



Interface
Compact-Flash



-Allo !
Maman ?
-Ici bébé !

Torche à LED

Fusible
électronique

Étoile de Noël

Dé à
microcontrôleur

MINI- CARTE-SON USB

À entrée et
sortie S/PDIF

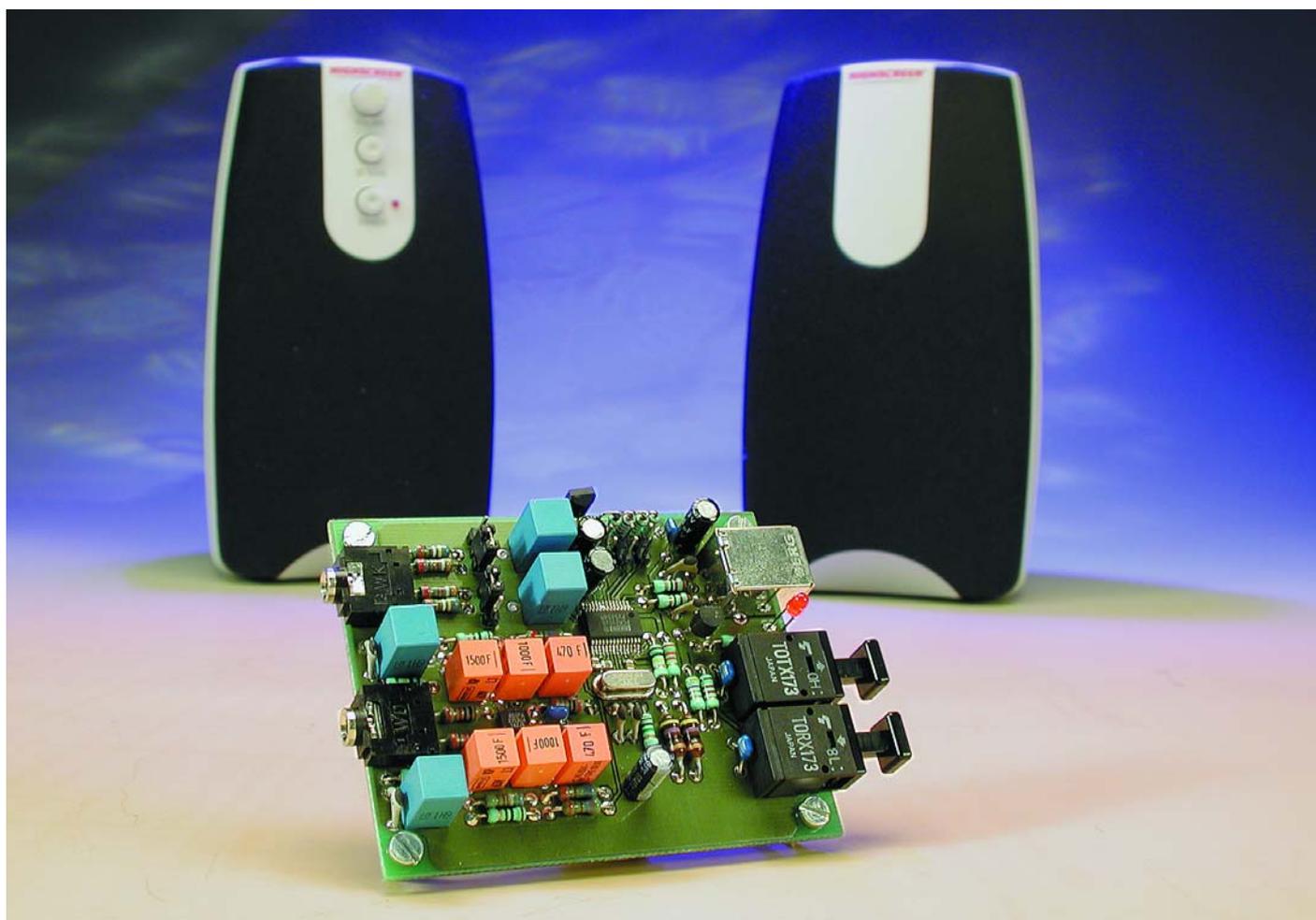


Mini-carte-son USB

À entrée et sortie S/PDIF

Projet : Ton Giesberts

L'« Audio-DAC USB » décrit dans le numéro de décembre 2000, a fait un véritable tabac. Il devient une mini-carte-son externe extrêmement populaire. La question la plus posée à son sujet était de savoir s'il n'y aurait pas moyen d'imaginer une carte capable, par le biais du port USB, d'effectuer des enregistrements. Qui peut le plus peut le moins. La mini-carte-son décrite dans le présent article répond à ce souhait, et ce sans que la complexité de l'électronique n'ait grandement augmenté.



Le montage décrit dans les prochains paragraphes repose sur un circuit intégré de Texas Instruments/Burr-Brown, un PCM2902, composant que l'on a doté de la dénomination sibylline de « Stereo Audio-Codec ». Il s'agit à l'évidence d'un membre de la famille du PCM2702 que nous avons utilisé dans l'Audio DAC USB, à la différence (importante) près que ce nouveau circuit dispose, outre d'une sortie analogique stéréo et bien entendu d'une interface USB, également d'une entrée analogique stéréo, d'une entrée S/PDIF et d'une sortie S/PDIF, ce qui permet même d'effectuer des enregistrements numériques depuis ou vers le PC. Il nous faut cependant placer ici un petit « Mais ». Le PCM2902 traite le signal d'audio numérique en respect des normes SCMS (*Serial Copy Management System*) et basculera partant, s'il s'agit d'une copie, vers l'entrée analogique. En cas d'application de données tant à l'entrée S/PDIF qu'à l'entrée analogique, le composant choisit automatiquement l'entrée numérique (si tant est que l'on y présente des données originales, vu le SCMS).

PCM2902 FUNCTIONAL BLOCK DIAGRAM

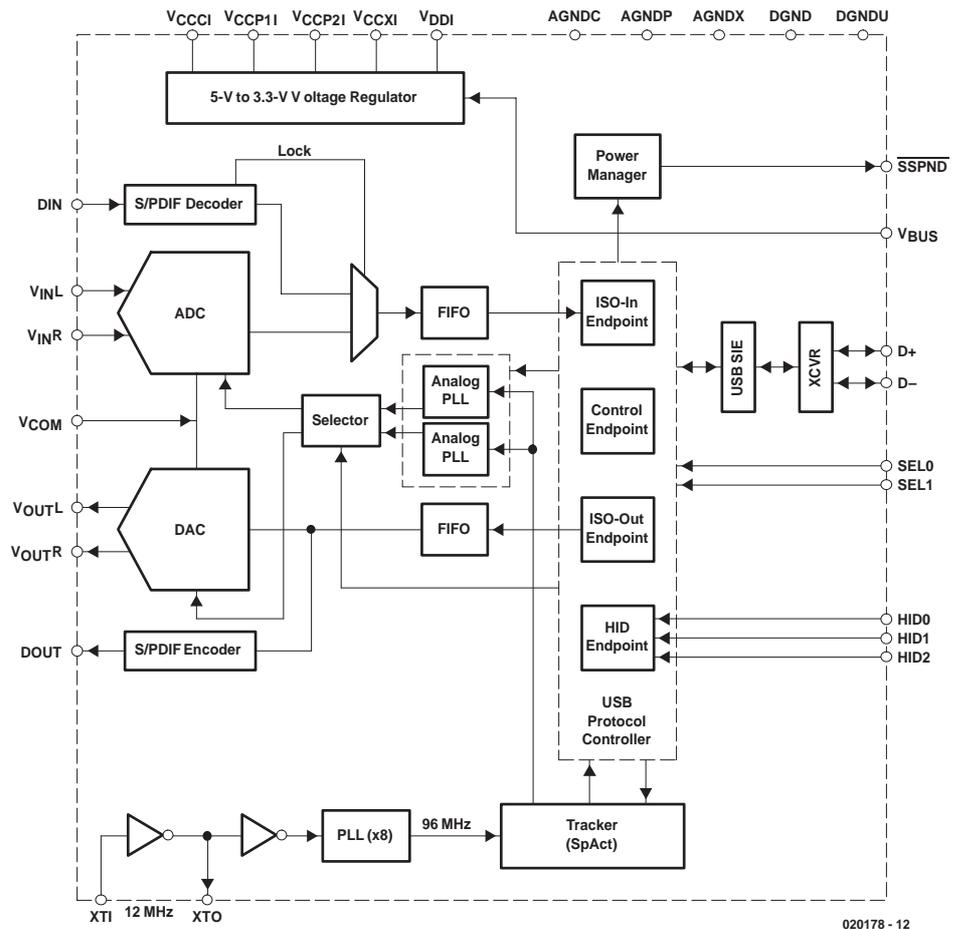


Figure 1. Synoptique de la structure interne du « Codec audio stéréo » PCM2902.

Si vous voulez en savoir plus quant aux tenants et aux aboutissants de ce circuit intégré nous vous recommandons de télécharger sa fiche de caractéristiques depuis le site Internet de Texas Instruments à l'adresse : www.ti.com.

En vue de satisfaire votre première curiosité, nous vous proposons, en **figure 1**, le synoptique de la structure interne du PCM2902. Tout comme c'était le cas avec le PCM2702, la qualité du traitement de signal est très acceptable. Les fanas de la Hi-FI n'ont rien à craindre. Il suffit, pour vous en convaincre, de jeter un coup d'oeil sur les spécifications de notre montage données dans le petit encadré.

La simplicité avant tout

Comme le prouve un coup d'oeil sur le schéma de la **figure 2**, l'électronique de notre mini-carte-son ne comporte que fort peu de composants. Si l'on fait abstraction du PCM2902, IC1, le coeur de cette réa-

Résultats de mesure

(VBUS = 4,84 V, VCCCI = 3,5 V)

Consommation de courant

≈ 90 mA

CNA

Tension de sortie nominale (0dB)

1,1 V_{eff}

Domaine de fréquences (-3 dB)

22,7 kHz (f_s = 48 kHz)

Amplitude 20 kHz

-0,8 dB (f_s = 48 kHz)

Bande passante du filtre analogique

28 kHz

Impédance de sortie

100 Ω

Rapport signal/bruit

> 95 dBA

THD+N (1 kHz)

0,005% (B = 22 kHz)

0,046% (B = 80 kHz)

Diaphonie

> 99 dB (1 kHz)

> 76 dB (20 kHz)

CAN

Plage d'excursion de la tension d'entrée

0,75 V_{eff} (LO)

2,1 V_{eff} (HI)

Impédance d'entrée

7,7 kΩ (LO)

10 kΩ (HI)

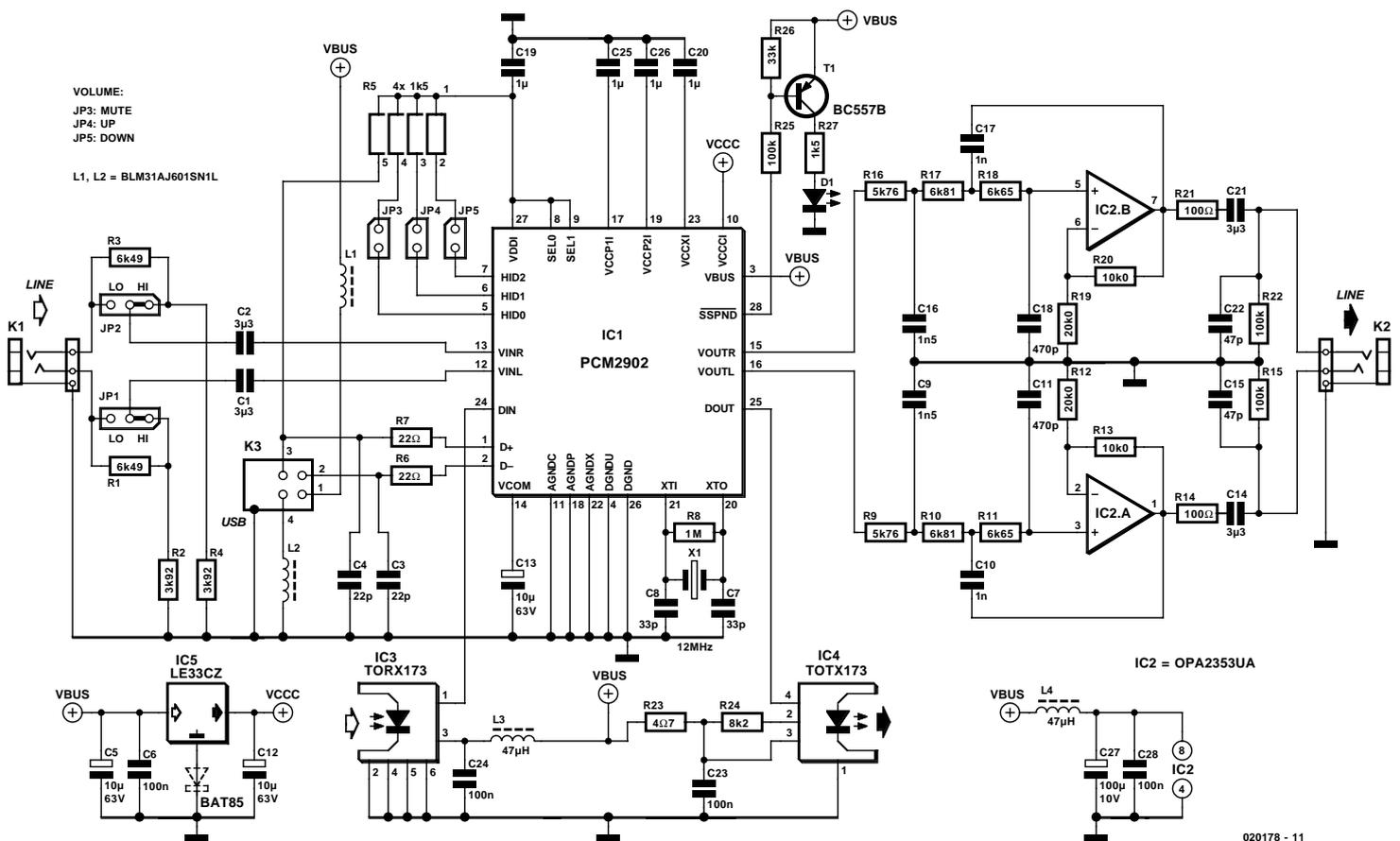
THD+N (1 kHz, -0,5 dBFS)

0,01% (B = 22 kHz)

Diaphonie

> 73 dB (1 kHz)

> 47 dB (20 kHz)



020178 - 11

Figure 2. La simplicité du schéma tient à l'intégration de l'ensemble de l'électronique dans IC1.

lisation, notre montage ne compte, à tout casser, qu'un unique transistor, un régulateur de tension intégré, un double amplificateur opérationnel et un module optique d'entrée et son homologue de sortie. Il nous faut en outre une minuscule poignée de composants passifs.

L'alimentation du montage se fait par le bus, ce qui signifie que le montage dérive sa tension de 5 V de la ligne correspondante du port USB. Ce n'est pas sans raison que nous avons placé les selfs L1 et L2 le plus près possible de l'embase USB, ceci de manière à déparasiter la tension d'alimentation à même la source. Ceci explique que nous avons placé ces selfs de type CMS sur les broches-même de l'embase K3, côté soudures (ex-côté « pistes ») - nous reviendrons sur ce point dans un prochain paragraphe. Notons au passage que le PCM2902 est doté de son propre régulateur de tension en interne.

L'entrée

L'entrée analogique possède une sensibilité prédéterminée, ce qui explique que la fenêtre des caractéristiques de l'enregistreur (*recorder specifications*) apparaissant après l'installation ne comporte pas le moindre réglage

de sensibilité. Ceci explique la présence, au niveau de l'entrée analogique, d'un cavalier par canal permettant, matériellement, de choisir entre 2 plages de niveaux d'entrée (HI/LO), le niveau HI étant fixé à de l'ordre de 2 V par le biais d'une paire de diviseurs de tension, R1/R2 et R3/R4. La plage des niveaux d'entrée du CAN (**C**onvertisseur **A**nalogique/**N**umérique) intégré dans le PCM2902 est définie par la taille de VCCCI (et égale à $0,6 \times VCCCI$). Vous avez bien entendu la liberté de remplacer le diviseur de tension par un potentiomètre stéréo. Les condensateurs C2 et C3 découplent la tension continue appliquée aux entrées de IC1. La tension en question vaut de l'ordre de $VCCCI/2$ (= VCOM) soit quelque 1,65 V.

Le fabricant recommande, en vue d'augmenter la qualité de la conversion A/N, de faire fournir à un régulateur externe la tension VCCCI. D'où la présence de IC5. Nous avons utilisé à ce niveau un régulateur de tension 3,3 V à faibles pertes de

l'écurie STMicroelectronics (un LE33CZ en boîtier TO-92), ce qui se traduit par des produits d'harmoniques se trouvant 10 dB plus bas qu'ils ne l'auraient été si nous avions utilisé la tension d'alimentation interne. Si vous tenez à être assuré de réduire au strict minimum la distorsion introduite par le CAN, vous pourrez, comme l'indiquent les pointillés sur le schéma, prendre une diode BAT85 en série avec la broche centrale du LE33CZ (anode de la diode vers le LE33CZ !). Cette adjonction a pour effet de relever la tension de référence à 3,5 V. Notons qu'il existe aussi, dans la série de régulateurs LEXXCZ, un composant pouvant fournir 3,5 V, le LE35CZ, mais ce composant n'est malheureusement pas disponible en standard. Le fait que le CAN soit doté, en interne, d'un filtre anti-repliement (*anti-aliasing*) a permis de simplifier énormément l'électronique de l'entrée analogique. Nous avons cependant opté, au niveau de C1 et C2, pour des condensateurs MKT plutôt

que pour des électrochimiques en vue d'augmenter encore la qualité, pour autant que cela soit possible.

La sortie

Le CNA (Convertisseur Numérique/Analogique) du PCM2902 ressemble beaucoup, tant un niveau de son concept que de ses spécifications, à ceux du « bon vieux » PCM2702; il est doté en interne d'un filtre passe-bas ayant un point de coupure situé à 250 kHz. Caractéristique typique d'un CNA Delta-Sigma tel que celui utilisé ici, une augmentation du bruit importante au-delà de 20 kHz (due à la mise en forme du bruit = *noise-shaping*). D'où la présence, en vue d'éliminer tous les produits de distorsion, d'un filtre passe-bas du 3^{ème} ordre (Butterworth, $f_c = 28$ kHz). Puisque

l'amplificateur de ce filtre est alimenté par le 5 V (VBUS), il est possible d'amplifier quelque peu le signal disponible à la sortie du CNA (facteur d'amplification de 1,5x). Ceci se traduit par un niveau de sortie maximum de plus de 1 V_{eff} . Nous avons utilisé, au niveau de cet amplificateur opérationnel (IC2), un OPA2353UA (en boîtier CMS SO-8), le même d'ailleurs que celui utilisé dans notre (fameux) « Audio-DAC USB ».

Cet amplificateur opérationnel double se caractérise par une vitesse élevée, un niveau de bruit et une distorsion faibles, mais aussi (surtout !) par sa capacité rail-à-rail tant en entrée qu'en sortie ! Les sorties des filtres sont dotées des condensateurs de découplage standard (en MKT bien évidemment) et de résis-

tances-tampon destinées à bloquer les charges capacitives. R15 et R22 font en sorte que les condensateurs de sortie soient toujours chargés (c'est d'ailleurs également la fonction des diviseurs de tension d'entrée au niveau du CAN). C15 et C22 ont pour fonction d'éliminer d'éventuels parasites HF. (Il ne faut oublier en effet que notre montage est relié galvaniquement au PC). Pour couronner le tout, l'amplificateur opérationnel est doté de son propre découplage d'alimentation sous la forme de la self L4 associée aux condensateurs C27 et C28.

