

analyseur de spectre audio avec dekatrons

une nouvelle façon d'utiliser les tubes rétro



Charles van den Ouweland (Pays-Bas)

Nous aimons trouver de nouvelles utilisations pour les composants rétro et surtout les vieux afficheurs. Ils sont trop beaux pour laisser dans un tiroir. Ce projet d'analyseur audio en est un exemple. Nous donnons une seconde vie aux dekatrons.

Les dekatrons sont des tubes de comptage à gaz qui datent des années 1950 [1]. Pendant une certaine période de l'histoire, ils ont été utilisés comme compteurs et cellules de mémoire dans les ordinateurs, notamment l'Harwell Witch [2]. Chaque tube pouvait contenir un chiffre de 1 à 10, donc il fallait un grand nombre de tubes pour construire un ordinateur. J'avais deux de ces tubes. Je pensais alors à un projet dans lequel ils pourraient être réutilisés. Ceux qui construisent des horloges nixie ajoutent parfois un dekatron comme compteur de secondes, ce qui est bien, mais je voulais trouver une nouvelle idée.

Dans cet article, je vous présente un analyseur de spectre audio stéréo à 7 bandes avec sortie dekatron. C'est un excellent outil pour tout audiophile possédant un amplificateur à tubes et souhaitant faire vivre le spectacle visuel ! L'idée est simple : au lieu de représenter le spectre audio à l'aide d'un graphique à

barres, d'un afficheur à LED - ou encore plus ennuyeux - d'un afficheur LCD, le volume du signal audio dans chaque bande de fréquence est indiqué par un certain nombre de points qui s'allument dans l'un des dekatrons. Chaque dekatron forme un cercle de points. La moitié gauche des points représente la partie gauche du signal stéréo tandis que la moitié droite représente le signal droit.

La conception de mon analyseur de spectre audio (**figure 1**) comporte une unité de contrôle pour le traitement du signal, et sept unités dekatron identiques pour l'affichage. Et, bien sûr, on a besoin d'une alimentation.

Unité de contrôle

La **figure 2** montre le schéma de l'unité de commande et de la double alimentation de l'analyseur de spectre. Pour commencer, j'ai recyclé et modifié une carte pour construire mon propre prototype pour ce circuit. A

présent, le schéma (fichier Eagle msgeq7c-trler_Elektor.sch) et la disposition du circuit imprimé sont disponibles pour téléchargement sur la page GitHub de ce projet [3].

Le microcontrôleur (IC1) est le fameux ATmega328P, comme sur l'Arduino UNO. Cependant, ici il n'est pas utilisé comme un Arduino. Le petit programme (300 lignes) est écrit en langage C avec Atmel Studio.

Pour déterminer le spectre audio, j'ai utilisé une puce MSGEQ7 dédiée de la société américaine Mixed Signals, qui analyse le signal audio et le divise en sept bandes de fréquences. Un microcontrôleur peut alors générer les sept niveaux en envoyant des commandes numériques à la puce. Cette dernière fournit une tension sur sa sortie, qui est converti par le CA/N du microcontrôleur. Ce projet comporte deux circuits intégrés analyseurs (IC2 et IC3), un pour chaque canal audio.

Unité dekatron

Le microcontrôleur communique avec les unités dekatron par une liaison série à trois fils, également appelée SPI, à 5 Mbits/s. Chaque tube dekatron est monté sur une unité compo-

Danger de haute tension !

Les dekatrons fonctionnent à des tensions d'alimentation pouvant atteindre 450 V. Si vous n'êtes pas familiarisé avec les hautes tensions, n'essayez même pas de réaliser ce projet ou de travailler avec ces tubes. Même après avoir coupé le courant alternatif, une haute tension peut encore être présente sur le condensateur haute tension. Avant de travailler avec le circuit, assurez-vous que le condensateur est déchargé.

Ni le concepteur ni Elektor n'assument la responsabilité de tout dommage, quel qu'il soit, résultant de la construction, des essais et de l'utilisation de (certaines parties de) ce projet !

