,C4,C5,C6,C7,C10,C12,C13,C18 = 100 n
C3,C11 = 33 p
C8,C9 = 12 p
C14 = 100 μ /25 V, radiaal
C15 = 100 μ /16 V, radiaal
C16 = 15 μ /500 V, radiaal

Spoelen:

L1 = 100 nH

Halfgeleiders:

D1...D8 = 1N4007
IC1 = ATMEGA328P
IC2,IC3 = MSGEQ7
IC4 = 78L05

Overig:

JP1,JP2 = 2-polige printkroonsteen
Q1 = kristal 20 MHz
SV1 = 6-polige SIL-header
SV2 = 8-polige SIL-header female
SV3 = 3-polige SIL-header of stereo klinkstekkerentree
Nettransformator 9 V/1 VA
Nettransformator 350 V/1 VA (zie tekst)

Dekatron-eenheid

JE HEBT AL HET ONDERSTAANDE 7 KEER NODIG!

Weerstanden:

R0..R7 = 10 k
R8 = 820 k

Condensatoren:

C20 = 100 n

Halfgeleiders:

T0...T6 = MPSA42
IC1 = 74HC595N

Verder:

SV1 = 8-polige SIL-header
SV2 = 8-polige SIL-header female
6802 Dekatron met voetje



Dit project bewijst dat een dekatron kan worden gebruikt voor een heel andere toepassing dan waar hij voor ontworpen is: een balk (of eigenlijk een boog) van lichtpunten die de signaalsterkte weergeven. Natuurlijk is dit principe ook te gebruiken voor andere toepassingen dan een audio-spectrumanalyser. Maar de retro-look van dit display is heel mooi, vooral in combinatie met een buizenversterker! 

210216-03

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen hebt naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via labs@vanden.eu of naar de redactie van Elektor via redactie@elektor.com. Zie voor meer informatie de projectpagina op Elektor Labs [4].



GERELATEERDE PRODUCTEN

- > **Six Digit Nixie Clock with IN-14 Tubes (SKU 20044)**
www.elektor.nl/20044
- > **Menno van der Veen, Modern High-End Valve Amplifiers (E-book) (SKU 18225)**
www.elektor.nl/18225
- > **Menno van der Veen, High-End Valve Amplifiers 2 (E-book) (SKU 18224)**
www.elektor.nl/18224

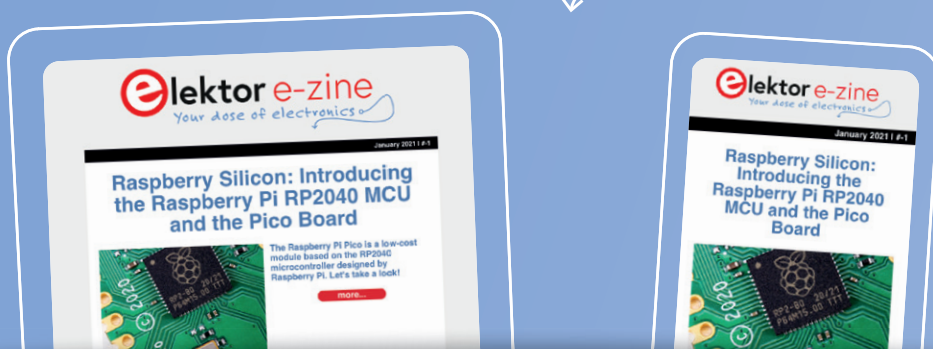
WEBLINKS

- [1] Dekatron: <https://en.wikipedia.org/wiki/Dekatron>
- [2] Harwell Witch-computer: https://en.wikipedia.org/wiki/Harwell_computer
- [3] GitHub-pagina met de downloads: <https://github.com/CharlesVanDen/AudioSpectrumAnalyserWithDekatrons>
- [4] Projectpagina op Elektor Labs: www.elektormagazine.com/labs/i-finally-found-a-useful-application-for-dekatrons
- [5] Sunil Malekar, "Simpele 7-band audio-spectrometer in twee versies", Elektor mei/juni 2019: www.elektormagazine.nl/magazine/elektor-98/42630

Advertentie

elektor e-zine

Your dose of electronics



Elke week dat u zich niet inschrijft voor ons Elektor E-zine mist u leuke elektronica-gerelateerde artikelen en projecten!

Dus, waarom nog langer wachten? Abonneer u vandaag nog op www.elektor.nl/ezone en ontvang bovendien een gratis Raspberry Pi projectboek!



elektor
design > share > sell