Quelques autres détails

Les entrée et sortie S/PDIF n'appellent guère de commentaires. Comme le montre le schéma, elles prennent la forme des modules TOSLINK classiques : un TORX173 pour l'entrée et un TOTX173 pour la sortie. Les dits modules, IC3 et IC4, sont connectés directement aux entrées correspondantes de IC1 et

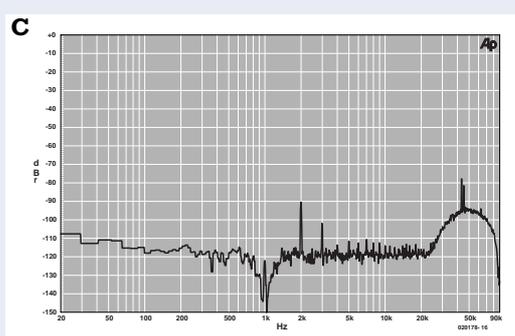
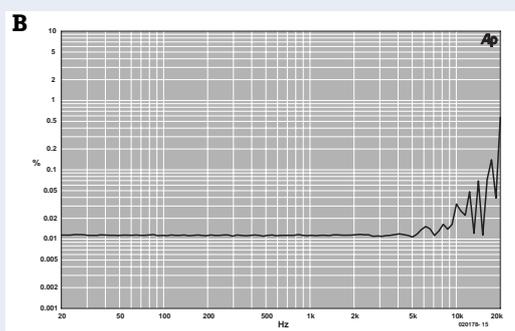
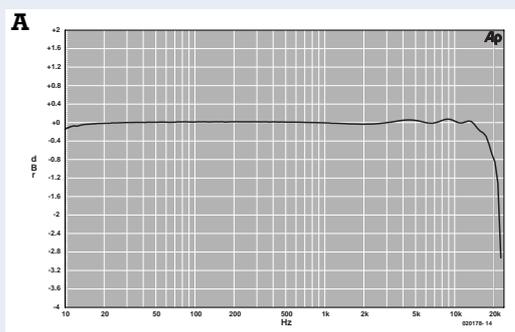
Courbes de mesure

Confucius ne disait-il pas qu'une image valait plus que mille mots ? Nous vous proposons partant une série de courbes plutôt qu'une liste sèche de nombres.

La première courbe, **A**, dérive de la pratique et montre la caractéristique d'amplitude dans le cas de la mesure en série du CAN et du CNA. Nous avons utilisé pour cela le ASIO Multimedia Driver de Cubase VST/32. Les pilotes USB audio standard de WIN98SE ne permettent pas de full duplex de sorte qu'il est impossible d'écouter ce que l'on vient d'enregistrer ! Il est cependant possible de jouer un CD audio ou un fichier wave à l'aide d'une autre application ! à 20 kHz, l'amplitude a subi une atténuation de 0,8 dB seulement. L'étalement extrême de l'échelle permet de bien voir l'ondulation résiduelle du filtre numérique du CNA, sachant que celle-ci n'est en rien due au filtre analogique !

La courbe **B** montre la somme distorsion + bruit en relation avec la fréquence. À nouveau la mesure s'est effectuée en prenant les CAN et CNA en série. La distorsion est due principalement au CAN, des analyses FFT nous ayant en outre appris que l'on y trouvait, outre la distorsion harmonique, également d'autres produits de mélange, dus sans doute à la PLL interne et aux filtres numériques.

À partir de 5 kHz, ce sont plus particulièrement les produits de repliement (aliasing) qui sont la raison de l'augmentation de la distorsion, mais ils n'ont rien d'alarmant dans le cas d'une réalisation aussi simple que celle-ci. La courbe **C**, pour finir, rend le spectre de fréquences à 997 Hz à pleine modulation du CAN (nous avons utilisé pour cela un CD de test). Toutes les harmoniques se situent en-deçà de -90 dB et sont la source majeure de la distorsion de 0,005% mesurée. Le bruit sensible au-delà de 20 kHz est typique dans le cas des convertisseurs delta-sigma à mise en forme de bruit (noise shaping); il se situe pratiquement en dehors de la plage de mesure si l'on a opté pour une bande passante de 22 kHz. Le filtre de sortie analogique limite pour une grande part le bruit et les produits de repliement au delà de 35 kHz. Si l'on fait passer à 80 kHz la bande passante dans laquelle se fait la mesure de distorsion harmonique totale + bruit (THD+N), ce bruit et les produits d'échantillonnage ont pour conséquence évidente une augmentation légère de ce facteur de distorsion (jusqu'à 0,046 %).



Liste des composants

Résistances :

R1,R3 = 6kΩ49
 R2,R4 = 3kΩ92
 R5 = réseau de 4 résistances de 1kΩ5
 R6,R7 = 22 Ω
 R8 = 1 MΩ
 R9,R16 = 5kΩ76
 R10,R17 = 6kΩ81
 R11,R18 = 6kΩ65
 R12,R19 = 20kΩ0
 R13,R20 = 10kΩ0
 R14,R21 = 100 Ω
 R15,R22,R25 = 100 kΩ
 R23 = 4Ω7
 R24 = 8kΩ2
 R26 = 33 kΩ
 R27 = 1kΩ5

Condensateurs :

C1,C2,C14,C21 = 3μF/50 V MKT au pas de 5/7,5 mm
 C3,C4 = 22 pF
 C5,C12,C13 = 10 μF/63 V radial
 C6,C23,C24,C28 = 100 nF céramique au pas de 5 mm
 C7,C8 = 33 pF
 C9,C16 = 1nF5 1% *
 C10,C17 = 1 nF/1% *
 C11,C18 = 470 pF/1% *
 C15,C22 = 47 pF
 C19,C20,C25,C26 = 1 μF/25 V, CMS format 0805 (chez Farnell par exemple, n° 317-640)
 C27 = 100 μF/10 V radial

*) polypropylène/polystyrène (EMZ)

Selfs :

L1,L2 = BLM31A60IS, CMS format 1206 (chez Farnell par exemple, n° 581-094)
 L3,L4 = 47 μH

Semi-conducteurs :

D1 = LED rouge à haut rendement
 T1 = BC557B
 IC1 = PCM2902 (Texas Instruments/Burr-Brown)
 IC2 = OPA2353UA (Texas Instruments/Burr-Brown)
 IC3 = TORX173 (Toshiba)
 IC4 = TOTX173 (Toshiba)
 IC5 = LE33CZ (STMicroelectronics, chez Farnell par exemple, n° 302-4568)

Divers :

JP1,JP2 = embase autosécable mâle à 3 contacts + cavalier
 JP3 à JP5 = embase autosécable mâle à 2 contacts
 K1,K2 = jack stéréo 3,5 mm encartable
 K3 = embase USB-encartable de type B
 X1 = quartz 12 MHz à résonance parallèle (C_{load} = 30 pF)

reliés à la ligne d'alimentation 5 V au travers de découplages standard. Le PCM2902 possède un réglage de volume HID (*Human Interface Device*) et une fonction de silencieux (*mute*). Les cavaliers JP3, JP4 et JP5

permettent de paramétrer respectivement le silencieux, une augmentation (*Volume UP*) et une diminution de volume (*Volume DOWN*). Il faudra bien évidemment connecter à cet endroit les 3 boutons-poussoirs cor-

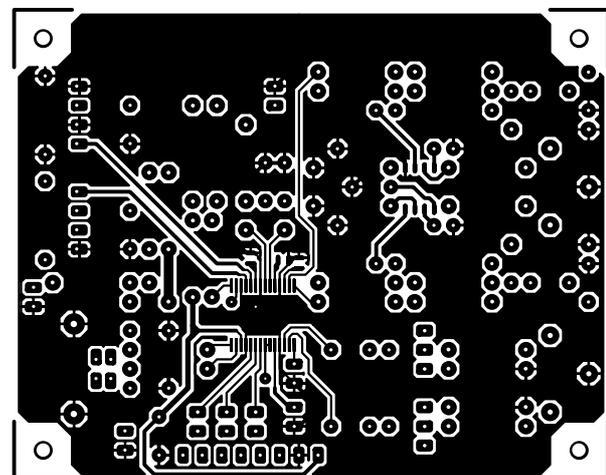
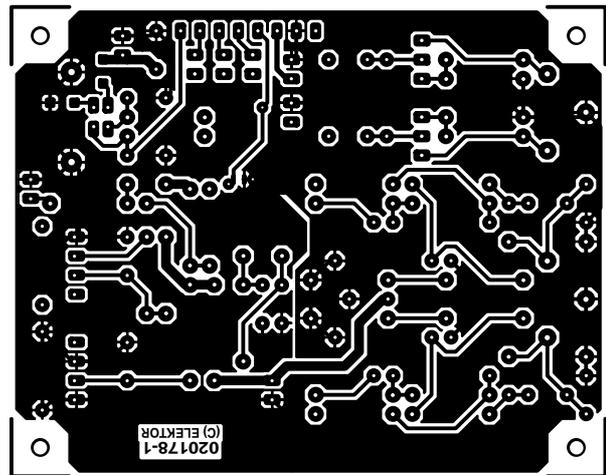
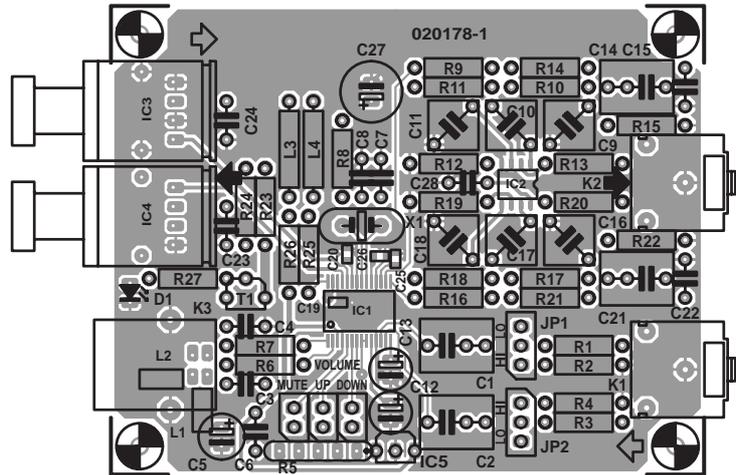


Figure 3. Le dessin des pistes de la platine double face à trous métallisés a été conçu en vue de garantir un fonctionnement optimal du montage.

respondants (attention à ne pas créer alors de court-circuit à la masse) de manière à pouvoir jouer sur ces paramètres sans avoir à passer par l'ordinateur.

Vu qu'après l'installation de ce composant (Windows 98 SE ou mieux au minimum) on a adjonction de 2 « périphériques » à la section HID (appareils à interface humaine) on a une rétroaction vers le système d'exploitation, ce qui lui permet d'actualiser à tout moment le niveau du volume et l'état du silencieux (ON ou OFF).

L'état de l'indicateur (*flag*) Suspend est disponible en broche 28, contact que nous mettons ici à contribution pour signaler, par le biais de l'allumage d'une LED, que le circuit se trouve momentanément hors-service. C'est à dessein que la LED est reliée à la masse de manière à éviter qu'en cas de court-circuit, ce circuit intégré (coûteux et difficile à dénichier) ne soit endommagé. Un petit convertisseur constitué par T1 transforme le niveau 0/3V3 de la sortie SSPND en un courant de 2 mA destiné à une LED à haut rendement. Cette approche offre en outre la possibilité d'indiquer à une autre application, depuis le point de connexion de la LED, par le biais d'un niveau logique de 5 V, que l'on n'a pas, momentanément, de signal.

Le circuit imprimé

La **figure 3** reproduit les recto et verso de la platine dessinée à l'intention de cette réalisation. Il va sans dire que nous avons veillé, pour garantir un fonctionnement impeccable, à ce que les connexions soient les plus courtes possible, ce qui se traduit par un circuit imprimé très compact. Comme d'habitude, les différents connecteurs ont été placés sur le pourtour de la platine.

90% des composants de cette réalisation sont de type standard, et ne devraient partant pas poser de problème de mise en place. L'utilisation de composants pour montage en surface (CMS) requiert elle une certaine expérience. On commencera de ce fait par « l'implantation » de ces derniers. Dans le cas de IC2, l'écartement entre les broches est tel que sa soudure est encore (relativement) facile, la soudure de IC1 requiert elle

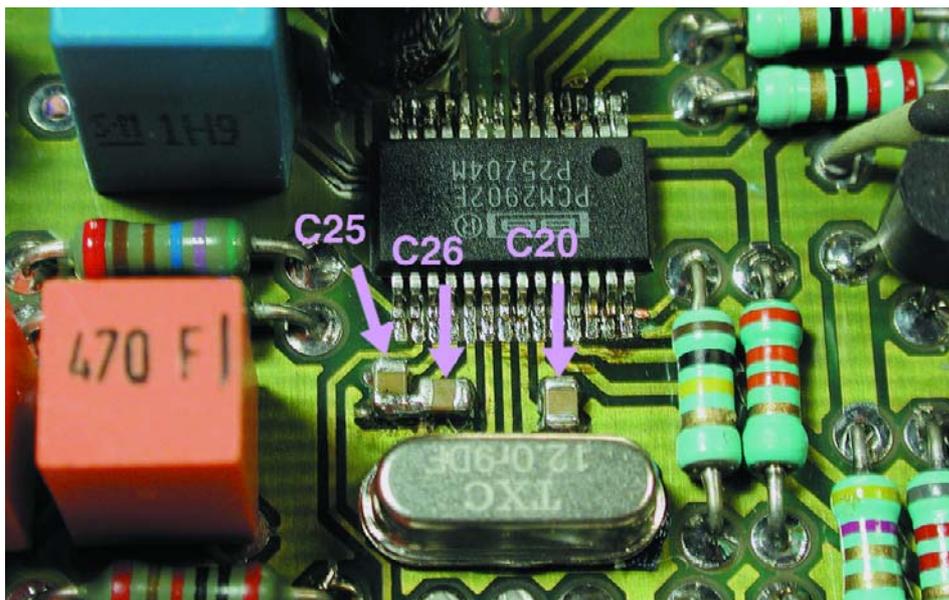


Figure 4. Le montage de IC1 et des condensateurs C20, C25 et C26 requiert une certaine expérience de soudage.

d'avoir une main bien ferme et d'utiliser un fer à souder à pointe (très) fine. Commencez par placer IC1 exactement à sa position en veillant à l'y maintenir d'une façon ou d'une autre. On soude ensuite 2 de ses pattes opposées diamétralement. Après avoir vérifié que ce circuit se trouve encore parfaitement à sa position, on pourra souder une à une ses autres broches. Utilisez de la tresse

à dessouder pour éliminer tout excédent de soudure et laissez le temps, de temps à autre, au composant de refroidir.

Cette réalisation utilise, outre les 2 circuits intégrés, quelques autres composants de type CMS, à commencer par les selfs de déparasitage de l'alimentation, L1 et L2 évoquées quelques paragraphes plus haut ainsi que 4 condensateurs de découplage pour quelques-unes des tensions d'alimentation internes au PCM2902. 3 de ces 4 mini-

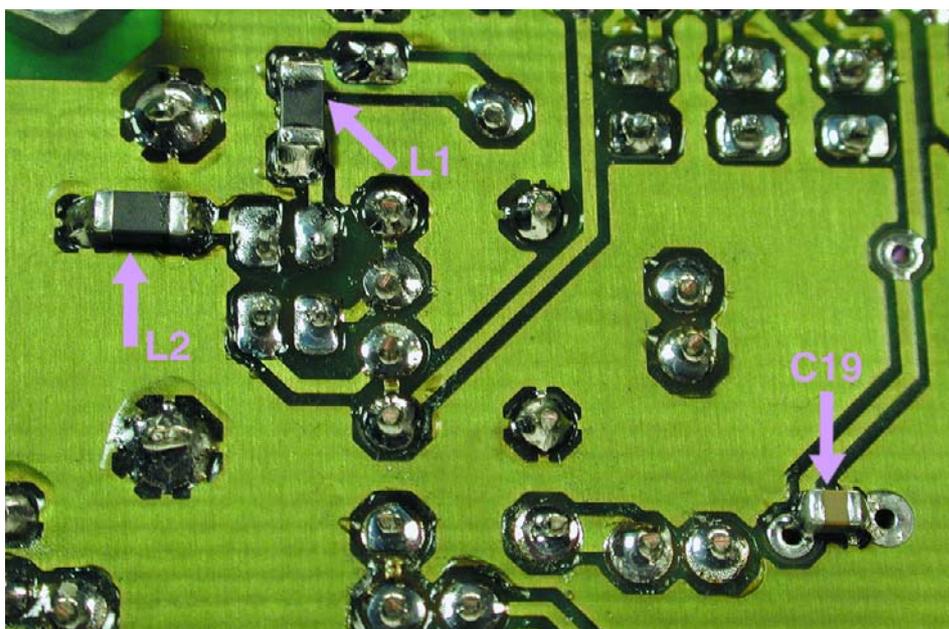


Figure 5. Les selfs CMS L1 et L2 ainsi d'ailleurs que le condensateur C19 sont à monter sur le dessous de la platine.

condensateurs de 1 μF , à savoir C20, C25 et C26, viennent se monter sur le dessus de la platine, se coinçant, comme l'illustre la photo de détail de la **figure 4**, dans l'étroit espace entre IC1 et le quartz X1. Le dernier, C19, sera lui soudé côté « soudures » (sur le dessous de la platine) entre 2 points de métallisation, les selfs L1 et L2 étant elles aussi soudées sur le dessous du circuit imprimé (cf. le détail de la **figure 5**). On pourra utiliser, dans le cas des condensateurs, soit des boîtiers de type 0603 soit une version un peu plus grande, la variante 0805. Ce dernier modèle est d'utilisation plus aisée et a partant notre préférence.

Installation

L'installation sous Windows 98SE ne devrait pas poser de problème. Le premier message apparaissant à l'écran signale la détection d'un « *USB Audio Codec* », périphérique qui requiert l'installation de pilotes (drivers). On voit ensuite arriver 3 périphériques (*devices*) (veillez à avoir votre CD-ROM Windows à portée de main) : un « *USB Composite Device* » suivi d'un « *USB Human Interface Device* » et pour terminer un « *USB Audio Device* ».

Sous l'écran de Configuration on retrouve ces 3 périphériques dans l'onglet Système/Gestion de périphérique (*System Properties/Device Manager*), comme l'illustre la recopie d'écran de la version anglaise en **figure 6**.

Il est ensuite fort probable que vous voyiez apparaître 3 commandes de réglage dans l'option Propriétés Audio (*Playback Properties*) : *Desktop*, *Wave* et *SW Synth*. Un quatrième organe de réglage, « *CD Player* » (qui sous Windows est en fait un simple lecteur de CD audio) est encore absent, mais il apparaîtra après un nouveau redémarrage de l'ordinateur.

Il ne faudra pas oublier, dans le Panneau de Configuration (Control Panel) de cocher, sous l'onglet Multimedia/Musique par CD, le repère « Valider Audio CD numérique pour ce lecteur (*Multimedia Properties /CD Music « Enable digital CD audio for this CD-ROM device* »). En cas d'interruption brève de la liaison USB il est probable que l'organe de réglage disparaisse. Ceci n'a rien d'anormal (dans le cas de Windows du moins) et est dû à la présence d'un bogue dans le dit système d'exploitation. Ce problème n'est pas résolu dans les nouvelles versions de Windows (ME et XP) sachant en outre que les dénominations de certaines des fenêtres et des organes de réglage ont changé. On pourra, pour effectuer un test rapide du fonctionnement de l'interface HID, ouvrir la fenêtre du

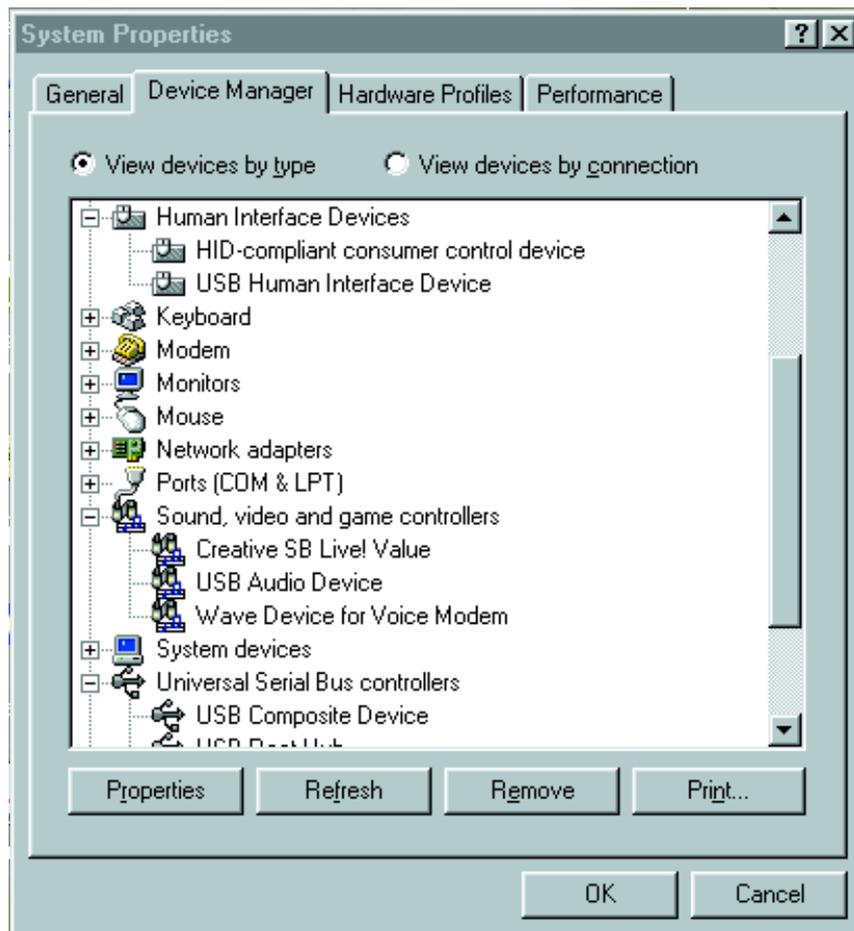


Figure 6. L'installation sur le PC n'a rien de bien sorcier (ici les Propriétés Système de la version anglaise de Windows 98SE).

réglage de volume. Si l'on établit un contact entre les broches des embases JP3 à JP5, on verra, respectivement, l'indicateur « Mute » du réglage du haut-parleur s'allumer ou s'éteindre et le potentiomètre de volume monter ou descendre.