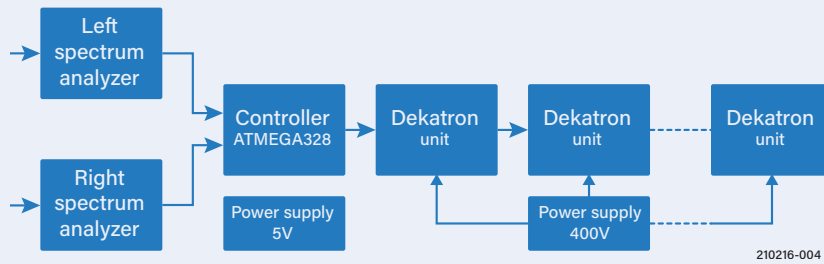


Figure 1. Schéma de l'analyseur audio.

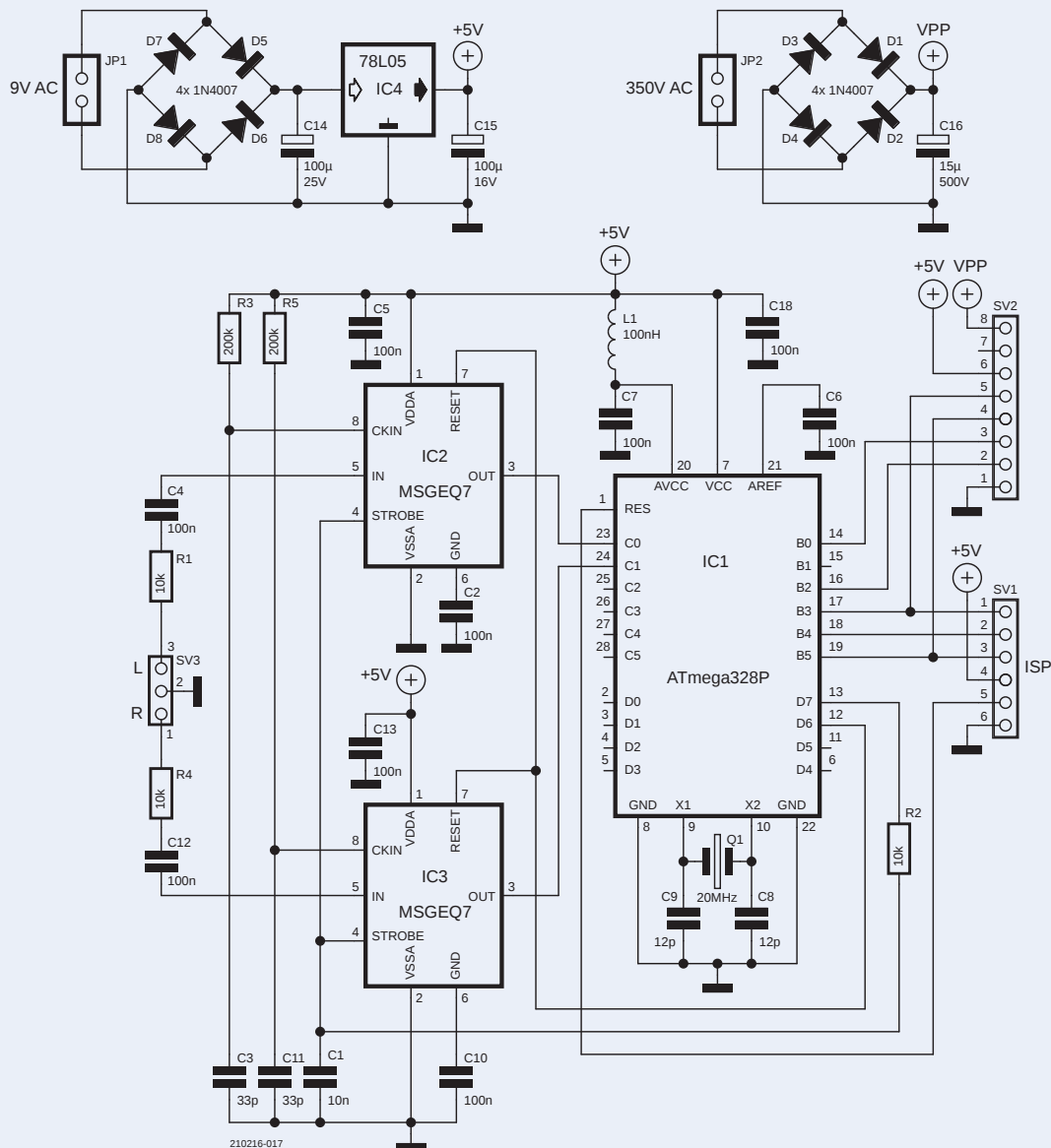
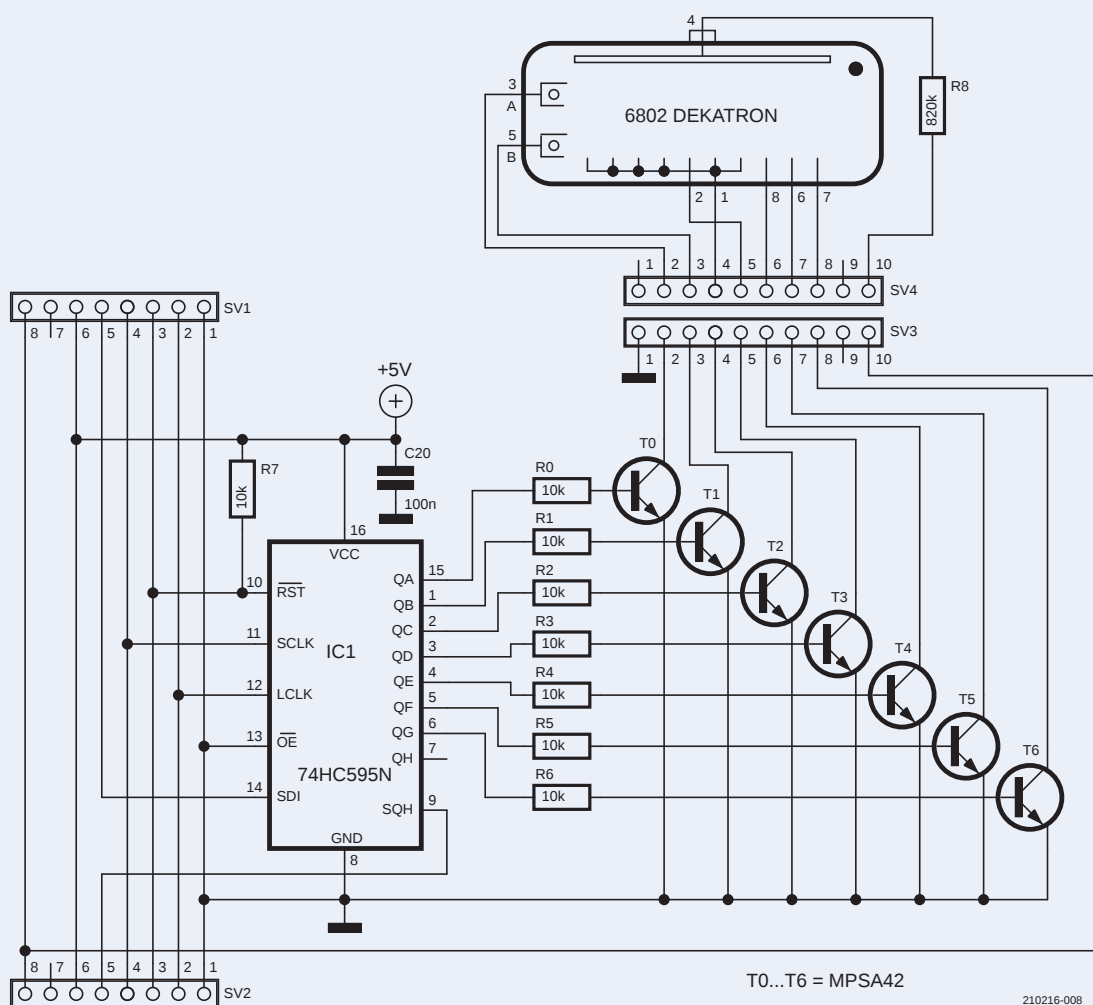


Figure 2. Schéma de l'unité principale.



Le schéma d'un afficheur dekatron. A réaliser en sept exemplaires.

sée d'un 74HC595 (registre à décalage SIPO) et d'un ensemble de transistors MPSA42, utilisés comme pilotes haute tension.

J'ai conçu un circuit imprimé (4 cm × 4,5 cm) pour cette unité, ou en fait deux circuits imprimés superposés. La **figure 3** montre le schéma de principe. Le circuit imprimé supérieur comprend le support du dekatron et la résistance d'anode, tandis que le circuit imprimé inférieur contient tous les autres composants. Les trous pour le support du dekatron sont adaptés à différents types de supports (aussi bien céramiques que métal/plastique). Le circuit imprimé inférieur comporte un connecteur mâle SV1 à gauche et un connecteur femelle SV2 à droite, de sorte qu'ils peuvent facilement être connectés en série à une chaîne de longueur arbitraire. La **figure 4** montre les circuits imprimés et les modules dekatron assemblés et connectés pour les tests.

Les fabricants de circuits imprimés proposent

souvent un service de prototypage de dix circuits imprimés pour un prix raisonnable. Vous obtiendrez donc des cartes pour la construction de dix unités à un prix avantageux. Les fichiers gerber et de perçage dont vous avez besoin pour commander ces cartes sont contenus dans Unit6802_Elektor.ZIP [3].

Alimentation

Le circuit exige deux tensions d'alimentation : 5 V et 400 V. Les composants électroniques nécessaires pour fournir ces tensions sont également représentés sur la **figure 2**. L'alimentation de 5 V est simple : un transformateur de 9 V avec un pont de diodes, un régulateur de tension 78L05 et quelques condensateurs.

Une décharge gazeuse a lieu dans les dekatrons. Pour que cette décharge s'allume, il est nécessaire de les mettre sous une tension VPP relativement élevée d'environ 400 V. Ma première solution était une alimentation à

découpage à base d'un MC34063 ou même d'un simple circuit 555. Mais j'ai découvert qu'une telle alimentation provoque beaucoup d'interférences avec les signaux analogiques (surtout sur la sortie du MSGEQ7), et il s'est avéré presque impossible d'obtenir de bons résultats. J'ai ensuite trouvé un fournisseur qui fabrique des transformateurs selon les spécifications des clients en Chine, Yollen Electrical Store. J'ai commandé quelques petits transformateurs 350 V, 1 VA, 50 Hz chez eux à 3 € par unité, plus 6,50 € de frais de livraison (**figure 5**). Ajoutez quatre diodes 1N4007 (D1 à D4) et un condensateur 10 µF 500 V (C16) - c'est tout ce dont vous avez besoin. Même si l'appareil fait appel à des tubes, la consommation d'énergie est très faible. Les dekatrons sont des composants à cathode froide, c'est-à-dire que la cathode n'est pas chauffée ! La seule puissance consommée par chaque tube est le courant anodique de 0,3 mA, qui ajoute une puissance d'environ

1 W au niveau de la partie haute tension. Les deux transformateurs ont une puissance nominale de 1 VA et la consommation totale est inférieure à 2 W.

Logiciel

Le logiciel pour l'ATmega328 est écrit en langage C avec Atmel Studio, le code source et le fichier HEX(adecima) pour programmer le microcontrôleur peuvent être téléchargés sur la page GitHub [3]. Les traitements sont effectués soit dans la boucle principale, soit dans des routines de service d'interruption. L'interfaçage avec les puces de l'analyseur (y compris le contrôle du CAN) se fait depuis la boucle principale, tandis que l'interfaçage avec les unités dekatron via SPI se fait avec les routines d'interruption.

Les signaux audio gauche et droit sont acheminés vers deux puces analyseurs. Chacune d'entre elles contient sept filtres passe-bande (60 Hz, 150 Hz, etc., voir **figure 6**), et le niveau de sortie de chaque bande est échantillonné. L'AVR reçoit les niveaux audio sous forme de signaux analogiques, les numérise et utilise les dekatrons comme dispositifs d'affichage.

Acquisition des données

Pour réduire le bruit au niveau du logiciel, chaque valeur est mesurée 15 fois et les meilleures valeurs parmi ces mesures sont retenues. Comme il y a sept bandes de fréquences et deux canaux (L et R), 210 conversions analogiques-numériques sont effectuées à chaque cycle, avant que l'affichage ne soit mis à jour !