Petit truc

Si vous voulez effectuer un test de cette mini-carte-son, vous pourrez trouver sur Internet l'un ou l'autre programme en *freeware* ou *shareware*. Notre quête nous a permis de découvrir « *audioTester* » à l'adresse :

www.sumuller.de/audiotester/.

Ce programme met à disposition un système de test audio comprenant un analyseur de spectre, un oscilloscope et un générateur de signaux. Nous avons essayé la version 1.4h (mais il existe depuis une version

2.0a) que nous ne pouvons que recommander. Le programme tourne en anglais ou en allemand.

(020178)

Interface CompactFlash pour μ C

À l'intention, entre autres, de la carte à 89S8252 Flash

Projet : Paul Goossens

Les cartes CompactFlash sont un moyen idéal pour le stockage de quantités importantes de données. Celles-ci restent présentes en mémoire des années durant sans nécessiter de tension d'alimentation. Après vous avoir proposé, en avril dernier, un disque « CompactFlash », nous vous présentons une réalisation similaire destinées aux cartes à microprocesseurs, en particulier la carte à 89S8252 Flash ô ! combien populaire utilisée dans le cadre du cours « Microcontrôleurs ».



La destination première de la présente interface est de servir d'extension à la carte à 89S8252 Flash décrite dans le numéro de décembre 2001. Il est également possible d'envisager d'utiliser cette réalisation avec d'autres cartes à microcontrôleur, mais cela implique qu'il vous faudra réaliser un connecteur d'adaptation entre notre interface CF et l'embase d'extension dont est dotée la dite carte à microcontrôleur. Notre interface permet au processeur d'écrire des données vers une carte CompactFlash et bien entendu également de les lire.

Cette approche permet d'accroître très sensiblement la taille de la mémoire mise à la disposition de l'application concernée. En effet, il existe de nos jours des cartes CompactFlash de 1 Goctet ! L'intérêt majeur de ce type de cartes est qu'elles sont faites de mémoire non volatile de sorte que des données importantes ne risquent pas d'être

perdues en cas de disparition de la tension d'alimentation.

Les domaines d'applications de cette extension vont de la centrale d'acquisition de données au dictaphone en passant par toutes les autres applications requérant une quantité importante de mémoire imaginables. Les possibilités d'applications ne sont limitées en fait que par les capacités d'imagination de l'utilisateur potentiel.

Le schéma

Le schéma de l'interface représenté en **figure 1** est renversant de simplicité. La seule « électronique » prend la forme d'un décodeur d'adresse qui se résume lui à une seule et unique porte logique, IC1.A. Ce décodeur d'adresses décode les lignes d'adresse A12 à A15 et sélectionne la carte CompactFlash lorsque ces 4 lignes se trouvent au niveau logique haut (« 1 »). On déduit de cette information que le domaine des adresses occupé par la CompactFlash va de F000_{HEX} à FFFF_{HEX}.

Le reste du montage a pour fonction d'interconnecter les lignes de la carte CF aux signaux fournis par le processeur adéquats et d'assurer l'alimentation de la carte de mémoire.

Les condensateurs C1 et C2 assurent un bon découplage de la tension d'alimentation. La tripléte constituée du condensateur C3, de la résistance R1 et de la diode D1 génère un signal de réinitialisation (*reset*) lors de l'application de la tension d'alimentation.

Ceci termine la description de l'électronique. Si vous voulez en savoir plus sur les dimensions physiques et le brochage des cartes CompactFlash, nous vous renvoyons à l'article cité en référence [1].

Nous vous proposons, en **figure 2**, le dessin des pistes et la sérigraphie de l'implantation des composants de la platine double face à trous métallisés dessinée à l'intention de cette réalisation. Le mini-connecteur à 50 contacts destiné à recevoir la carte CF et l'embase à 34 contacts servant à la connexion vers la carte à 89S8252 Flash occupent près de la moitié de la surface de cette platine d'interface.

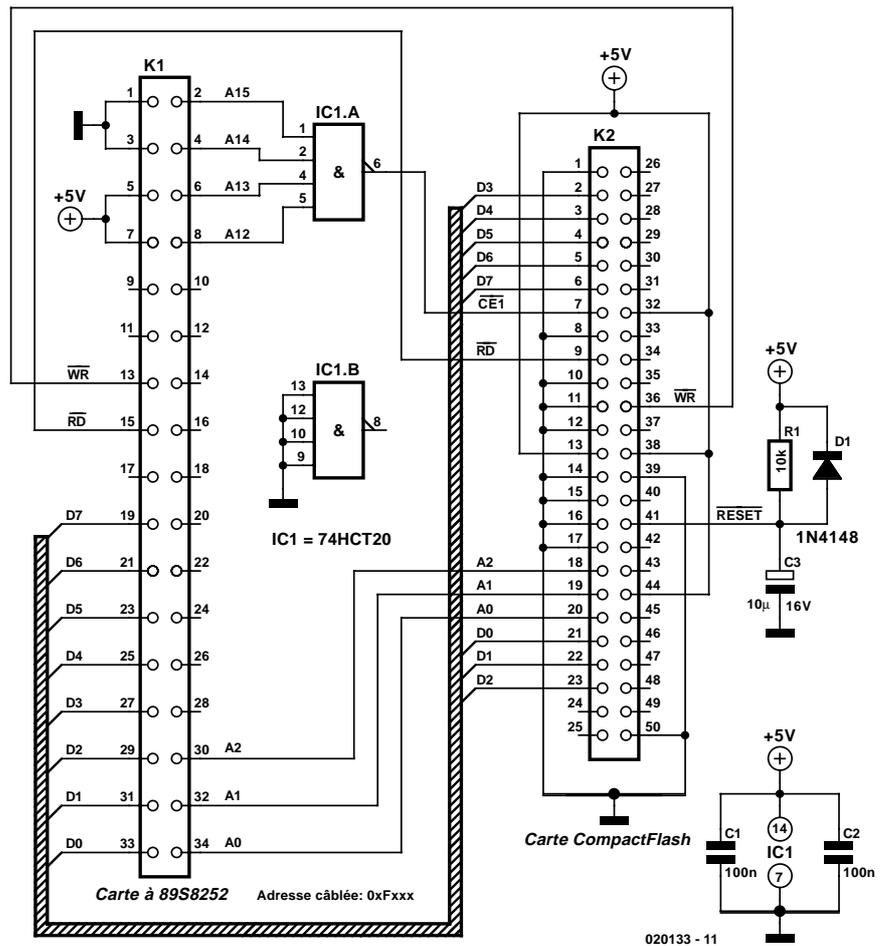


Figure 1. L'électronique de l'interface CompactFlash se résume en fait à une paire de connecteurs et à un rien d'électronique additionnelle.

L'implantation des composants sur le circuit imprimé n'a rien de bien sorcier, seule la soudure du connecteur destiné à la carte CF requiert une attention particulière en raison du très faible écartement entre les broches de ce dernier.

La liaison entre l'interface (K1) et la platine de la carte 89S8252 Flash (K8) se fait à l'aide d'un morceau de câble plat à 24 conducteurs doté à chacune de ses extrémités d'un connecteur à sertir. La photo en début d'article montre comment disposer cette mini-interface pour carte CF à côté de la dite carte d'expérimentation à 89S8252. La carte CF est enfichée dans le connecteur K2 de manière à ce que son étiquette soit visible d'en haut (cf. la photo). Il est à noter que cette extension NE FONCTIONNE PAS si l'on a connecté un affichage LCD à la carte à 89S8252 Flash !

Le logiciel

En l'absence de programme adéquat, un processeur ne saurait que faire du fait qu'il se trouve connecté à une carte CompactFlash prise dans sa cartographie de mémoire. Il faut que le programmeur donne au processeur l'instruction d'écrire ou de lire des données sans même parler de lui expliquer comment s'y prendre pour cela.

Les spécifications complètes de l'interface CompactFlash sont relativement complexes, ce qui pourrait « bloquer » certains d'entre

Tableau 1.

Les principales fonctions de l'interface CompactFlash.

Instruction	Nom	Code
Lecture de secteur(s) (Read Sector(s))		0x20
Écriture de secteurs (Write Secto(s))		0x30
Identification lecteur (Identify Drive)		0xEC

nos lecteurs dans leur tentative de procéder à leurs propres expériences avec une carte CompactFlash. De manière à vous débarrasser de cette « horreur de l'eau froide » nous avons écrit un petit programme de démonstration mettant à contribution les fonctions

les plus importantes (et utiles) de l'interface. Le **tableau 1** récapitule les différentes fonctions utilisées. Ce petit programme pourra servir d'exemple lorsque vous déciderez d'écrire vos propres programmes, vous évitant d'avoir à passer en revue, péniblement, toutes les spécifications.

Si vous tenez à tout connaître des spécifications de l'interface CompactFlash, nous ne pouvons que vous conseiller de faire un saut sur le site Internet de la CompactFlash Association, où elles sont dispo-

nibles au téléchargement gratuitement. L'adresse est la suivante : www.compactflash.org.

Le code-source de ce programme et le fichier de programmation sont disponibles au téléchargement sur le site Internet d'Elektor ou sur disquette (EPS020133-11) auprès des adresses habituelles. Le programme en C a été écrit sous le compilateur C de Tasking. Il faudra, pour pouvoir utiliser le code-source avec un autre compilateur, modifier l'un ou l'autre détail. Citons, au nombre de ces adaptations, la déclaration des

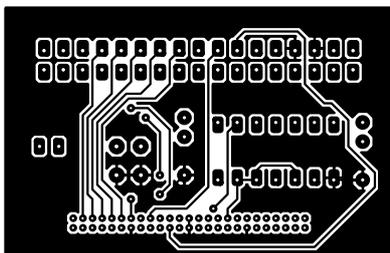
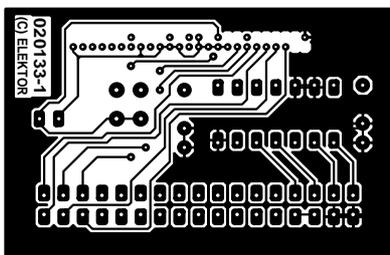
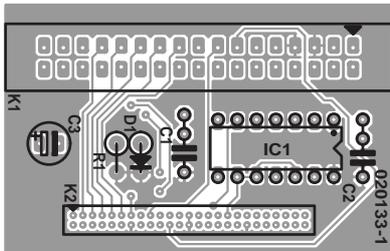
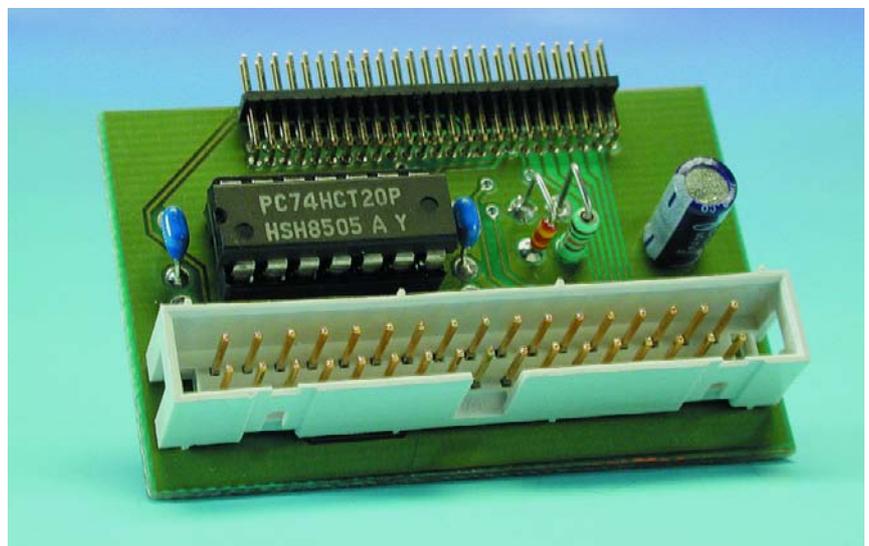


Figure 2. Dessin des pistes et sérigraphie de l'implantation des composants de la platine double face à trous métallisés.



Liste des composants

Résistances :

R1 = 10 kΩ

Condensateurs :

C1, C2 = 100 nF
C3 = 10 μF/16 V radial

Semi-conducteurs :

D1 = 1N4148
IC1 = 74HCT20

Divers :

K1 = embase à 2 rangées de 17 contacts
K2 = embase grid à 50 contacts en équerre au pas de 0,05" (telle que, par exemple, Farnell n° cde 307-8127)

Tableau 2.

Les registres de l'interface CompactFlash.

Adresse	Lecture	Écriture
F000h (atrDATMSB)	D8 à D15	D8 à D15
F006h (atrALTERN) (atrDEVCTRL)	Alternate Status	Contrôle d'appareil (Device control)
F007h (atrDRIVEAD)	Adresse Lecteur (Drive Address)	
F008h (atrDATLSB)	Données (Data)	Données (Data)
F009h (atrERROR) (atrFEATURE)	Erreur (Error)	Caractéristique
F00Ah (atrSECCNT)	Numéro de secteur (Sector Number)	Numéro de secteur (Sector Number)
F00Bh (atrSECNR)	Sector Number	Sector Number
F00Ch (atrCYLLOW)	Cylindre Bas (Cylinder Low)	Cylindre Bas (Cylinder Low)
F00Dh (atrCYLHIG)	Cylindre Haut (Cylinder High)	Cylindre Haut (Cylinder High)
F00Eh (atrDRHEAD)	Lecteur/Tête (Drive/Head)	Lecteur/Tête (Drive/Head)
F00Fh (atrSTATUS) (atrCOMMAND)	État (Status)	Instruction (Command)

Tableau 3.

Les instructions du programme cité en exemple.

?	Help	Explique succinctement toutes les fonctions du programmer
D	Display	Montre le contenu du tampon sous format hexadécimal et ASCII
I	Info	Montre les principales caractéristiques de la carte CompactFlash connectée
R	Read	Lit un secteur de la carte CompactFlash et en place le contenu dans le tampon
W	Write	Écrit le contenu du tampon vers un secteur de la carte CompactFlash
C	Cylinder	Change le cylindre sélectionné à cet instant
H	Head	Change la tête sélectionnée à cet instant
S	Sector	Change le secteur sélectionné à cet instant

registres de la CompactFlash.

Le code-source de ce programme et le fichier de programmation (à mettre dans le microcontrôleur) sont disponibles au téléchargement sur notre site (www.elektor.fr) ou sur disquette (EPS020133-11) pour ceux d'entre nos lecteurs qui n'auraient pas accès à Internet. Le programme en C a été écrit sous le compilateur C de Tasking. Il faudra, pour pouvoir utiliser le code-source avec un autre compilateur C, procéder à quelques modifications de détails. Citons, au nombre d'éventuels changements à effectuer, la déclaration des registres de la carte FC. Le transfert du fichier de programmation vers le AT89S8252 pourra se faire directement à l'aide de « MicroFlash », un utilitaire qu'il n'est sans doute plus nécessaire de vous présenter si le sujet vous tient à coeur. HyperTerminal, un programme tournant sous Windows sur tout PC, permettra de tester les différentes fonctions de l'interface. Le paramétrage de HyperTerminal sera : 9 600 bauds, 1 bit de début (*start bit*), 8 bits de données sans parité.

Lors du lancement du programme, on a réinitialisation de la carte CF et prise en compte des paramètres requis. On a en outre lecture du secteur situé à l'adresse CYL 1, HEAD 1, SECTOR 1, informations qui sont ensuite stockées dans le tampon. Le programme attend ensuite une instruction de l'utilisateur.

Que reste-t-il à ajouter

Il est prudent, lors de l'utilisation d'une carte CompactFlash, de com-

mencer par effectuer une sauvegarde du contenu de cette dernière de manière à pouvoir expérimenter à loisir sans la moindre arrière-pensée. Une fois que l'on aura écrit sur cette carte CF par le biais d'un système à microcontrôleur il ne faudra pas être étonné qu'il faille, avant de pouvoir réutiliser cette carte sur un autre système, un PC ou un appareil photo numérique par exemple, la reformater avant qu'elle ne puisse être utilisée normalement. Cela tient au fait que les appareils photo et les PC utilisent un système de fichiers pour pouvoir ordonner les données sous la forme de fichiers et de répertoires

sur une carte CF. Cette structure de classement occupe une partie de la carte CF car il lui faut se rappeler où commencent les différents répertoires et fichiers. Il arrive aussi que les fichiers soient distribués par petits morceaux sur l'ensemble de la carte. Le système de fichiers mémorise alors des données additionnelles qui indiquent dans quel ordre doit se faire la lecture des secteurs concernés en vue de pouvoir procéder à la prise en compte d'un fichier donné.

Nous ne pouvons pas, dans le cadre de cet article, entrer dans le détail de la structure des différents systèmes de fichiers. C'est à dessein que notre programme servant d'exemple n'utilise pas de structure de fichiers, ce qui permet d'écrire un programme simple qui remplit parfaitement sa tâche, à savoir illustrer clairement comment se fait l'écriture de secteurs et comment on en effectue ensuite la lecture.

Si vous êtes intéressé par le sujet, la structure de fichiers, qui est en fait l'essence du système d'exploitation, utilisée par les systèmes tournant sous Windows, nous pouvons vous donner une adresse où trouver une documentation exhaustive :

www.microsoft.com/hwdev/download/hardware/FATGEN103.doc

(020133)

Bibliographie :

- [1] « Disque CompactFlash », Elektor n°286, avril 2002, page 48 et suivantes

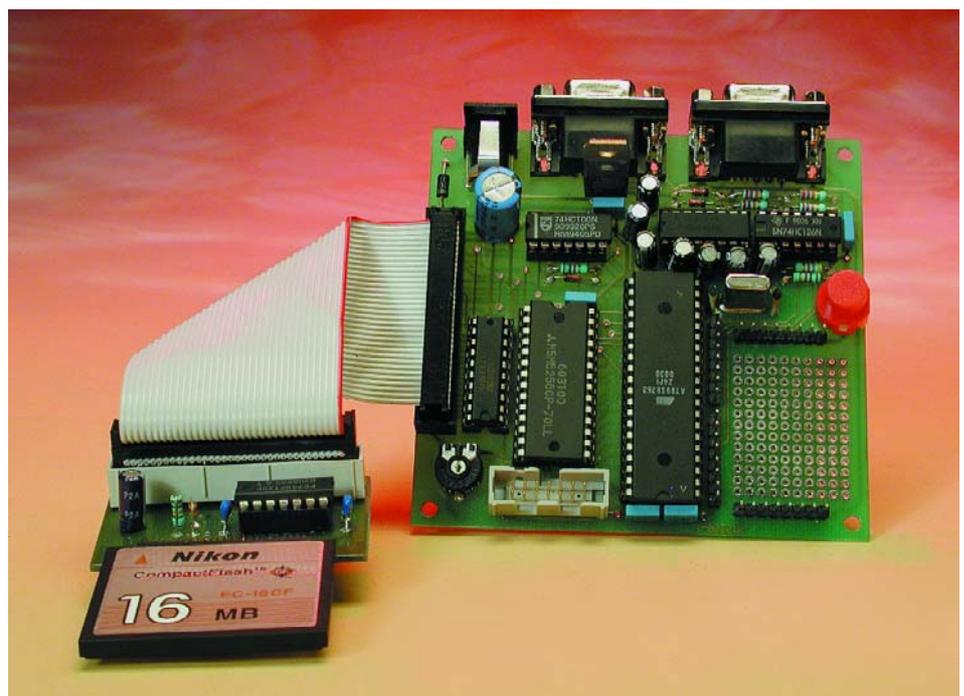


Figure 3. Voici comment interconnecter cette réalisation à la carte à 89S8252 Flash.