Voici comment cela fonctionne en détail. On réinitialise les deux puces MSGEQ7 de l'analyseur. Ensuite, on envoie une impulsion d'échantillonnage aux deux puces. À ce stade, le signal de la première bande de fréquence est émis sur les broches 3 des puces. Une conversion analogique-numérique (C/AN) est effectuée sur le canal gauche, puis sur le canal droit. Les deux conversions sont répétées deux fois, ce qui donne trois valeurs pour le canal gauche et trois valeurs pour le canal droit. Les valeurs les plus basses et les plus hautes sont supprimées, et seules les valeurs intermédiaires sont gardées. Une deuxième impulsion d'échantillonnage est envoyée aux deux circuits intégrés, les faisant passer à la deuxième bande de fréquence et le processus d'échantillonnage est répété. Il en va de



Figure 4. Circuits imprimés et unités assemblées.



Figure 5. Transformateur de l'alimentation CA à 350 VAC.

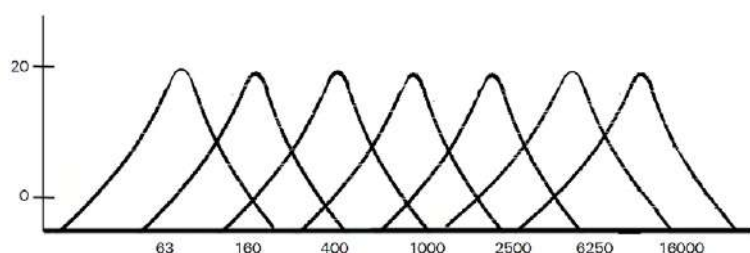
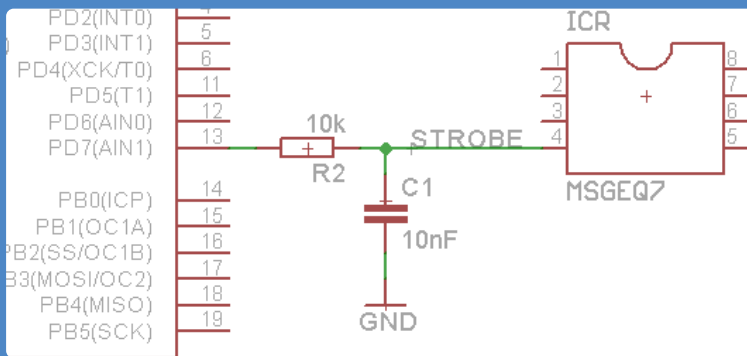


Figure 6. Bandes de fréquences du MSGEQ7. (Source : Mixed Signal Integration).

MSGEQ7... authentique ou faux ?

Attention aux fausses puces ! J'ai acheté des MSGEQ7s auprès d'un vendeur sur Aliexpress et auprès d'un vendeur sur Ebay et les deux sont arrivés avec de fausses puces. Celles que j'ai commandées chez Sparkfun pour 4,95 \$ étaient authentiques. Il est assez facile de déterminer si vous avez une puce authentique ou une fausse. Il suffit de connecter 5 V entre la broche 1 (VDD = +5 V) et la broche 2 (VSS = 0 V) et de mesurer le courant. Un MSGEQ7 authentique consomme moins de 1 mA. Les faux que j'ai eus consommaient plus de 10 mA.

Bruit de sortie du MSGEQ7



Cette puce a déjà été traitée dans le numéro de mai 2019 d'Elektor [5], mais ce dont on ne vous a pas parlé, c'est de la façon de gérer les problèmes de bruit avec cette puce.

Mes premières tentatives d'utilisation de la puce n'ont pas été très fructueuses à cause de problèmes de bruit. Une valeur sur dix environ provenant de la puce était erronée. Sur les forums Internet, on voit d'autres personnes se battre contre le même problème. J'ai pris une multitude de mesures pour le surmonter.

J'ai d'abord essayé de le résoudre par logiciel, en échantillonnant la même valeur plusieurs fois et en obtenant plusieurs mesures de la puce. J'ai supprimé les valeurs les plus élevées et les plus basses et j'ai calculé la moyenne du reste des échantillons. Cela a aidé un peu. J'ai également essayé diverses mesures au niveau matériel.

Tout d'abord, j'ai essayé de maintenir une distance entre le microcontrôleur et le MSGEQ7. Cela n'a pas du tout aidé. J'ai essayé des alimentations différentes pour les deux, également en vain. Puis je me suis débarrassé de l'alimentation à découpage, j'ai recouru à une alimentation très simple avec des transformateurs secteur, des diodes, des condensateurs et un 7805. Cela a aidé un peu. En désespoir de cause, j'ai placé un filtre RC sur le signal STROBE. À ma grande surprise, cela a résolu le problème ! Vous pouvez le trouver sur le schéma : C1 et R2 forment un simple filtre RC qui fait de STROBE un mauvais signal numérique, mais le MSGEQ7 fonctionne toujours et le bruit a (presque) disparu !

même pour les bandes de fréquence de la troisième à la septième. Ce cycle d'acquisition de données est répété cinq fois et seules les plus petites valeurs sont utilisées car j'ai découvert que les valeurs plus élevées ne sont généralement que du bruit.

Affichage des spectres

Le contrôle des dekatrons fonctionne comme suit. Pour le programme, chaque dekatron a 60 états. Pourquoi 60 ? Un dekatron n'est-il pas censé avoir 10 états, d'où le mot « deka » ? Si, initialement, le dekatron était censé avoir 10 états stables. Mais entre deux états stables, il y a deux états transitoires. Au total, le dekatron possède 30 cathodes, 10 pour chaque état stable, et entre chaque état stable, il y a deux cathodes pour la transition. Au départ, le dekatron était utilisé comme un tube de comptage. En appliquant une impulsion décalée sur la première puis la deuxième cathode de transition, il pouvait compter « un en plus ». Cela signifie également qu'avec un dekatron, il est facile de déplacer le point

d'une position à une autre adjacente. Mais ce n'est pas si simple (et la plupart du temps, ce n'est même pas possible) de déplacer le point directement d'une position à une autre, quelconque. Cela n'est possible que pour quelques-unes des positions stables. Pour certains dekatrons, les dix positions stables ont un « accès aléatoire », mais d'autres dekatrons en ont moins. Sur le 6802 que j'ai utilisé, seules les positions 0, 5, 8 et 9 ont une. J'utilise initialement la position 0 comme point de départ et par la suite, le point se déplace uniquement vers les positions adjacentes, soit d'un pas vers la gauche, soit d'un pas vers la droite.

Dans cette application, nous utilisons les 30 cathodes. En outre, il est possible d'exciter deux cathodes voisines en même temps, ce qui sert de transition entre une cathode et la suivante. De cette façon, nous obtenons 60 états.

Cette application dépend de la commutation rapide entre les états, si rapide qu'elle trompe l'œil. L'état actuel du dekatron *i* (un nombre compris entre 0 et 59) est stocké

dans `values[i]`. Comme un essuie-glace, il varie rapidement entre deux états limites (**figure 7**). Les états limites sont stockés dans `minima[i]` et `maxima[i]`. La direction actuelle du mouvement (vers *minima* ou vers *maxima*) est stockée dans le paramètre booléen `orient[i]`. L'état est modifié toutes les 150 µs. Lorsque 150 µs se sont écoulées, la routine de service d'interruption `TIMER0_COMPA_vect()` est appelée. À ce stade, les nouvelles valeurs à envoyer aux dekatrons sont déjà prêtes dans `buffer[]`. La première valeur sortant de `buffer[]` est envoyée sur le bus SPI à un taux de 5 Mbit/s. Lorsque cette transmission SPI est terminée, la routine de service d'interruption `SPI_STC_vect()` est appelée, ce qui envoie l'octet suivant sur le bus SPI. Lorsque les sept octets sont envoyés ainsi, `SPI_STC_vect()` calcule les nouvelles `values[]`. La valeur de `buffer[]` est calculée, et elle doit contenir la configuration binaire réelle à envoyer sur le bus SPI. Le tableau `map[]` contient la conversion de `values[i]` en `buffer[i]`. Si nécessaire (lorsque `values[i]` est égal `minima[i]` ou `maxima[i]`), `orient[i]` est inversé.