Adaptateur pour diagnostic de véhicule (2)

Logiciel Windows pour l'interface OBD2

Gerhard Müller

L'interface OBD2 pour diagnostic de véhicule décrite dans le numéro précédent fournit les données à l'interface série d'un PC sous forme de caractères ASCII. Bien qu'il soit possible de les lire avec un programme usuel d'émulation de terminal, l'évaluation des codes ne sera pas très conviviale. Le programme Windows – gratuit – présenté ici l'est bien davantage.



Figure 1. Page de démarrage du programme de diagnostic OBD2 de Scantool.net.

Rien à redire en principe au format ASCII des données fournies par l'interface OBD à l'ordinateur. Mais le ou les ordinateurs du véhicule – généralement le bloc électronique

de commande du moteur, bien qu'on ne puisse pas exclure la présence dans le véhicule d'autres blocs de commande connectés par

un système de bus – répond(ent) aux interrogations du programme de terminal par des caractères hexadécimaux peu révélateurs. Il faut donc disposer d'un programme qui, primo, facilite la saisie de commandes d'interrogation et, secundo, rende tout d'abord les réponses lisibles. Ce programme afficherait par exemple, outre les numéros des codes d'erreur du diagnostic, une description en clair qui remplacerait dans un premier temps les listes et les manuels.

Comme l'expliquait déjà l'article précédent, la création d'un tel programme est facilitée tant par la précision des définitions de la norme OBD2 que par les fonctions « câblées » de la puce ELM. La puce ELM effectue l'initialisation du bloc électronique de commande du véhicule, une fonction normalement indispensable pour avoir accès à la prise de diagnostic OBD2, et veille ensuite au maintien de la communication. L'initialisation consiste à envoyer la valeur hexadécimale 0x33 à 5 bauds par la connexion L broche 15 (connexion K broche 7 dans les

blocs de commande des véhicules les plus récents), puis à attendre la réponse en hexadécimal 0x55 (à 10 400 bauds). La puce ELM garde la liaison initialisée en envoyant en mode automatique continu une information fictive toutes les 5 secondes. Le calcul du total de contrôle (CRC) des informations envoyées est aussi automatique.

Les normes du protocole OBD2 décrites dans l'article précédent et la description de la communication avec la puce ELM323 (voir les références en fin d'article) nous montrent que le programme peut être basé sur les blocs fonctionnels suivants :

- lecture et effacement des codes d'erreur
- lecture et représentation éventuelle des données en temps réel des capteurs
- lecture des données « freeze frame » (instantanés)
- saisie des résultats des tests effectués par l'électronique du véhicule ainsi qu'éventuellement
- lecture de données supplémentaires qui dépendent du fabricant du véhicule ou ne peuvent faire l'objet d'une interrogation qu'avec des véhicules récents, par exemple le numéro du châssis (VIN ou FIN).

Logiciel Windows gratuit

Un programme de ce genre a été développé sous forme de source ouverte par les programmeurs de ScanTool.net (www.scantool.net). La version en langue allemande de ce programme a été réalisée en collaboration avec l'auteur de cet article (figure 1). Pour respecter la charte du Logiciel Libre, les compilateurs et les outils de développement utilisés sont tous disponibles gratuitement, ce qui facilite la portabilité du code source sous d'autres systèmes d'exploitation que Windows. Récemment encore, ces restrictions ne permettaient au programme de tourner sous Windows que lors d'une session DOS, le problème bien connu des sous-programmes d'accès direct au

matériel du PC (par exemple l'écran ou l'interface série) ; cet inconvénient, toutefois, n'en était pas un sous Windows 95, 98 et ME. Sous Windows NT, 2000 et XP, qui sont dépourvus de mode DOS, il fallait par contre lancer cette version à partir d'une disquette d'initialisation comportant le système d'exploitation DOS. ScanTool.net a réussi entre-temps à combler cette lacune. Le programme fonctionne dans une fenêtre Windows à partir de la version 1.04. Les 2 versions (en anglais) seront disponibles dans la partie Téléchargement du site Web d'Elektor pour le numéro de décembre 2002. Le code source en C y sera aussi disponible. Il suffit de cliquer sur l'option de menu TÉLÉCHARGEMENTS du site Web (www.elektor.fr) puis de choisir Elektor Déc. 2002. Vous obtiendrez la liste des fichiers disponibles pour le téléchargement.

Les indications du fichier « readme.txt » (en anglais) permettent de compiler soi-même le code source sans aucun problème. Les sources d'approvisionnement du compilateur libre gcc et des autres outils, par exemple du progiciel Allegro qui permet de créer l'interface utilisateur du programme, y sont mentionnées, ce qui permet à chacun de pousser plus loin le développement. Pour celui qui reculerait devant cet effort et n'est pas désireux d'approfondir les secrets du « On-Board Diagnostic », il suffit de consulter de temps à autre l'adresse Web pour les mises à jour.

Fonctions

La version actuelle du programme permet de lire et de réinitialiser les codes d'erreur ainsi que d'interroger l'état actuel de 11 capteurs au plus. Les fonctions énumérées plus haut, par exemple la lecture d'un enregistrement « Freeze Frame » (créé et enregistré dans le-bloc de commande à partir des données actuelles des capteurs lorsqu'un code d'erreur est généré) et la représentation des résultats d'un auto-contrôle du bloc de commande, sont déjà présentes dans l'interface utilisateur du programme mais ne sont pas encore implémentées. Il manque aussi une fonction d'enregistrement

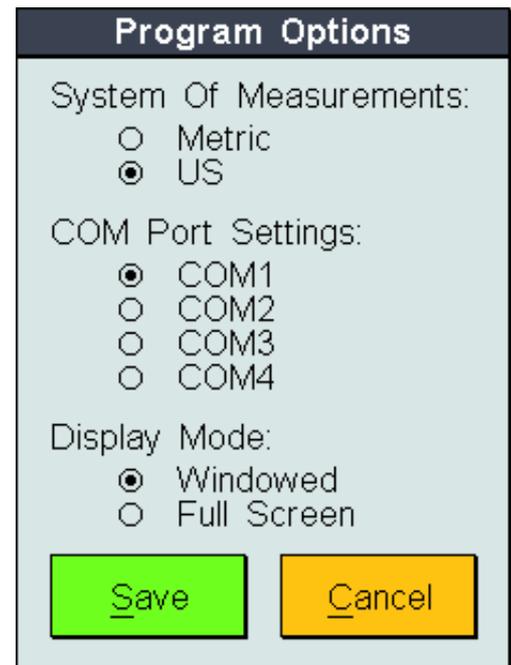


Figure 2. Possibilités de réglage sous « Options ».

des données. La version DOS ne permet que de sauvegarder une copie d'écran dans le répertoire actuel sous le nom de fichier « screenshotX.pcx » (« X » est un nombre incrémenté consécutivement) au moyen du raccourci de clavier Ctrl + Alt + Impr écran (Prt Sc). Dans la version Windows, il suffit de presser la touche Impr écran pour créer la copie d'écran dans le presse-papiers.

Les boutons des fonctions « Read Codes » (Codes d'erreur) ou « Sensor Data » (Données capteurs) ne sont actifs que si le PC est relié à la prise du véhicule par l'interface OBD2 et si l'allumage est en marche. On obtient sinon un message demandant de configurer le port sériel utilisé. Il est plus logique de procéder à cette opération en premier en cliquant sur le bouton « Options » (Options) qui permet aussi de commuter l'affichage des données des capteurs entre le système métrique et le système d'unités US (figure 2). Le fichier « scantool.cfg » de mémorisation des réglages créé à ce moment-là contient aussi le mode d'interrogation (« ON » ou « OFF » = Marche/Arrêt) des 11 champs représentant les données des capteurs. Les messages d'erreur -y compris ceux qui sont engendrés lorsque le programme ne démarre pas correctement- sont enregistrés dans le fichier « log.txt ».

Les 2 autres fichiers, « codes.dat » et « scantool.dat » doivent se trouver dans le répertoire du programme « scantool.exe ». Ils sont chargés avec le programme lors du lancement de celui-ci. Le fichier « codes.dat » contient en fait la description en clair de

3107 codes d'erreur, soit tous les codes standardisés et un grand nombre de codes spécifiques aux fabricants. Ce fichier comporte aussi un champ de données (encore vierge) destiné à contenir des conseils sur l'élimination des erreurs éventuellement détectées. « Simulate » (Simulation) à la page « Read Codes » du programme (figure 3) fait apparaître une sélection des textes qui suivent les codes d'erreur. Le bouton « Read » (Lecture) permet de lire les véritables codes d'erreur s'il s'en trouve et les affiche à côté de l'état de la lampe d'erreur. Avant d'effacer les codes d'erreur, le bouton « Clear » (Effacement) fait apparaître, conformément aux prescriptions de sécurité, une demande de confirmation de l'effacement. Pour la classification alphanumérique des codes d'erreur, veuillez consulter les articles précédents et les pages Internet mentionnées à la fin de cet article. Le texte en clair qui les accompagne est aussi facile à interpréter que les désignations dans l'affichage des capteurs. Seule la désignation « Fuel System Status » (état de l'injection) peut poser un problème dans le cas des ces derniers. Il s'agit de l'état de la régulation du catalyseur. Celui-ci peut avoir pour valeur « LOOP OPEN », ou « LOOP CLOSED » (boucle ouverte/fermée) et quelques états d'erreur. Lors du démarrage à froid du véhicule, la régulation du catalyseur se trouve en effet dans un état de boucle ouverte ; en fait, les sondes lambda sont encore froides et ne fournissent pas encore de signal de régulation utilisable (boucle de régulation ouverte). Le régime « en boucle fermée » (boucle de régulation fermée) n'est atteint qu'à la température de fonctionnement des sondes lambda et ce relativement rapidement lorsque –comme c'est cela est devenu le cas général– les sondes lambda sont réchauffées par une source externe.

Le fichier « scantool.dat » contient les matrices binaires des graphiques et boutons utilisés, leur palette de couleurs, ainsi que les polices de caractères utilisées. Pour simplifier la portabilité dans d'autres langues nationales, il est prévu de réunir les textes, qui se trouvent actuellement dans le programme principal, dans un fichier supplémentaire, ce qui permettrait de limiter les modifications à ce fichier lors de la traduction. Le format des 2 fichiers « *.dat » est comprimé et il faut les éditer au moyen de l'outil « Grabber » du progiciel Allegro. Il devient alors possible de placer ses propres diagnostics dans le champ encore vide du fichier « codes.dat » sans recompiler. Ce champ, « Possible Causes And Known Solutions » (Causes possibles et remèdes connus), qui a été mentionné plus haut, fait partie du

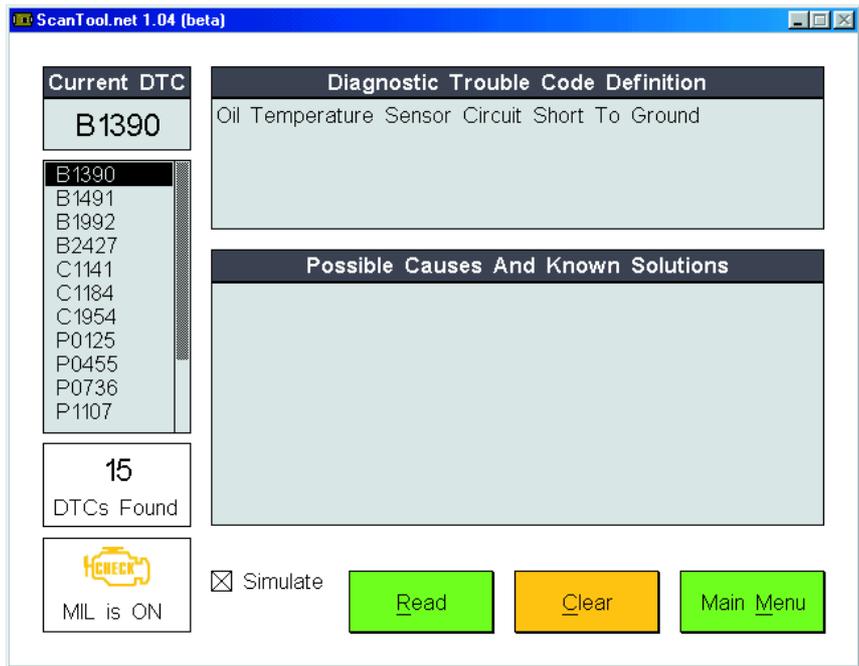


Figure 3. Affichage des codes d'erreur et description.

domaine des codes d'erreur. Des digressions supplémentaires sur la manière de programmer ses propres extensions, voire d'écrire un autre programme déborderaient du cadre de cet article. Le code source est toutefois si bien commenté et si modulaire que même les programmeurs du dimanche s'y retrouveront. Vous trouverez les informations

récentes sur OBD2 qui vous sont nécessaires sur Internet, par exemple sous <http://www.obd-2.de>, repère « Techn. Info » (Informations Techniques en allemand, pas de version anglaise pour l'instant). Notons que la fonction « Freeze Frame » n'est pas encore implémentée dans la version 1.04 du programme considérée comme une version beta.,

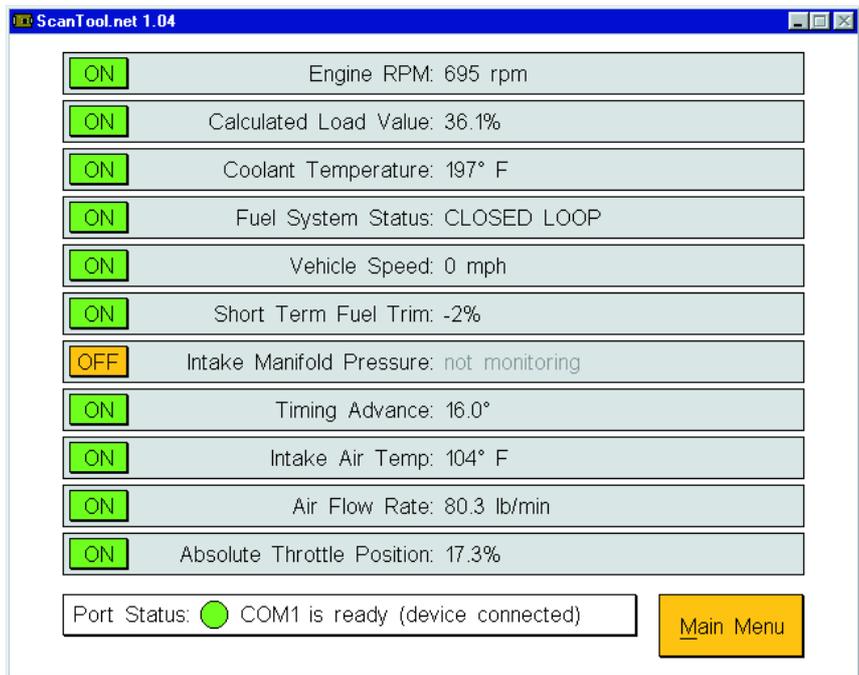


Figure 4. Page des données capteurs du programme de Scantool.net.

La poursuite du développement évoquée plus haut a pour but d'inclure les fonctions manquantes dans le programme et d'en offrir également une version Linux dans un délai raisonnable. Toute modification effectuée par l'utilisateur doit respecter les termes de la licence GNU pour logiciel libre.

Quelques conseils pratiques

Si l'on s'imagine qu'avec l'affichage et la description des codes d'erreur les réparation sont devenues un jeu d'enfant, il faudra malheureusement déchanter. Pour éviter l'échange souvent coûteux d'un capteur au comportement problématique, il faut procéder tout d'abord à une recherche

systematique de l'erreur. Le câble et le connecteur sont-ils intacts ? De nombreuses descriptions des codes d'erreur mentionnent déjà la possibilité d'une rupture du câble ou d'un court-circuit à la masse ou à la tension de bord. Le message n'est-il que le symptôme d'une autre erreur ? Un mélange erroné air-carburant dû par exemple à une défektivité du dispositif de mesure de la quantité d'air n'est pas du tout du goût des sondes lambda. Les manuels du fabricant destinés aux garages permettront de s'épargner bien du temps, des efforts et peut-être des dépenses inutiles. Ils contiennent en effet des arbres de décision qui permettent de localiser l'erreur avec précision. Mais la lecture des codes d'erreur peut aussi s'avérer utile lorsque le véhi-

cule doit être confié à un atelier de réparation : elle permet de décider si une réparation immédiate s'impose ou s'il est possible d'utiliser encore le véhicule pendant quelques jours pour se rendre au travail. En effet, certaines erreurs, par exemple des ratés de l'allumage ou un mélange trop pauvre, provoquent des dommages coûteux qui rendent préférable une réparation immédiate. Mais quel que soit le cas, le programme d'urgence du bloc électronique de commande du moteur n'assure pas un fonctionnement économique. Une lecture « désintéressée » de la mémoire d'erreurs peut aussi contribuer à éviter les coûts de réparation superflus. Comme l'ont montré à maintes reprises les tests d'ateliers de réparation, il est tout à fait raisonnable d'avoir recours si nécessaire à ses propres données OBD. En d'autres termes, « prudence est mère de sûreté »

(020138-3)

« Manips tachymétriques » et systèmes de diagnostic

Les associations d'automobilistes ne sont pas seules à soupçonner de nombreux vendeurs de véhicules d'occasion de « bidouiller » le tachymètre d'un véhicule pour en gonfler le prix de vente.

Selon les estimations de l'organisme de certification allemand DEKRA, le compteur kilométrique d'un tiers des véhicules serait trafiqué. On trouve sur Internet et dans les petites annonces des offres révélatrices (hélas ???, rares en version française...), par exemple sous « programmation de compteurs digitaux ». Il s'agit, pour appeler un chat un chat, de l'utilisation d'un ordinateur portable et d'un logiciel permettant aux clients de positionner en quelques minutes le tachymètre électronique sur le nombre de kilomètres désiré.

Les lecteurs d'Elektor et les électroniciens en général se demanderont bien sûr immédiatement s'il est possible de détecter un compteur faussé en comparant sa valeur aux données du système de diagnostic. Il est en effet arrivé par le passé que les données des indicateurs d'intervalles de service des véhicules de certains fabricants présentent une différence par rapport au kilométrage ou que les données enregistrées dans le tachymètre électronique changent lorsqu'il était intégralement remplacé. Mais l'enregistrement systématique d'une copie du kilométrage dans d'autres blocs électroniques de commande, par exemple dans un module de l'éclairage, dans la commande de la boîte de vitesses, dans le système antiblocage, voire dans le système de verrouillage électronique est un phénomène récent. La plupart des constructeurs européens appliquent ces mesures de sécurité aux véhicules de série depuis 2001, mais il n'en va pas de même chez les fabricants d'Extrême-Orient. Les véhicules les plus récents équipés d'OBD2 comportent des informations supplémentaires sur le kilométrage dans PID 21 et 31 (extension de Parameter Identification).

Mais une petite excursion sur Internet fera bien vite déchanter ceux qui s'imaginaient que ces mesures permettraient enfin de mettre un terme aux agissements des faussaires du kilométrage.

Les sites Web (en langue allemande) des entreprises assurant le « Tachoservice » révèlent quelles données de chaque modèle de véhicule permettent d'ajuster le kilométrage sans laisser de traces. Ces précisions sont destinées à convaincre le client potentiel que « la détection de manipulations du tachymètre est pratiquement exclue lorsque les techniques les plus modernes sont appliquées avec suffisamment de savoir-faire ».

Il ne s'agit, hélas, pas d'un argument publicitaire. S'il en coûte environ 100 € pour une « simple » diminution du kilométrage, les mesures de sécurité basées sur des copies secrètes de l'état des compteurs ne conduisent qu'à un surcroît de travail de « sabotage » pour lequel la facture se monte actuellement à 800 €. Et lorsque, dans 3 ans, les conducteurs de véhicules au bénéfice d'un contrat de crédit-bail se précipiteront en masse pour un « étalonnage », il n'en coûtera même plus 200 €...