La boucle principale change `minima[i]` et `maxima[i]` en fonction des valeurs provenant du CAN. « L'essuie-glace » est censé balayer entre ces deux valeurs. Mais parfois sa position actuelle (`values[i]`) est inférieure à la nouvelle valeur de `minima[i]` ou supérieure à celle de `maxima[i]`. Lorsque cela se produit, la fonction `checkRange()` le détecte et change la direction (`orient[i]`) de cet essuie-glace qui reviendra automatiquement à une position entre `minima[i]` et `maxima[i]`.

Construction mécanique

J'avais deux options : tout comme un amplificateur à tubes, placer les tubes verticalement, ou les monter horizontalement. J'ai choisi cette dernière option, car les tubes sont des modèles à vue de dessus. Je voulais un boîtier pour le projet qui puisse être placé parmi une rangée d'équipements audio. J'ai donc pris le boîtier métallique d'un vieux décodeur de câble de télévision. Je l'ai utilisé à l'envers et j'ai ajouté un nouveau panneau avant, qui consiste en une plaque métallique provenant d'un vieux PC et dont les trous ont été percés avec précision avec une mèche dite « étagée ». Ne percez jamais un trou dans une tôle avec une mèche normale, car vous

n'obtiendrez pas un trou rond, mais plus ou moins triangulaire ! Les trous ont d'abord été soigneusement percés avec une fraiseuse CNC 3018, que j'utilise normalement pour le fraisage de circuits imprimés.

La **figure 8** montre la vue de dessus de l'analyseur de spectre audio à l'intérieur du boîtier ouvert. J'ai monté les transformateurs et les circuits imprimés sur un morceau de planche à découper en plastique, que j'ai ensuite fixé au boîtier. Les unités dekatron étaient toutes vissées sur une barre métallique solide qui était également fixée au boîtier. Par sécurité, j'ai relié le boîtier métallique à la terre.

Possibilités de modifications

Il est possible de simplifier la conception en réalisant un modèle mono, ce qui permet d'épargner un MSGEQ7 ; ou de réduire le nombre de bandes de fréquences et donc le nombre de tubes dekatron nécessaires. Il est possible, par exemple, de regrouper les sept bandes en cinq bandes en combinant les deux gammes les plus élevées et les deux gammes les plus basses. Il est également possible d'utiliser d'autres dekatrons que le 6802, il n'y a pas de caractéristiques particulières du 6802 qui le rendent plus adapté à cette application que tout autre tube.

Ce projet montre qu'un dekatron peut être utilisé comme un type d'affichage totalement différent de celui pour lequel il a été



Figure 7. Zoom sur trois afficheurs dekatron.

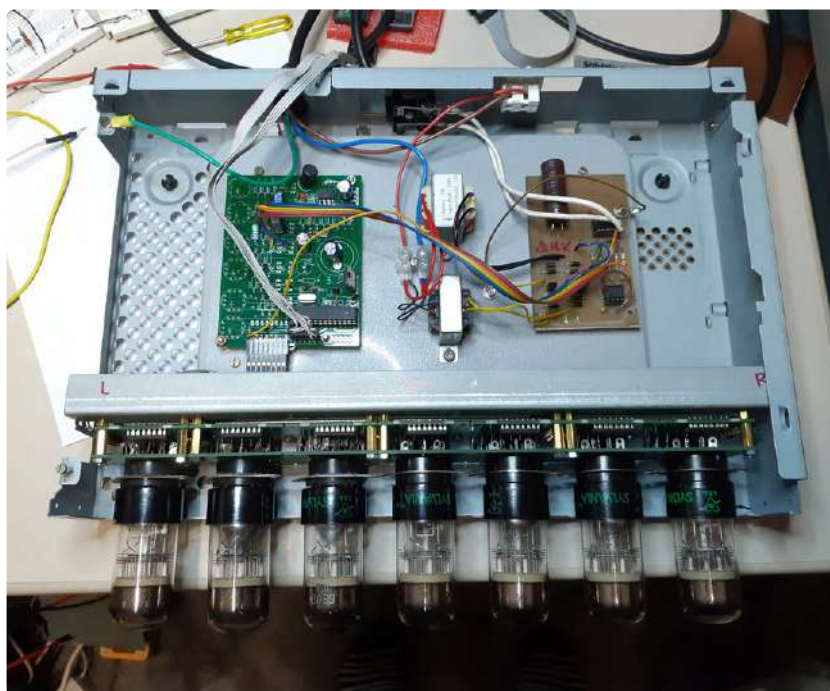


Figure 8. Vue de l'intérieur du boîtier de l'analyseur de spectre audio.



LISTE DES COMPOSANTS

Unité principale

Résistances

R1,R2,R4 = 10 kΩ
R3,R5 = 200 kΩ

Inductances

L1 = 100 nH

Condensateurs

C1 = 10 nF
C2,C4,C5,C6,C7,C10,C12,C13,C18 = 100 nF
C3,C11 = 33 pF
C8,C9 = 12 pF
C14 = 100 μF, 25 V radial
C15 = 100 μF, 16 V radial
C16 = 15 μF, 500 V radial

Semiconducteurs

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

Autres

JP1,JP2 = bornier à 2 voies
Q1 = cristal 20MHz
SV1 = connecteur SIL à 6 voies
SV2 = prise SIL 8 voies
SV3 = embase SIL à 3 voies ou connecteur jack stéréo
Transformateur secteur 9 V 1 VA
Transformateur secteur 350 V 1 VA (voir texte)

Unité Dekatron

Il vous en faut 7 pièces de tout ce qui est indiqué ci-dessous !

Résistances

R0..R7 = 10 k
R8 = 820 k

Condensateurs

C20 = 100 nF

Semiconducteurs

T0...T6 = MPSA42
IC1 = 74HC595N

Autres

SV1 = Connecteur SIL 8 voies
SV2 = Prise SIL 8 voies
6802 Dekatron avec support 8 voies



conçu : une barre (ou plutôt un arc) de points indiquant la force du signal. Bien sûr, ce principe peut être utilisé pour d'autres applications que les analyseurs de spectre audio. Mais le style rétro de cet afficheur de spectre est génial, surtout lorsqu'il est associé à un amplificateur à tubes ! ◀

210216-04

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (labs@

vanden.eu) ou contactez Elektor (redaction@elektor.fr). Des informations supplémentaires sont disponibles sur la page de ce projet sur Elektor Labs [4].



PRODUITS

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

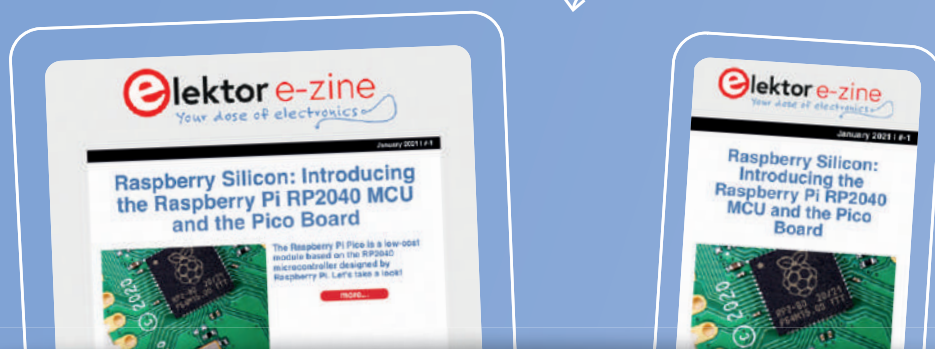
LIENS

- [1] Dekatron: <https://fr.wikipedia.org/wiki/Dekatron>
- [2] Ordinateur d'Harwell: https://fr.wikipedia.org/wiki/Ordinateur_d%27Harwell
- [3] Github page with downloads: <https://github.com/CharlesVanDen/AudioSpectrumAnalyserWithDekatrons>
- [4] Ce projet sur Elektor Labs: www.elektormagazine.fr/labs/i-finally-found-a-useful-application-for-dekatrons
- [5] Sunil Malekar, « spectromètres audio simples à 7 bandes », Elektor 5/2019: www.elektormagazine.fr/magazine/elektor-97/42590

Publicité

e lektor e-zine

Your dose of electronics



Chaque semaine où vous n'êtes pas abonné à l'e-zine d'Elektor est une semaine de grands articles et de projets électroniques qui vous manquent !

Alors, pourquoi attendre plus longtemps ? Abonnez-vous dès aujourd'hui à www.elektor.fr/ezine et recevez également le livre gratuit du projet Raspberry Pi !



e lektor
design > share > sell