Il serait plus facile d'attaquer le problème s'il était possible, comme dans certains pays, de vérifier l'état du véhicule par l'entremise de bases de données centralisées. Il devient beaucoup plus facile de repérer les fraudeurs lorsque les données sur les contrôles techniques (émissions de gaz d'échappement et généraux), les révisions, le changement de détenteur, les dommages et les déclarations d'accident peuvent être rassemblées et évaluées. Le respect des libertés individuelles ancré dans diverses lois sur la protection des données ne profite malheureusement pas qu'aux honnêtes gens.

Il ne faudrait toutefois pas trop minimiser l'efficacité des mesures techniques. Ce sont les assurances (et non les pouvoirs législatifs) qui ont imposé les systèmes électroniques d'interdiction de démarrage. Ces derniers peuvent être « craqués » « lorsque les techniques les plus modernes sont appliquées », mais le nombre de véhicules volés a diminué.

(La Rédaction)

Fusible électronique

La sécurité et le luxe

Ulrich Licht

Une sonde qui dérape au milieu de l'échec de conducteurs d'un montage réalisé sur une platine d'expérimentation, il n'en faut pas plus pour provoquer un court-circuit aux conséquences les plus désastreuses pour les semi-conducteurs les plus sensibles.

Si, comme elle est protégée contre un court-circuit, l'alimentation est bien capable de se tirer d'affaire, il n'en va pas de même en ce qui concerne la charge. C'est là la raison du développement de ce fusible électronique qui permettra d'ajuster le courant de charge et de le couper quasi-instantanément comme le ferait un fusible classique.

Un premier coup d'oeil au schéma de la **figure 1** nous apprend que nous ne nous trouvons pas en présence d'une application de fusible électronique standard. Le « fusible » prend ici la forme d'un BTS432E2 de Infineon, dont on pourra trouver la fiche de caractéristiques à l'adresse :

www.infineon.com/cmuc_upload/0/000/008/

685/bts432e2.pdf.

Il s'agit en l'occurrence d'un membre de la famille PROFET, un FET de puissance à canal N doté de toute une panoplie de fonctions de sécurisation. En mode de fonctionnement normal, le FET est mis en et hors-fonction par le biais de son entrée IN. Dès qu'il constate une situation anormale, sous-tension, surtension, mise en court-circuit vers le plus de la tension d'alimentation ou la masse, mise en l'air d'une sortie ou dépassement de température, le BTS432E2 prend les mesures qui s'imposent.

La **figure 2** vous propose un coup d'oeil à l'intérieur de ce composant. Le second composant important dans le sous-ensemble « Commutateur & Mesure de courant » s'appelle IC1; il s'agit d'un LM3813. Ce

détecteur de tension de précision surveille le courant qui circule par la ligne de masse en introduisant la chute de potentiel la plus faible possible et produit un signal MLI (modulé en largeur d'impulsion = PWM pour *Pulse Width Modulation* en anglais) fonction du courant dérivé par le biais des lignes Sense+ et Sense-. Un courant positif produit un rapport de modulation en largeur d'impulsion compris entre 0,5 (à 0 A) et 0,955 (à 1 A). Si le courant circule dans l'autre sens, le rapport de modulation en largeur d'impulsion varie entre 0,5 à 0,045 (à -1 A). Signalons au passage qu'il existe des versions ±1 et ±7 A de ce composant. La durée d'échantillonnage (et de ce fait le temps de réaction minimum à la circulation d'un courant trop important par la charge) est

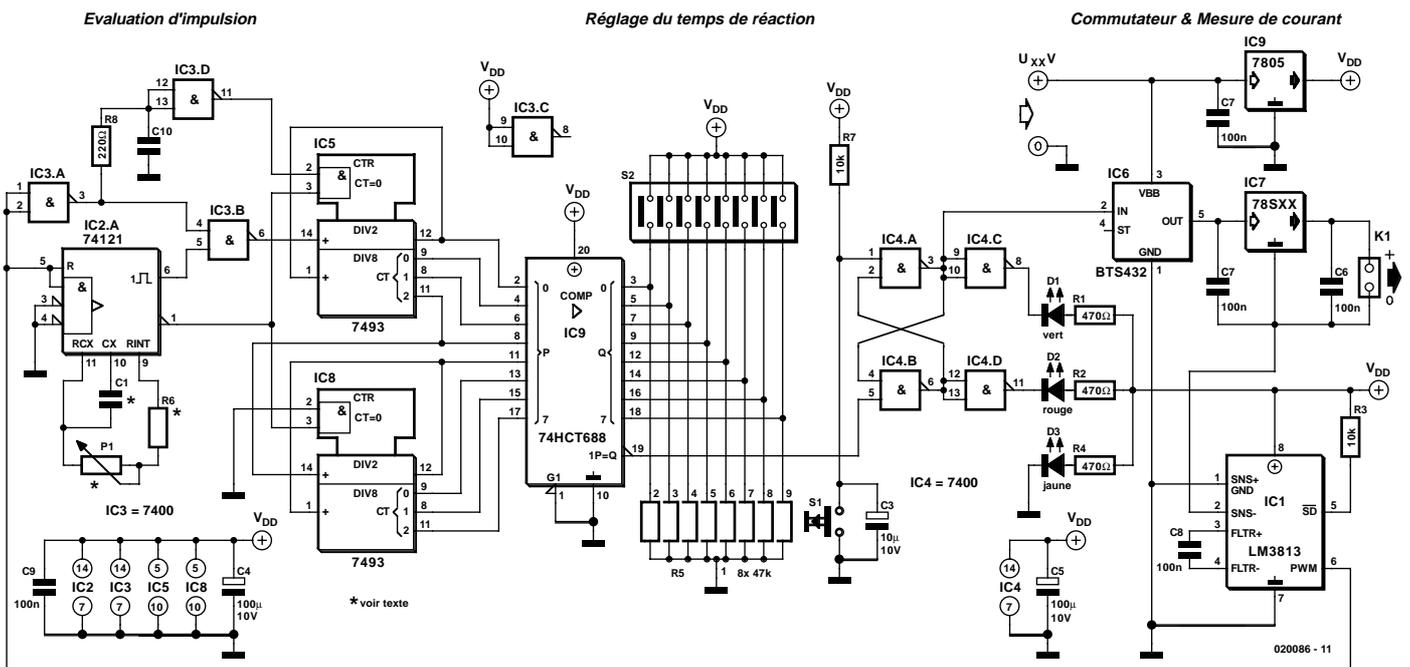


Figure 1. L'électronique est subdivisée en 3 sous-ensembles, évaluation d'impulsion, réglage du temps de réaction et commutateur et mesure de courant.

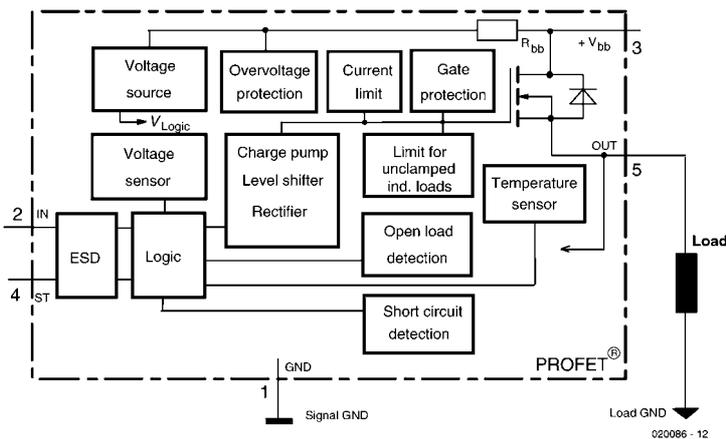


Figure 2. Structure interne du PROFET.

de 50 ms; si l'on a besoin d'un fusible électronique encore plus rapide, il faudra faire appel au LM3813, la durée de réaction de ce composant n'étant que de 6 ms. On trouvera les fiches de caractéristiques de leurs homologues pour les mesures côté chaud, sur le site de National Semiconductor aux adresses suivantes :

www.national.com/ds/LM/LM3812.pdf
www.national.com/ds/LM/LM3814.pdf

IC9 est un régulateur de tension fixe tout ce qu'il y a de plus classique, chargé de fournir la tension d'alimentation régulée de +5 V requise par l'électronique, le régulateur de tension IC7 définissant quant à lui la tension de sortie en fonction du type utilisé.

Compare et compte

Le signal MLI proportionnel au courant de charge I est appliqué à la bascule monostable IC2. Le LM3813 déclenche le multivibrateur monostable du type 74121 toutes les 50 ms. On trouve aux sorties une impulsion, P_{MAX}, dont la longueur est déterminée par la position du potentiomètre P1.

IC3.A inverse le signal MLI, IC3.B en effectuant ensuite la comparaison avec le signal P_{MAX}. Tant que le

courant de charge reste à l'intérieur de la fenêtre définie, la sortie de IC3.B reste en permanence au niveau haut, si tel n'est pas le cas, on trouvera sur cette broche une impulsion de niveau bas dont la longueur est fonction de la durée pendant laquelle le signal MLI dépasse la tension de référence P_{MAX}.

Les compteurs binaires à 4 bits du type 7493 montés en cascade, IC5 et IC8, utilisent cette impulsion de niveau bas en tant que signal d'horloge. Le retard de transfert introduit par le réseau RC et IC3.D que subit le signal MLI (PWM) évite que l'impulsion MLI ne puisse être considérée comme une impulsion de comptage avant que IC2 ne puisse fournir une impulsion de comparaison; il pousse pour ainsi dire le temps de réaction du 74121.

En aval de la paire de compteurs binaires on trouve un comparateur de grandeurs à 8 bits du type 74HCT688. La sortie $\overline{P=Q}$ devient active (au niveau bas) lorsque le patron de bit appliqué aux entrées P correspond à la valeur (binaire) prédéfinie par les positions des contacts de l'interrupteur DIP, patron de bits appliqué lui aux entrées Q. Avec la variante 50 ms de la jauge de courant (car c'est ainsi que National

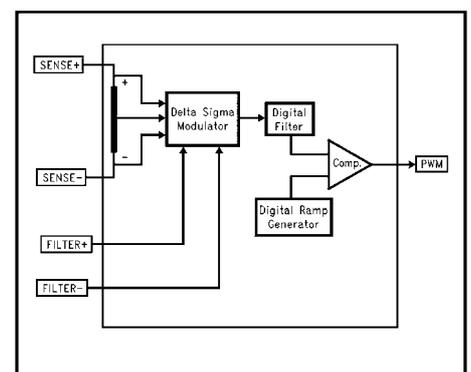
Semiconductor a baptisé son composant) il est possible d'ajuster le temps de réaction de notre fusible électronique entre 50 ms et 12,8 s et ce par pas de 50 ms, la variante 6 ms permettant de définir ce retard d'entrée en fonction entre 6 ms et 1,5 s, par pas de... 6 ms bien évidemment. S'il devait se faire que le courant maximal admissible soit dépassé sans que cependant la valeur de comptage prédéfinie ne soit atteinte (le courant de charge est retombé en-dessous de la valeur maximale), on a remise à zéro du compteur par le biais de la sortie inverseuse du 74121. On en aura déduit que le fusible ne réagit pas aux dépassements d'intensité de courant de courte durée.

L'impulsion $\overline{P=Q}$ positionne (met à « 1 ») une bascule bistable qui prend la forme de IC4. Cette bascule coupe, par le biais de IC6, la tension appliquée à la charge. Le bouton-poussoir S1 permet de remettre la bascule à zéro. Les LED D1 à D3 visualisent l'état du fusible électronique : la LED jaune allumée signale que l'électronique est parée, la LED Verte reste allumée tant que le courant de charge se trouve en-deçà de l'intensité maximale ajustée, une extinction de la LED verte et un allumage de la LED rouge indiquant une coupure de la tension d'alimentation de la charge due à l'entrée en fonction du fusible électronique.

Modes d'utilisation

Il existe une double variante de mise en oeuvre : on pourra envisager de protéger une source de tension non régulée, le régulateur de tension étant dans ce cas-là monté en aval du fusible La seconde possibilité est la protection d'une source de tension régulée. Dans ce cas-là il faudra relier la sortie positive du régulateur à la broche 3 (VBB) de IC6.

(020086)



020086 - 13

Figure 3. Mesure de courant avec un LM3813/LM3815.

CI	Courant max.	Temps de réaction	CI	P1	R6
LM3813M-1.0	±1 A	50 ms	1 μF	30 à 40 kΩ	2 kΩ
LM3813M-7.0	±7 A	50 ms	1 μF	30 à 40 kΩ	2 kΩ
LM3815M-1.0	±1 A	6 ms	100 nF	40 kΩ	Pont
LM3815M-7.0	±7 A	6 ms	100 nF	40 kΩ	Pont

Reconnaissance vocale interactive

L'avenir proche ?

Ing. Franz-Peter Zantis

Bientôt, il nous semblera normal de parler aux ordinateurs. Dans cette perspective, tentons de dégager les grandes lignes de la reconnaissance vocale et de ses applications en saisie de texte, commande d'engins, mais aussi dans l'échange d'informations.

La reconnaissance vocale, certains y ont recours quotidiennement et elle leur simplifie singulièrement la vie. À quoi ressemble cette technique ? L'opération est-elle difficile, pour un ordinateur ? En réalité, il y a quantité d'obstacles à vaincre pour amener un système informatique à s'y retrouver dans le langage humain, d'autant qu'on lui demande aussi de s'assurer que le résultat obtenu soit signifiant.

En premier lieu, il faut tenir compte de la **variance** dans la manière de s'exprimer. On ne reproduit jamais le discours à l'identique. L'analyse vocale montre que chaque signal, du point de vue technique, est unique. Quand une personne prononce deux fois le même mot de la même façon, les signaux correspondants sont loin d'être superposables. À fortiori, les variations s'accroissent encore si le mot est prononcé par des personnes différentes. Selon les circonstances, les dispersions peuvent encore être poussées à l'extrême. On peut, lors d'un effort physique intense ou immédiatement après, lancer une phrase de manière toute différente de ce qu'on aurait énoncé au repos.

Le contenu du discours peut aussi être **ambigu**. Comment détecter, dans le signal audio, le moment précis où un mot commence et quand il finit ? Qui n'a pas joué à trouver

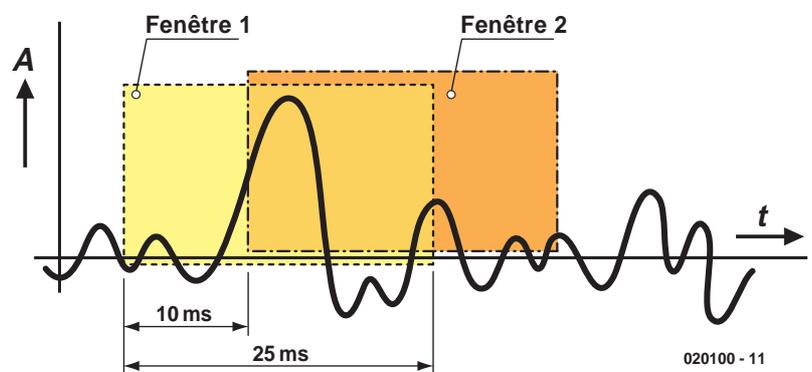


Figure 1. Le signal vocal d'entrée est sans cesse échantillonné avec chevauchement.

des graphies différentes de sons identiques, du style « Ampère sévère – en perd ses verres », « hampe, herse et vers », « en Perse, eh ! Vert » etc. De quoi donner le tournis à l'oreille électronique ou humaine la plus pointue !

Même quand on veut reconnaître une lettre dans un texte, l'affaire peut se compliquer. Suivant le mot dans lequel elle figure, sa prononciation risque d'être différente. L'ambiguïté provient autant des lettres que des mots ou des séquences de mots.

Le **canal de communication** détermine l'apparence du signal à analyser. À travers le téléphone, la voix sonne autrement que si elle passait directement, propagée par l'air ambiant lors d'une conversation en tête-à-tête.

L'**environnement** influence aussi le signal. On pense tout naturellement à l'effet dit de « cocktail party ». L'homme est capable de concentrer son attention sur le discours de son interlocuteur, même dans une

ambiance bruyante. Son cerveau fait abstraction des autres voix et du brouhaha environnant. Mais le signal qui est envoyé au système de reconnaissance vocale charrie tous ces parasites.

Autres complications de la reconnaissance vocale, l'étendue du vocabulaire et les changements de débit. Certaines personnes s'expriment posément et articulent clairement, d'autres parlent vite ou sont imprécises, sans évoquer celles qui avalent une partie de leurs mots. Raison

pour laquelle plusieurs logiciels simples du commerce tentent de s'adapter aux habitudes du locuteur. Ils demandent, avant de pouvoir en faire bon usage, de longues heures d'entraînement au cours desquelles le logiciel de reconnaissance vocale va emmagasiner une foule de détails sur la façon de s'exprimer.

Prise de son

À la base de la solution, on trouve le traitement numérique du signal. La modulation sonore est continuelle-

ment enregistrée et numérisée. Le processus se concentre sur le domaine des fréquences vocales, le reste du spectre subit un filtrage énergétique avant la numérisation.

Puisque le domaine principal de fréquence du signal vocal est confiné entre 300 Hz et 3 400 Hz, la numérisation subit peu de contraintes. Une fréquence d'échantillonnage autour de 7 kHz et une profondeur de quantification de 16 bits suffisent amplement. N'importe quelle carte sonore ou de télécommunication (analogique ou RNIS) moderne pour PC s'acquitte de la tâche sans aucune difficulté.

On analyse le signal par bloc et grâce à un chevauchement voulu des fenêtres temporelles, aucune information ne risque de se perdre (**figure 1**). La reconnaissance vocale démarre sur le signal vocal numérisé. Mais le signal d'entrée sous la forme $a = f(t)$ ne suffira pas à la reconnaissance vocale. Il faut encore faire subir au signal numérisé une transposition dans le domaine de fréquence, un filtrage et une analyse, pour aboutir à une fonction dont l'amplitude dépend du temps aussi bien que de la fréquence, sous la forme :

$$a = f(\nu, t)$$

dans laquelle :

a = amplitude du signal

ν = fréquence

t = temps

(**figures 2a à c**, avec TFR).

Pour la transformation de la fréquence, on utilise la transformée de Fourier rapide (TFR, ou FFT en anglais) ou mieux la transformée Wavelet. Cette dernière est la réponse impulsionnelle d'un filtre passe-bande (non ordonnée linéairement) dans le domaine temporel.

Méthodes

de reconnaissance vocale

On rencontre d'habitude deux manières d'aborder la reconnaissance vocale : celle basée sur les mots et celle qui étudie les phonèmes. Fondamentalement, c'est pareil. On compare le signal incident à des modèles mémorisés et l'on choisit celui qui y ressemble le plus, mot ou phonème, selon la méthode. Comme, au départ, la parole est un signal aléatoire (stochastique), nous n'avons affaire qu'à des probabilités de concordance qu'il faudra évaluer. La **figure 3** illustre le procédé de reconnaissance de mot, l'exemple est choisi dans le domaine temporel. En haut, le mot à reconnaître, en bas, trois signaux de référence. On prend celui qui y ressemble le plus.

En reconnaissance vocale basée sur les mots,

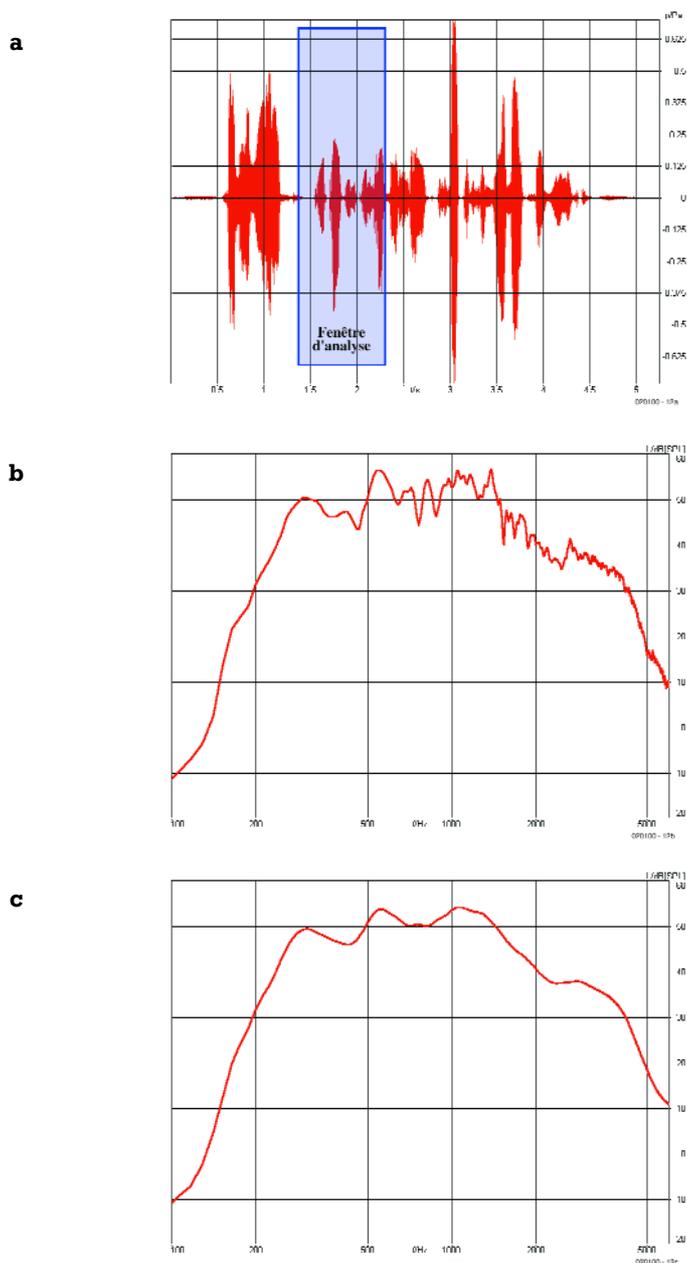


Figure 2. La phrase mise sous forme de a – signal temporel b – spectre instantané c – spectre de fréquence égalisé.

il faut constituer une banque de données reprenant tous les mots à identifier, y compris leurs transcriptions apparentées. Idéalement, on devrait y inclure la totalité du vocabulaire parlé, même dans les différents dialectes. Inutile de dire que pareille somme d'information serait bien difficile à manipuler. C'est pourquoi les systèmes existants se limitent aux mots indispensables. Et lorsqu'on veut apporter une modification au glossaire, il faut tout retranscrire.

Dans la méthode basée sur les phonèmes, on ne met plus de mots dans la banque de données de référence, mais bien des phonèmes, évidemment. Toute langue parlée peut se décomposer en phonèmes. Il s'agit de la plus petite partie constitutive du langage parlé. Chaque langue a ses phonèmes propres, lesquels sont indépendants des lettres de l'écriture. On les note sous forme d'un alphabet des sons que l'on retrouve dans les bons dictionnaires, ce sont les signes de l'Association internationale de phonétique ou API (comme alphabet phonétique international). Le monde de l'informatique, lui, par souci de mécanisation, s'en est choisi un autre, le SAMPA (speech assessment methods phonetic alphabet). La plupart des idiomes de l'Europe occidentale peuvent se ramener à une quarantaine de phonèmes.

Avantage essentiel de la reconnaissance sur les phonèmes, la possibilité de traduire en phonèmes aussi bien des mots que des groupes de mots. Voilà qui permet d'inclure les acceptions dialectales, voire carrément quelques mots étrangers, comme des termes anglais dans le dictionnaire français, moyennant une modification dans la liste des transcriptions.

Le boulot du système de reconnaissance vocale consiste donc, maintenant, à exploiter les propriétés physiques du signal :

- fréquence,
- comportement temporel du signal
- amplitude

ainsi que les propriétés sémantiques telles que :

- durée spécifique du phonème
- position particulière du phonème

pour rechercher la correspondance dans la liste de référence.

Cette recherche constitue une des questions cruciales de la reconnaissance vocale. D'abord parce que la prospection doit avancer très vite, la parole défile en temps réel. Ensuite, l'utilisateur lui demande un taux de réussite comparable à celui d'un auditeur. Pour atteindre ce résultat, il convient de

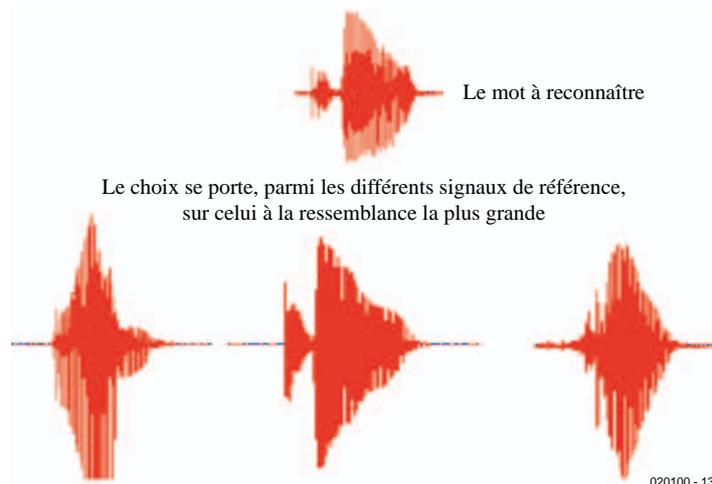


Figure 3. Le contenu de chaque fenêtre temporelle capturée est comparé aux signaux de référence disponibles.

mettre en œuvre soit des réseaux neuronaux (la logique floue ou *fuzzy*), soit un procédé statistique de grande envergure, comme ceux qui utilisent les modèles cachés de Markov (HMM, *hidden Markov models*). À la base, les HMM sont des états. On aura à évaluer aussi bien la probabilité de chaque état par rapport à la correspondance avec le signal sonore que la probabilité d'avoir

reconstitué la séquence correcte. Lors de la reconnaissance vocale, chaque état représente une partie de phonème (**figure 4**). Pour les états S_1, S_2, \dots, S_N , définis en HMM, il s'agit de parties de phonèmes (en haut) et au milieu, le calcul des probabilités de passage d'un état (S_1, S_2, \dots, S_N) au suivant. Les états peuvent, par exemple lors d'un discours lent, se répéter (effectuer des

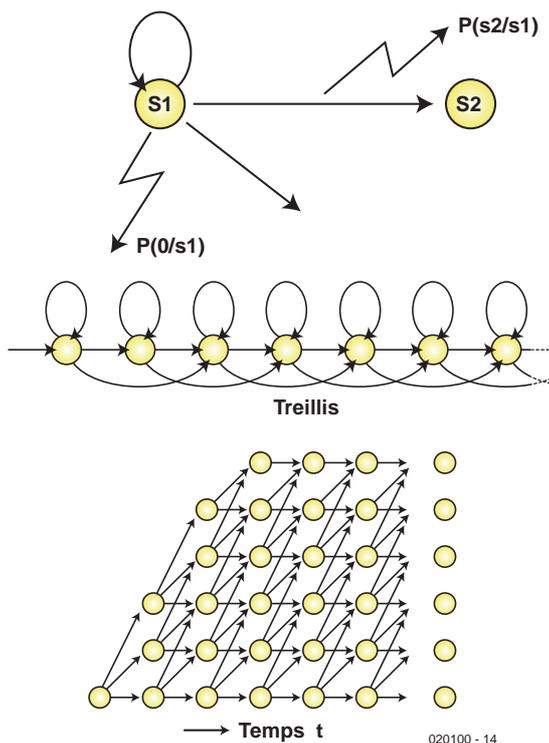


Figure 4. Le principe du Hidden Markov Model (HMM) en reconnaissance vocale.

boucles). Pour un nombre N d'états, il se crée un treillis pluridimensionnel, un enchaînement de phonèmes (en bas) qui recueille la plus haute probabilité de se trouver dans le signal de la parole.

La définition des états ainsi que les informations sur les séquences de phonèmes, le signal de référence, donc, se trouve dans un modèle acoustique (AM). Le contenu du modèle acoustique dépend de la langue à reconnaître. Le système matériel de reconnaissance vocale est quant à lui indépendant du langage, il est apte à travailler dans n'importe lequel, pour autant qu'on lui fournisse la référence acoustique

correspondante.

Arrivés à ce stade-ci, nous ne sommes toujours nulle part. Les phonèmes, c'est bien, mais ce que nous voulons, ce sont les mots. La **figure 7** nous montre une représentation schématique de l'appareil de reconnaissance vocale. Le modèle acoustique recèle une suite de phonèmes. Il nous faut encore un lexique dans lequel tous les mots à reconnaître, y compris les transcriptions apparentées, sont répertoriés en alphabet phonétique, ou plutôt en code SAMPA. Le lexique détermine donc le vocabulaire du système. Seuls peuvent être reconnus les mots contenus dans le lexique. Mainte-

nant que nous avons une suite reconnue de phonèmes, nous pouvons lancer une recherche dans le lexique pour y trouver une transcription comparable. S'il s'y trouve une correspondance exacte, nous pouvons sortir le mot en question. La reconnaissance vocale est un processus ininterrompu, aussi les mots identifiés vont-ils se succéder dans ce que l'on appelle un graphe sémantique, lequel reste vide aussi longtemps qu'une correspondance n'aura pas été repérée entre la séquence de phonèmes et une entrée du lexique. Au départ, le modèle de langue représenté dans la figure est facultatif. Il résulte d'une phase d'entraînement et contient des informations sur des mots et des suites de mots typiques de l'application, dans le but d'améliorer le résultat. Le graphe sémantique produit contient deux pistes :

- « rouler en auto »
- « voyager avec l'auteur »

(rouler en voiture ou voyager avec l'auteur). Les mots « Auto » et « Auteur » se ressemblent fort, à l'audition, et présentent des spectres de fréquences très similaires. Bien que le système, lors de la reconnaissance vocale, ait attribué au mot « Auteur » une plus haute probabilité, le modèle de langue, sur base d'informations statistiques dans les enchaînements logiques, accorde la préférence au mot « Auto ».

Le travail du système de reconnaissance vocale peut aussi s'exprimer sous forme mathématique. Recherchons la probabilité maximale de reconnaissance d'un mot w dans une liste W à partir d'un signal acoustique x :

$$W_{spoken} = \arg \max_{w \text{ from } W} P(w | x)$$

Si P(x) est le signal acoustique entrant, P(w) la probabilité de concordance d'une séquence de mots (dans le cadre du modèle de langue) et P(x|w) la probabilité d'atteindre la correspondance entre l'information acoustique et la séquence de mots w (contenue en principe dans le modèle acoustique), alors on peut évaluer comme suit la probabilité de reconnaissance d'une séquence de mots w dans la liste W :

$$P(w | x) = \frac{P(w | x) \cdot P(w)}{P(x)}$$

P(x) est déterminée par la fréquence d'échantillonnage, la résolution (8 bits ou 16 bits) et le traitement ultérieur du signal, elle est donc constante.

La séquence de mots reconnue, donc le

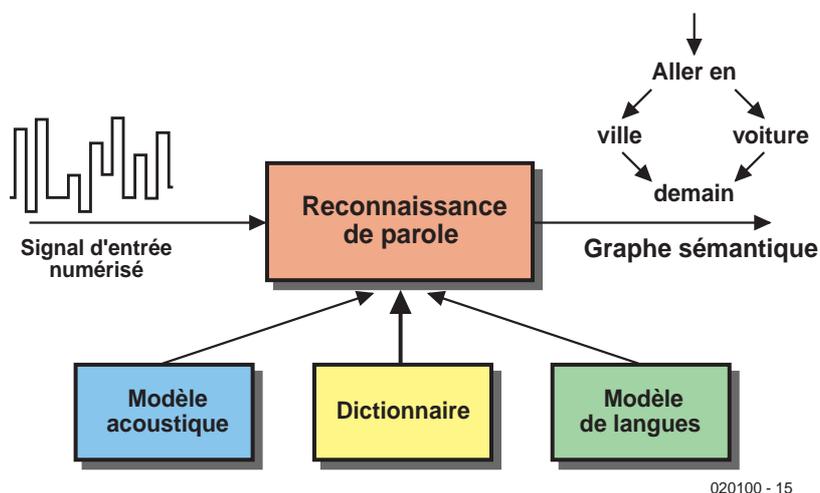


Figure 5. Schéma d'un appareil de reconnaissance vocale.

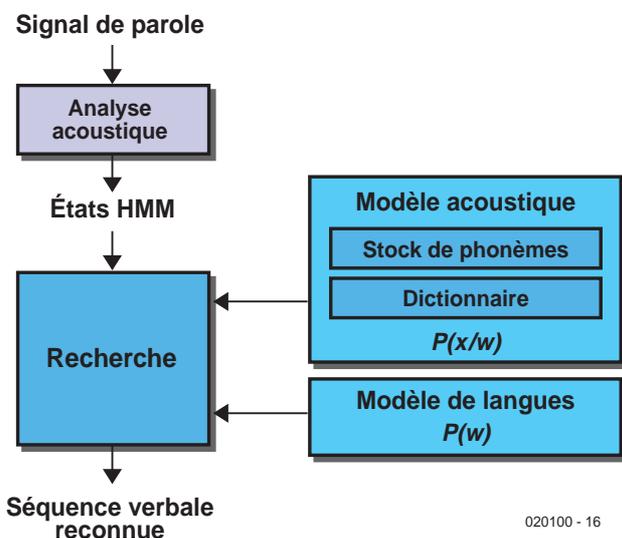


Figure 6. Mission essentielle de la reconnaissance vocale, identifier les phonèmes prononcés pour reconstituer le mot correspondant.

graphe sémantique, peut à présent être transmise directement au convertisseur de la parole au texte (**figure 6**). Des applications typiques de cette méthode sont les traitements de texte à saisie vocale. Mais dans un système interactif, qui fonctionne en questions et réponses, il faut en outre analyser le sens des phrases énoncées. Le travail n'est donc pas encore terminé.

La fabrication du modèle acoustique n'est pas une mince affaire (**figure 7**), elle nécessite la compilation de quantités énormes de données sous forme de signaux de parole. Toute cette matière constitue un corpus, un ensemble linguistique cohérent composé de données sonores ainsi que les listes de mots bien compris et corrigés. Le lexique englobe, dans ce cas-ci, l'intégralité du dictionnaire de la langue et plus seulement le vocabulaire nécessaire à traiter d'une spécialité donnée. Constituer la somme des modèles acoustiques correspondants, même sur des ordinateurs modernes, prend de deux à trois semaines de temps de calcul ininterrompu.

Maîtriser la langue

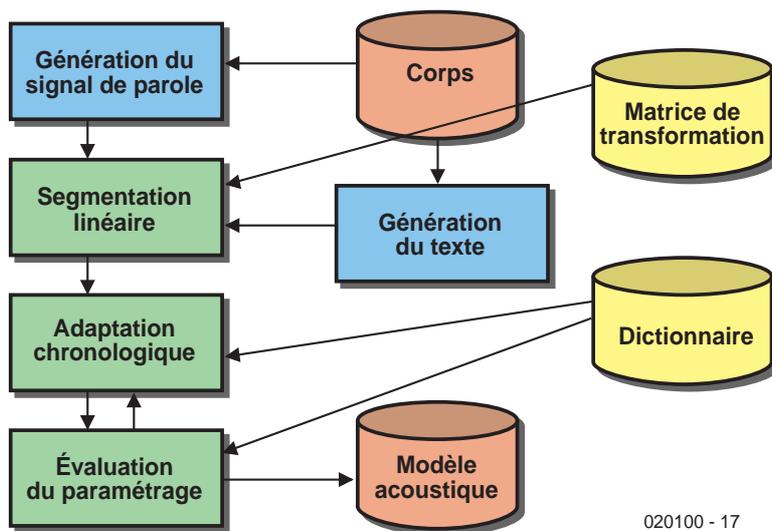
L'idée qui sous-tend la compréhension de la langue n'est pas de vouloir percer les finesses de la totalité de la phrase ou du signal, mais seulement d'y discerner les informations utiles. Quelles sont les données à considérer, cela dépend beaucoup du contexte. Comprendre les paroles prononcées a pour objectif de traduire le graphe sémantique en un mot ou en une suite de mots pertinents.

Une méthode pour y parvenir consiste en une analyse grammaticale statistiquement orientée, sans s'occuper du contexte.

Le module de compréhension de la parole recherche dans les graphes sémantiques des notions probables, basés sur des mots individuels ou des groupes de mots. Chaque notion découverte sera traitée comme une information utile.

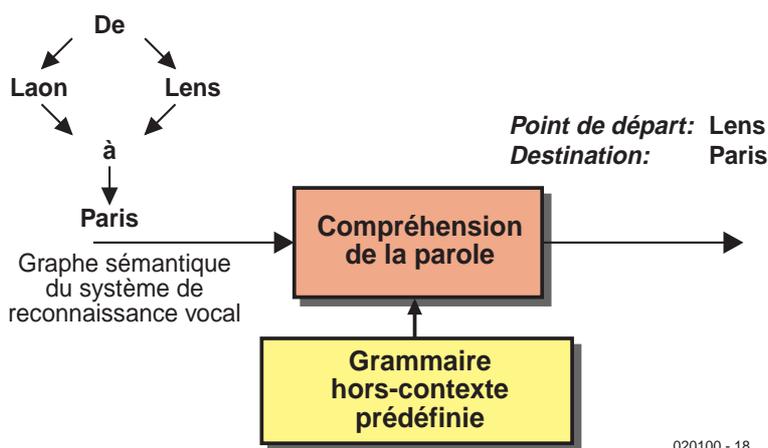
Supposons qu'il s'agisse de répondre à une demande de renseignement sur les horaires. Il y sera forcément question de *lieu de départ*, de *destination*, de *date* et d'*heure d'embarquement*. On pourrait par exemple y trouver « de Lens » ou « à Paris » ou alors de manière plus générale « de [nom de ville] » « à [nom de ville] ». L'information de *lieu de départ* est identifiée par le mot « de », tandis que le mot « à » introduit la *destination*.

Dès que l'on découvre une séquence d'information, on la place dans une variable. Si par exemple on trouve la séquence « à Paris », le programme pourra introduire dans la variable *Destination*, sous forme de chaîne de caractères (*string*), le contenu « Paris » : *Destination* = « Paris ».



020100 - 17

Figure 7. Schéma d'un modèle acoustique destiné à servir de référence lors de la reconnaissance de la parole.

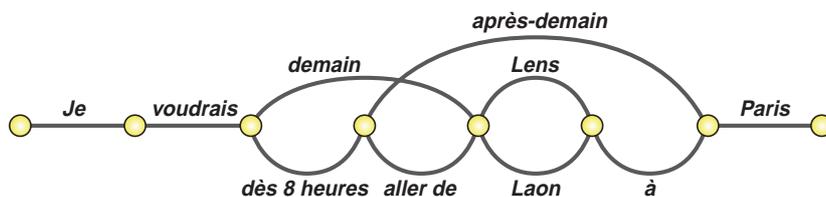


020100 - 18

Figure 8. L'unité fonctionnelle de compréhension de la parole soumet le graphe sémantique du mot reconnu à une analyse grammaticale adaptative.

En finale de la phase de compréhension de la langue, le contenu des variables est disponible et signifi-

tif. L'exemple de la **figure 8** montre que les variables « *Point de départ* » et « *Destination* » ont été complétées.



020100 - 19

Figure 9. Un graphe sémantique peut recouvrir de nombreux mots différents qui ne sont pas nécessairement pertinents ni ceux qu'il fallait trouver.

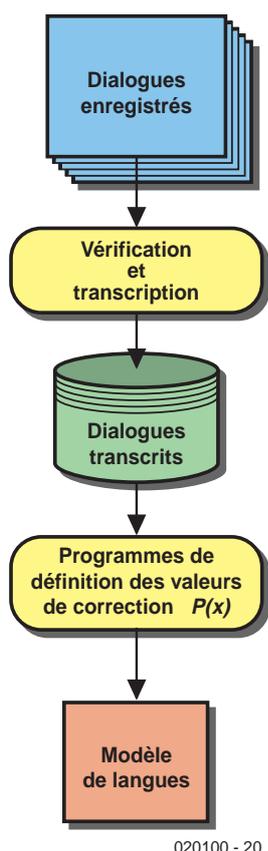


Figure 10. Pour produire le modèle de langue, on se sert de dialogues enregistrés.

Les valeurs doivent encore être traduites en code logiciel pour servir de base au traitement ultérieur. On peut de la sorte, à l'aide de *If-Else* ou d'une instruction *Case*, déterminer la valeur contenue dans chaque variable et, en fonction de cela, décider de ce qu'il faut en faire. On peut par exemple vérifier l'information en posant une question :

```

If Point_de_départ <> 0 and
Destination <> 0
  « Vous souhaitez aller de »
  & Point_de_départ & « à »
  & Destination & « Est-ce
  exact ? »
End if
  
```

Conduite du dialogue

L'exemple de la figure 9 présente un graphe sémantique un peu plus complexe. Au cours de la phase de compréhension de la langue, le sys-

tème peut, dans ce cas-ci, mémoriser dans la variable le nom « *Lens* ». Mais s'il existe un doute à propos du lieu de départ, on peut le remplacer. Il y a deux manières de s'y prendre. On peut s'abstenir de compléter le champ « *Point de départ* » et s'assurer que le renseignement est correct avant de le stocker dans la variable, en posant une question orale ciblée : « *Pardon, je n'ai pas compris votre point de départ* ». L'alternative, c'est de se servir du mot qui offre la plus haute probabilité pour poser une question judicieuse susceptible de vérifier le nom de ville :

– *Est-ce bien de « Laon » que vous voulez partir ?*

Dans le scénario adopté, nous pourrions assister à un dialogue tel que :

Appelant : – Je voudrais aller de *Lens* à *Paris*

Système : – Est-ce bien de *Laon* à *Paris* que vous voulez vous rendre ?

A : – Non, pas de *Laon*, mais de *Lens*

S : – Et quand voulez-vous aller de *Lens* à *Paris* ?

A : – lundi prochain

... et ainsi de suite.

Synthèse de parole

Indispensable à la conduite du dialogue, c'est le synthétiseur de voix, qui de son côté a recours à un texte (écrit), phrases et mots à prononcer. Pour commencer, on peut utiliser un convertisseur de texte en paroles. Le procédé est particulièrement simple, mais cette voix de robot, produite artificiellement, manque toujours de naturel. Une variante consiste à enregistrer une voix humaine qui prononce toutes les phrases et à s'en servir comme fichiers sonores, classés en répertoire. Le système y recherchera les expressions nécessaires et les rejouera dans l'ordre voulu. Les questions et réponses auront l'air d'autant plus naturelles que l'on s'efforcera d'enchaîner des propositions plutôt que des mots isolés.

La réponse du système à la demande de la figure 9 pourrait provenir d'un seul fichier dont le contenu serait le suivant :

– *Est-ce bien de Laon à Paris que vous voulez vous rendre ?*

ou de six fichiers sonores assemblés de cette manière :

– « *Est-ce bien* » | « *de* » | « *Laon* » | « *à* » | « *Paris* » | « *que vous voulez vous rendre ?* »

La segmentation des phrases ne se fait évidemment pas au petit bonheur. Les noms de villes émanent d'une banque de données, ils constituent des propositions séparées, de même que l'annonce introductive « *Est-ce bien* ». Le désavantage de cette procédure se manifeste dès qu'on doit apporter des modifications au répertoire. Il faut définir chaque nouvelle phrase, la prononcer, l'enregistrer et revoir le classement des fichiers sonores.

La pratique

Si l'on veut de bons résultats, un tel système de dialogue automatisé doit subir au préalable une étape expérimentale, au cours de laquelle on enregistrera des dialogues représentatifs pour les analyser ensuite. Elle doit viser à adapter le modèle de langue, le lexique et éventuellement le modèle acoustique (figure 10). Lorsque le système est au point, on atteint des taux de reconnaissance supérieurs à 90 %, ce qui correspond à la perception auditive humaine.

Des systèmes tels que décrits ici fonctionnent déjà depuis quelques années. La puissance de calcul requise du système est énorme et jusqu'à récemment, seul un « workstation » spécial, un ordinateur intermédiaire entre le macro-ordinateur et le PC, arrivait à suivre le rythme. Mais aujourd'hui, un bon PC du commerce y parvient.

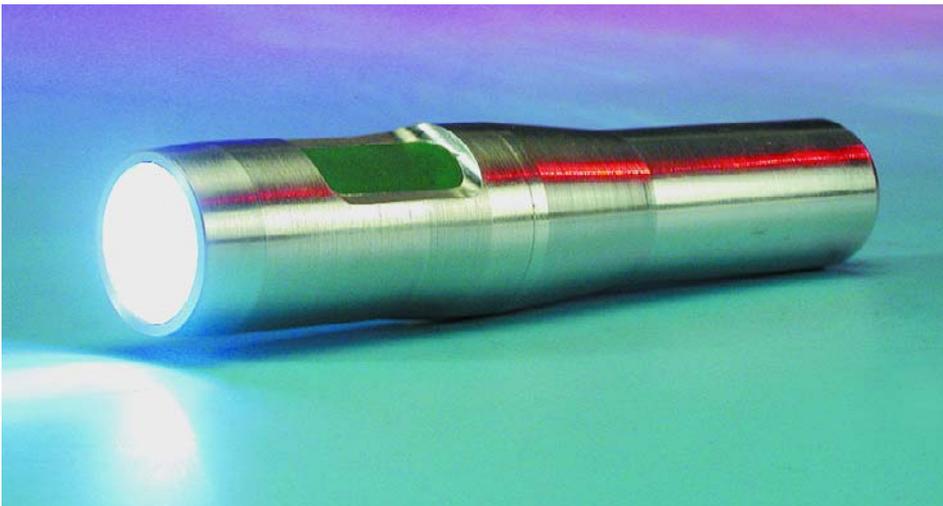
Il existe de nombreux domaines d'application pour des systèmes de reconnaissance de la parole comme celui que nous venons de décrire. À côté des appareils de dialogue, on peut aussi penser à la commande vocale. Dans les cas les plus simples, on peut envisager un asservissement à la voix sans réponse, à la maison. On installe un microphone dans la pièce et le système de reconnaissance vocale écoute en permanence. On peut alors lui donner des ordres tels que « *allume le lustre* ». S'il peut reconnaître les phrases sans se tromper, on peut alors lui confier la commande d'un bus de domotique. Et, même si l'idée est assez futuriste, ce n'est pas de la science fiction que d'imaginer, en cas de difficulté de compréhension, qu'un haut-parleur pose la question : « *Voulez-vous que j'allume le lustre ?* »

(020100-1)

Torche à LED

Modèle High-Tech à microcontrôleur

Dans le numéro double du tournant du siècle, nous vous avons déjà proposé notre première lampe de poche à LED qui « brillait » surtout par sa simplicité. Si le modèle que nous vous présentons ici ne comporte guère qu'un ou deux composants supplémentaires, il peut se vanter de comporter un microcontrôleur (!) ce qui lui donne des fonctionnalités intéressantes.



Une lampe de poche à LED se doit d'être aussi petite et légère que possible pour être mise facilement dans un sac à main, la poche ou un sac à dos sans causer la moindre gêne. Pour des raisons de respect de l'environnement et de coût, elle se doit de comporter un accumulateur rechargeable qui doit pouvoir fonctionner même si l'on a oublié de recharger la lampe de poche depuis de nombreux mois, voire années. Elle devrait indiquer l'état de charge momentané de l'accumulateur et être dotée d'un dispositif de protection au cas où on aurait oublié de la couper. Ajoutons à ce cahier des charges qu'elle devrait éclairer dans un rayon de plusieurs mètres de manière à pouvoir servir, par exemple, d'éclairage d'appoint ou de lampe de chevet pendant plusieurs heures. À luminosité réduite,

elle devrait pouvoir tenir plusieurs nuits sans interruption et même pouvoir, alors que son accumulateur est pratiquement vide, si on lui laisse le temps de récupérer un peu, avec le reste du dernier pour cent de sa capacité (10 mA) pouvoir servir, pendant quelques heures encore, d'éclairage de secours utilisable. Vu l'énormité de ce cahier des charges, nous avons opté pour des LED blanches plutôt que pour une ampoule halogène. Contrairement à une opinion répandue, le rendement des LED de couleur blanche n'est pas sensiblement plus élevé que celui des ampoules de faible puissance (1 W) voire même inférieur à

celui d'ampoules à incandescence de 3 W et plus. Il n'en reste pas moins que les LED blanches présentent nombre d'avantages par rapport aux ampoules : il est possible d'abaisser les LED au millième de leur puissance maximale sans nuire à leur rendement alors que dans le cas d'une ampoule, son rendement s'effondre à zéro quasiment en deçà de la moitié de sa puissance nominale. La durée de vie d'une LED est mille fois supérieure à celle d'une ampoule, ce qui évite d'avoir à prévoir d'ampoule de rechange. L'optique intégrée de la LED permet de donner à la lampe de poche des dimensions sensiblement plus compactes que ne le permettrait l'utilisation d'une ampoule avec réflecteur.

En ce qui concerne l'accumulateur, nous avons opté pour un accu Li-Ion, qui bien que sensiblement plus cher qu'un accu CdNi, mais qui pour la même énergie ne pèse que le tiers du poids de ce dernier, se caractérise par une auto-décharge de quelque 10% par an seulement (comparés aux 20% par mois (!) dans le cas d'un CdNi et qui ne connaît pas d'effet de mémoire.

Le concept du circuit

Vu que la tension d'accu d'un accu Li-Ion pleinement chargé est de 4,1

Tableau 2. Niveaux de luminosité

Niveau	Puissance	Consommation à $U_{pile} = 3,6 V$	Durée de vie de la pile (1,3 mAh)	Mise hors-fonction auto après
0	0,5 W	140 mA	10 h	20 min
1	0,2 W	55 mA	24 h	40 min
2	65 mW	18 mA	3 d	2 h
3	20 mW	6 mA	9 d	5 h
4	8 mW	2,4 mA	22 d	10 h
5	4 mW	1,0 mA	55 d	10 h
Stand-by	-	30 μ A	5 a	24 h
Éteint	-	2 μ A	70 a	-

– En mode *stand-by* la lampe de poche clignote le plus économiquement possible, permettant, par exemple, de la retrouver dans une tente obscure. Avant que la lampe de poche ne passe en mode *stand-by*, le patron du clignotement indique la charge approximative de l'accu. Une impulsion de clignotement correspond à 10% de la pleine charge, un état de charge inférieur à 10% étant signalé par de courtes interruptions qui représentent chacune 1% de charge.

Le **tableau 1** récapitule les différentes fonctions que les 2 boutons-poussoirs permettent de choisir. Le **tableau 2** donne des informations chiffrées quant aux différents niveaux de luminosité.

Réglage de la puissance

La tâche première du convertisseur rehausseur est, nous le disons plus haut, de fournir aux LED une puissance constante en dépit des variations de la tension d'alimentation. L'énergie *E* stockée dans la self *L* par le passage d'un courant *I* répondant à la formule suivante :

$$E = \frac{1}{2} \cdot L \cdot I^2$$

Après mise en fonction pendant une durée t_{ON} du commutateur T1 (voire hors-fonction t_{OFF}) le courant est passé, linéairement, de 0 à une valeur *I* :

$$I = \frac{U_{pile} \cdot t_{on}}{L}$$

On dérive de *E* et de la puissance moyennée sur la dite durée *P*, la formule suivante :

$$E = \frac{1}{2} \cdot \frac{U_{pile}^2 \cdot t_{on}^2}{L}$$

$$P = \frac{1}{2} \cdot \frac{U_{pile}^2 \cdot t_{on}^2}{L \cdot (t_{on} + t_{off})}$$

La variation de *P* est quadratique par rapport à U_{pile} , en l'absence de régulation *P* varierait d'un facteur 2 entre les valeurs de tension extrêmes de 2,8 et de 4,1 V. Le courant moyen vaut P/U_{pile} . On peut maintenant, en fonction de U_{pile} , jouer sur les temps t_{ON} et t_{OFF} de manière à garder constante la puissance *P*.

Pour pouvoir garantir une régulation de puissance efficace et correcte, le microcontrôleur doit être informé en permanence de la tension aux bornes de l'accu, U_{pile} . La mesure de cette tension n'est pas faite directement pour éviter de compliquer inutilement l'électronique requise. Nous avons mis la ligne AN0 à contribution, le CAN utilisant la tension $V_{DD} = U_{pile}$ en tant que référence (= pleine échelle). La tension de la référence de barrière de potentiel (*bandgap*) TL431AC est de 2,495 V. Le résultat de mesure ADRES du

CAN 8 bit est partant le suivant :

$$AD_{RES} = \frac{U_{pile} - 2,495V}{U_{pile}} \cdot 255$$

ou en inversant

$$U_{pile} = \frac{2,495V}{1 - AD_{RES} / 255}$$

La mesure de U_{pile} et la régulation se font une fois toutes les 100 ms.

Disposant de la valeur de la tension aux bornes de l'accu, le logiciel recherche dans un tableau (comportant 16 plages de valeurs) une valeur convenable pour t_{ON} et t_{OFF} et pilote en conséquence la sortie GP4.

La tension inverse dans *L* étant de l'ordre de $6 \cdot U_{pile}$, le courant *I* est, lors de chaque période, mis à zéro au cours de la durée t_{OFF} , si tant est que l'on ait veillé à ce que la relation $t_{OFF} > 1/6 \cdot t_{ON}$ soit vraie.

Le rendement du circuit dépend principalement des pertes au niveau de la diode Schottky et des pertes ohmiques dans la self, le FETMOS ainsi que de la résistance interne de l'accumulateur et des interconnexions. À la puissance maximale, le rendement atteint de l'ordre de 94%. On notera cependant que le rendement des LED, même si elles proviennent d'un même fabricant, peut varier fortement. Les LED blanches ayant actuellement le meilleur rendement sont fabriquées par Nichia. Bien qu'elles ne soient fabriquées que pour des utilisateurs

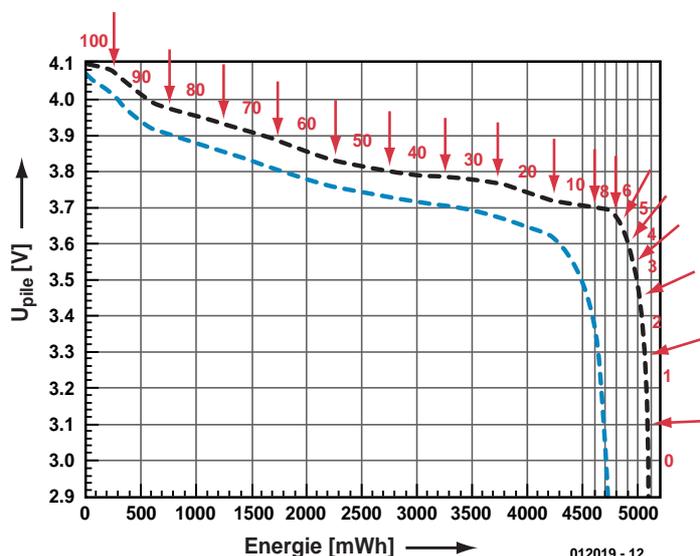


Figure 2. Courbe de décharge typique d'une cellule Li-Ion

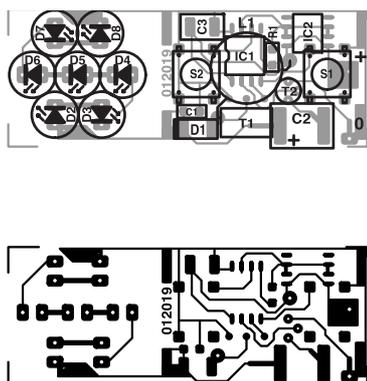
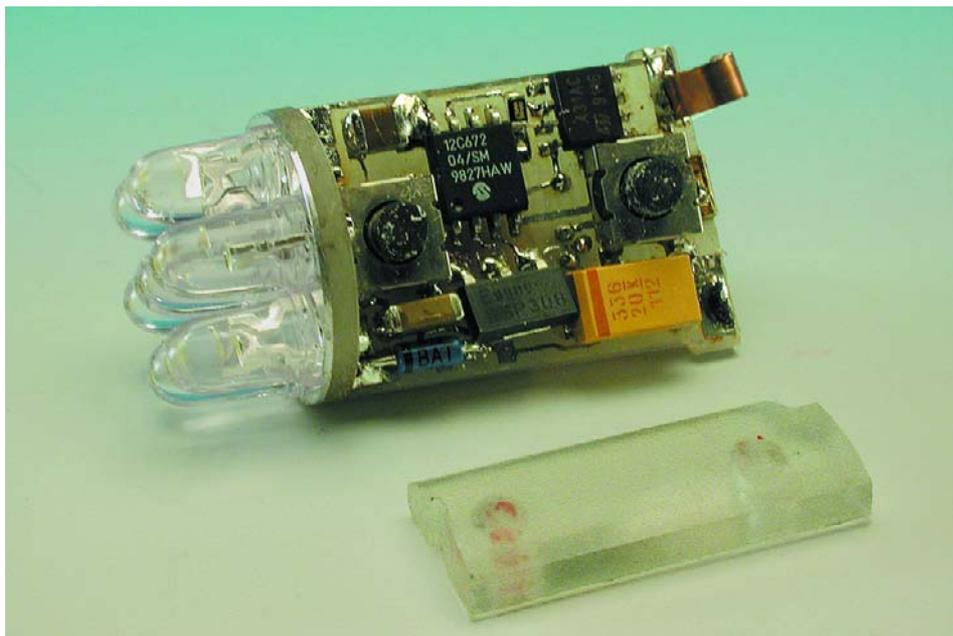


Figure 3. Dessin de la platine en 2 parties destinée à recevoir les composants CMS.



Liste des composants

Résistances :

R1 = 680 Ω CMS 0805

Condensateurs :

C1 = 220 nF céramique CMS 0805

C2 = 33 μF/20 V CMS 2220

C3 = 1 μF/6V3 CMS 1210

Bobines :

L1 = 233 μH (100 kHz) ESR = 0,27 Ω

pot ferrite P9.0x5.0/N26 (Al = 250) doté de 30,5 spires de fil de cuivre émaillé de 0,22 mm de diamètre

Semi-conducteurs :

D1 = BAT48 ou 1N4148

D2 à D8 = LED 5 mm blanche, 6400 mCd telle que NSPW500BS (Nichia) (www.nichia.co.jp)

T1 = BSP308, BSP319

($R_{ON} = 50 \text{ m}\Omega$)

T2 = BSI170, BSS138 ($R_O = 5 \Omega$)

IC1 = PIC12C672 04/SM

(programmé **EPS012019-41**)

IC2 = LM9140-2.5 ou TL431

Divers :

S1, S2 = interrupteur unipolaire

CMS tel que, par exemple,

(Mentor 1254.1007 ou

1301.9314 ou Omron B3FS-1052 chez Farnell)

pile = Sanyo UR18650 Li-Ion, 1350 mAh (diam. 18 mm, long. 65 mm, poids 40 g) ou Sanyo

UR18500 Li-Ion, 1100 mAh

(18 mm, long. 50 mm)

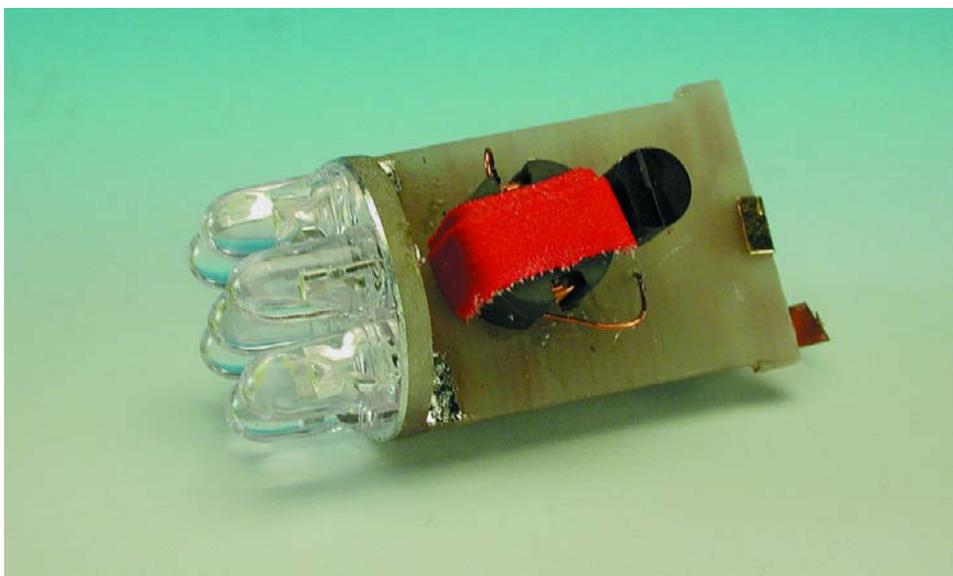
OEM (*Original Equipment Manufacturer*), on en trouve quand même dans le commerce.

L'état de charge

Pour éviter que la lampe de poche à LED ne laisse son propriétaire dans le noir en raison de l'épuisement de l'accu, le microcontrôleur doit connaître en permanence l'état de charge de l'accu. Comme la tension de décharge dépend non seulement du courant de décharge, mais aussi de l'historique des décharges précédentes et de la température, nous éliminons ces facteurs en faisant en sorte que le contrôleur ne détermine

l'état de charge que hors-circulation de courant, et ce longtemps après (1 000 s) après la coupure de la lampe de poche.

La **figure 2** montre la caractéristique de décharge d'un accu Li-Ion. Une fois la lampe de poche coupée, la tension aux bornes passe, en 1 000 s, de la courbe du bas vers la courbe du haut. Les flèches et les valeurs identifient l'état de charge exprimé en pour cent tel que l'affiche le microcontrôleur. L'erreur de mesure est de l'ordre de $\pm 30 \text{ mV}$; déterminer fiablement l'état de charge dans les derniers 4% est quasiment impossible. En tout état de cause, même si l'erreur lors de la détermination de l'état de charge peut sembler quelque peu importante, il est parfaitement possible de faire la différence entre un niveau de charge de 80, 50 ou 20%. L'autodé-



charge est négligeable (inférieure à 5%/a), de sorte que l'accu pourra être utilisé, sans être rechargé, plusieurs années plus tard.

Un dispositif de *timeout* évite une décharge de l'accu au cas où l'on aurait oublié, par inadvertance, d'éteindre la lampe de poche. La coupure se fait (indépendamment de la puissance) après une consommation de l'ordre de 3% de la capacité maximale de l'accu qui est de 5 Wh.

Lorsque la charge de l'accu tombe en deçà de 10%, il est pris un certain nombre de mesures d'économie de courant destinées à éviter une décharge complète de l'accu. Selon le niveau de luminosité du moment, on passe, assez rapidement, à une puissance moindre. Normalement, suite à cette réduction de la demande, la tension de décharge remonte quelque peu. Si elle continue cependant de diminuer alors que l'accu se trouve dans un état de décharge avancé, le processeur éteint la lampe de poche purement et simplement. Il n'en reste pas moins possible de rallumer la lampe de poche pendant une durée relativement courte et à une puissance faible. S'il ne reste plus qu'1% de charge, la lampe de poche pourra éclairer 5 heures encore au niveau de luminosité 4, voire plus de 12 heures au niveau 5, à luminosité plus faible. Il ne devrait partant vous arriver de vous retrouver complètement dans le noir...

Accumulateur et self

L'accu que nous avons utilisé possède en fait, avec ses 1,3 Ah (près de 5 Wh), une taille trop importante pour la lampe de poche. Cependant, il faut noter qu'il constitue actuellement la taille standard au prix le plus inté-

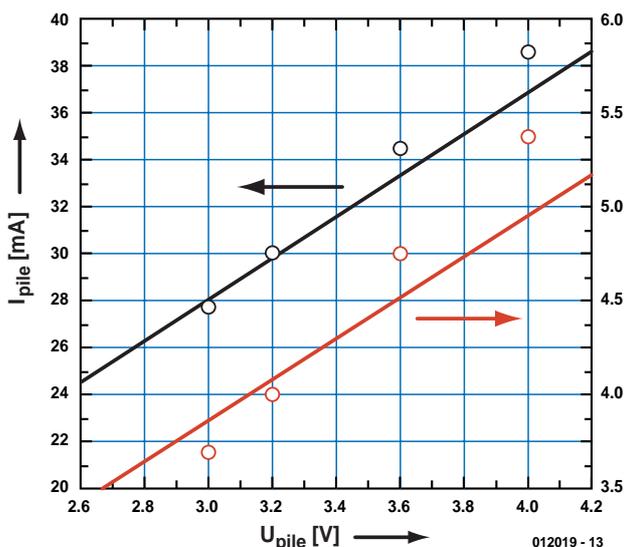


Figure 4. Consommation de courant de la lampe de poche à LED.

ressant sachant qu'on le retrouve dans les ordinateurs portables. Les accumulateurs pour téléphones portables sont plus difficiles à trouver sans même parler du fait qu'ils coûtent plus cher.

Si le sujet vous intéresse, vous pourrez trouver des fiches de caractéristiques consacrées aux accus Li-Ion sur Internet à l'adresse :

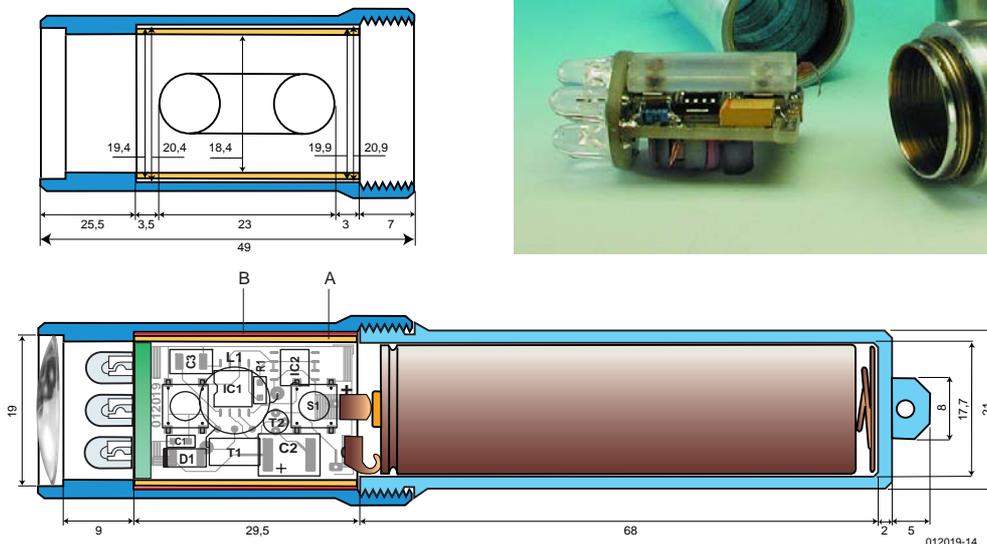


Figure 5. Croquis coté pour la réalisation d'un boîtier étanche pour la lampe de poche.

www.sanyo.com/industrial/batteries/industrial_liion.html

ou

www.panasonic.com/industrial/battery/oem/chem/lithion/index.html

On peut trouver la self dont l'inductivité sera comprise entre 220 et 250 μH et ayant une I_{MAX} égale à 0,5 A, toute faite si tant est que l'on connaisse une source d'approvisionnement (et que l'on y ait une entrée). Si l'on veut faire sa self soi-même il faudra utiliser du fil de cuivre émaillé de 0,2 à 0,22 mm de diamètre que l'on bobine en couches de 10 spires sur un noyau EPCOS ayant une valeur Al de 250. De manière à minimiser la capacité de la self et obtenir un bobinage bien régulier d'une couche à l'autre, intercaler 2 tours de film plastique de 0,1 mm d'épaisseur. Il suffira, en principe, de faire 30,5 spires de manière à obtenir l'inductance recherchée, à savoir :

$$L = 30,5^2 \cdot 250 \text{ nH} = 233 \text{ mH.}$$

Il est préférable de prévoir une spire supplémentaire (qui accroît L de 6%), de mesurer le résultat voire l'essayer expérimentalement comme nous le décrivons dans le prochain paragraphe. Il ne sera pas difficile ensuite, si nécessaire, d'enlever 1 ou 2 spires. Une petite remarque au passage. Sachez, s'il vous venait à l'idée de fixer la lampe de poche sur une surface métallique à l'aide d'un aimant, qu'il faudra placer le dit aimant à au moins 2 cm de la self sous peine de voir chuter sinon l'inductivité à moins de 10% de sa valeur.

Réalisation

Nous en savons maintenant assez sur le fonctionnement. Venons-en à l'aspect pratique ! Nous allons commencer par découper en 2 parties la platine représenté en **figure 3** : la partie LED et la partie du reste des composants qui sont tous, exception faite de la self, de type CMS (montage en surface). Le seul composant « normal » de la platine du processeur, la self, vient, cela va de soi, s'implanter côté « composants ». Une fois les composants montés, les 2 platines sont soudées à l'équerre de manière à ce que les

LED puissent recevoir leur tension d'alimentation.

Il faudra disposer, pour le premier test de bon fonctionnement, d'une alimentation de laboratoire (avec ampèremètre ou en combinaison avec un ampèremètre à faible résistance interne si l'alimentation ne comporte pas d'affichage de courant); on la branchera au montage. Pour être réinitialisé correctement, le PIC a besoin d'une raideur de pente de croissance de la tension d'alimentation minimale; il ne saurait partant être question « d'ouvrir » progressivement la tension; il faudra, à une tension de 2,8 V, connecter l'alimentation au montage, vu que sinon le FET peut entrer en conduction et drainer une intensité de courant importante. Il est recommandé, pour éviter toute mauvaise surprise lors du premier test, d'intercaler une résistance de limitation de courant de 10 Ω (limitation du courant à 0,4 A) dans la ligne d'alimentation.

Après la réinitialisation, le PIC se lance, sans qu'il ne soit nécessaire d'appuyer sur l'une des 2 touches, dans l'exécution de la première de 2 routines de test. Ces routines de test ne sont exécutées qu'une seule et unique fois, lors de la connexion de l'accu au système (mais bien évidemment également après chaque déconnexion de l'accu !).

On s'assurera de la présence d'un signal d'horloge d'une fréquence de 4 MHz (ce qui correspond à une durée de cycle de 1 μs) avant de vérifier l'inductance de L1 et le fonctionnement du CAN.

Au cours de la première routine de test la lampe de poche clignote à une période de 200 ms et un rapport cyclique de 0,5; ce clignotement dure 1 000 périodes (200 s) pour cesser ensuite si aucune des touches n'a été actionnée. Si les durées varient fortement par rapport aux valeurs données (en particulier si t_{ON} dure plus de 20% plus longtemps, encore qu'il y ait peu de raison que cela soit le cas) cela signifie que les LED sont traversées par un courant beaucoup trop important. Il faudra dans ce cas-là réduire le nombre de spires de la self d'1 ou de 2. Dans la première routine de test, les durées de t_{ON} et $(t_{\text{ON}}+t_{\text{OFF}})$ ont été fixées à 10 et 100 μs respectivement. On peut déduire la consom-

mation de courant à l'aide de la courbe inférieure de la **figure 4**.

Si tout se passe comme prévu, on appuiera brièvement sur l'une des touches. La lampe de poche s'éteint et le programme passe à la seconde routine de test qui prend la forme d'une mise en fonction continue avec un t_{ON} de 20 μs et un $(t_{\text{ON}}+t_{\text{OFF}})$ de 100 μs .

Un doublement de t_{ON} se traduit par un quadruplement de la puissance, la lampe de poche restant en fonctionnement continu. Si on compare les 2 routines de test, la puissance a été multipliée par 8. La consommation de courant devrait correspondre à la courbe supérieure de la figure 4.

En l'absence d'action sur une touche, la lampe de poche s'éteint au bout de 50 s. On évite de cette manière une mise en fonction ininterrompue après une réinitialisation du processeur lorsque, par exemple, à la suite d'une chute, l'accu devait être déconnecté, un court instant, du reste du circuit. Une action rapide sur l'une des touches se traduit par l'allumage de la lampe en mode (presque) normal puisqu'on se trouve au niveau de luminosité 2 (c'est-à-dire le 3^{ème} niveau de puissance).

Travaux pratiques

Il n'est malheureusement pas possible de démonter une mini-lampe de poche du commerce et en utiliser le boîtier pour y installer notre électronique, pour la simple et bonne raison qu'un accu Li-Ion possède une section sensiblement supérieure à celle d'une cellule R6 (mignon).

Si donc vous voulez disposer d'un boîtier compact et attrayant et que vous ne disposez pas d'un tour et d'une fraiseuse, il vous faudra faire appel à l'aide d'un atelier disposant de ces machines.

Le croquis coté de la **figure 5**, disponible sous format Micrographx sur le site Internet d'Elektor, donne toutes les cotes des 2 parties qui constituent le boîtier et que l'on vissera l'une sur l'autre après avoir glissé la platine dans la partie supérieure et l'accu dans la partie inférieure.

On notera la forme légèrement conique du fraisage de la partie dans laquelle se glisse l'électronique. Cela permet d'assurer une bonne fixation du tube de cuivre et du joint de caoutchouc qui assurent l'étanchéité des 2 touches.

La platine sera dotée de 2 contacts de laiton chargés d'assurer un bon contact avec la cellule Li-Ion. Un ressort placé dans le fond de la partie inférieure du boîtier en dessous de la cellule génère la pression de contact nécessaire et suffisante.

(012019)

Digital Radio Mondiale

Radio numérique en GO, OM, OC

Hans Weber

Les ondes courtes aussi passent au numérique ! Dès cette année commence une enquête de terrain auprès des auditeurs des ondes courtes et des radioamateurs du monde entier pour évaluer le nouveau procédé de transmission. Une présérie de récepteurs a été présentée à l'IBC d'Amsterdam. Le lancement officiel est prévu en juin 2003.



D'année en année, les gammes de radiodiffusion inférieures à 30 MHz, les ondes longues, moyennes et courtes, tombent davantage en désuétude, principalement à cause de la médiocre qualité sonore qu'elles offrent, et cela du fait de l'exiguïté des canaux de radiodiffusion dans lesquels les programmes doivent passer : 9 kHz dans les gammes de fréquences basses, 10 kHz en ondes courtes. En outre, la modulation d'amplitude, qui y est de règle depuis les balbutiements de la T.S.F., si elle a le mérite de la simplicité technique, pêche par manque d'efficacité. À partir de procédés de modulation plus complexes, d'un traitement de données et d'une compression plus moderne, il est aujourd'hui possible de couvrir le même spectre de fréquence, mais

en proposant des programmes radio d'une qualité audio voisine de celle du CD. En retour, on profite de l'excellente portée des OC et l'on s'affranchit de la nécessité de rester en vue d'un satellite ou connecté à un serveur Internet.

Développement

Tout a commencé par deux systèmes distincts.

La Deutsche Telekom et le Département des techniques d'émission de Siemens ont présenté leur Projet T2M, qui, travaillant sur porteuse unique, permettait d'émettre, dans une largeur de bande de 10 kHz, simultanément en analogique et en numérique. Il faisait appel au procédé de modulation APSK, une combinaison de modulation d'amplitude avec un déplacement de phase par bonds. Les premières émissions expérimentales ont eu lieu en 1996 (**figure 1**). L'année suivante, le premier récepteur, un autoradio Blaupunkt modifié, était mis en vente. De son côté, la firme française Thomcast (aujourd'hui Thales) concevait le système « Skywave 2000 » qui,

lui, utilisait un grand nombre de porteuses, au moins 47, distantes de 66,666 Hz, avec pour conséquence une plus grande longueur par symbole de données qu'en T2M. Le procédé s'est appelé un temps du « DAB rétréci ». En fait, la technique COFDM (*Coded Orthogonal Frequency Division Multiplex*) fait inévitablement penser au DAB (*Digital Audio Broadcasting*, notre radio numérique), mais se contente d'une bande passante beaucoup plus étroite. Et puis Skywave 2000 permettait aussi la transmission simultanée de signaux analogiques et numériques.

Chaque porteuse est modulée en QAM (*Quadrature Amplitude Modulation* = modulation d'amplitude en quadrature), une variante de l'APSK, dans laquelle deux signaux déphasés de 90 degrés sont modulés en amplitude par échelons avant d'être additionnés. Ainsi, avec quatre échelons, on obtient seize états de modulation (le symbole de données en 16 QAM), mais avec huit niveaux, on en atteint 64 (c'est le 64 QAM).

Il n'y avait aucune raison d'introduire deux standards différents. Pour doter

d'un cadre international les travaux entrepris en commun, sous l'appellation, elle-même plurilingue, de DRM pour Digital Radio Mondiale, un consortium a été fondé en mars 1998. Il compte actuellement plus de soixante-dix firmes ou organisations, parmi lesquelles des stations de radio internationales comme Deutsche Welle, la BBC ou Radio Nederland, aussi bien que des fabricants d'émetteurs (Thales, Telefunken, Harris) et de récepteurs (Sony, Sangean). Des instituts de recherche y sont également associés, comme la fondation Fraunhofer. À la tête du comité de direction de la DRM, c'est Peter Senger qui a été élu, alors ingénieur en chef, à présent directeur technique, des ventes et du marketing chez Deutsche Welle.

Il a encore fallu une longue année pour que les membres aboutissent à un accord sur le procédé. Le mode de transmission retenu est le COFDM, celui auquel les Français avaient accordé la préférence. Quelques éléments de l'autre système se sont également intégrés au standard commun, en particulier la détection et la correction d'erreur. Le type de modulation choisi, le 64 QAM, offre une plus grande insensibilité aux parasites et donc une meilleure qualité de transmission que le 16 QAM.

Encore fallait-il définir le codage à utiliser, la conversion du signal audio en un flux de données numériques. On opta pour l'Advanced Audio Coding (AAC), conçu par « l'Institut Fraunhofer pour les circuits intégrés », situé à Erlangen (Bavière). Il ne faut pas confondre l'AAC avec le MP3, bien qu'ils soient tous deux issus du même sérail. La compression de données dans l'AAC est encore plus poussée qu'en MP3, mais le débit d'information est plus lent, de manière à s'adapter à l'étroitesse de la bande passante dans la gamme des ondes courtes.

Pour élargir encore le spectre audio, on emploie une méthode appelée SBR (Spectral Band Replication, duplication de bande spectrale). En bref, le système simule les fréquences supérieures à 6 kHz par production d'harmoniques, ce qui permet d'atteindre une bande passante audio jusqu'à 15 kHz sans avoir à transmettre les fréquences

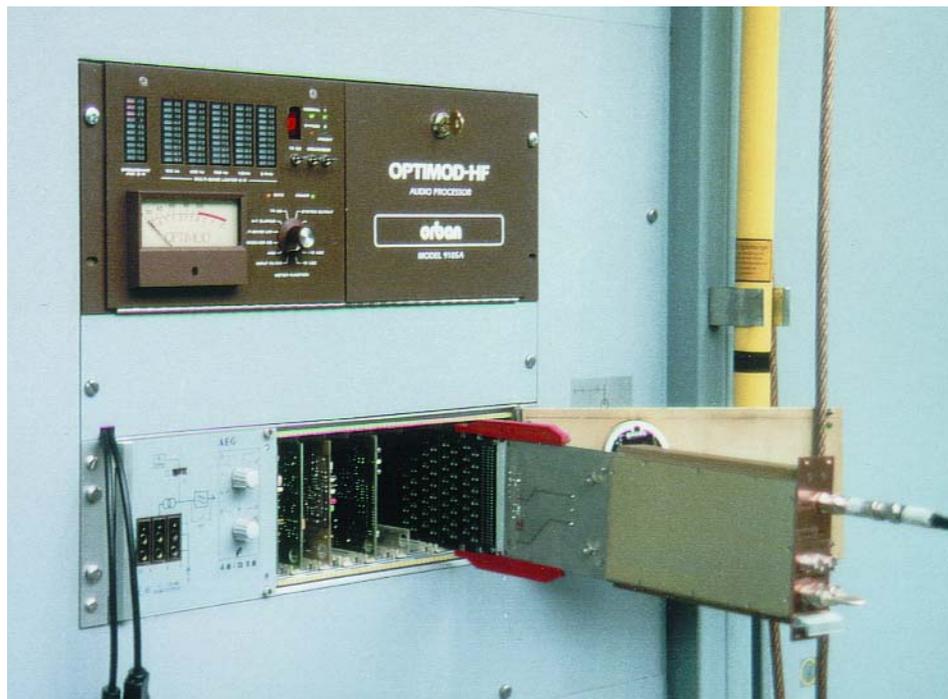


Figure 1. Une platine supplémentaire et voilà les premiers émetteurs numériques sur antenne, dans les ondes courtes (Station Telekom de Jülich, RFA, en avril 1996, photo de H. Weber).

les plus aiguës.

Il semble bien que les promoteurs du système soient quelque peu revenus de l'idée de transmettre simultanément du numérique et de l'analogique sur les mêmes fréquences. Ils préféreraient réserver un domaine dans les bandes de radiodiffusion au profit de DRM, de manière à éviter de se brouiller mutuellement.

L'ITU (*International Telecommunica-*

tions Union) a approuvé la norme DRM en avril 2001 comme recommandation aux pays membres, de quoi ouvrir la voie vers une diffusion mondiale de la nouvelle technique.

Étude de terrain

L'étape suivante consiste en une mise en service expérimentale pendant deux ans avec la participation de tous les radioamateurs et auditeurs des ondes courtes qui le

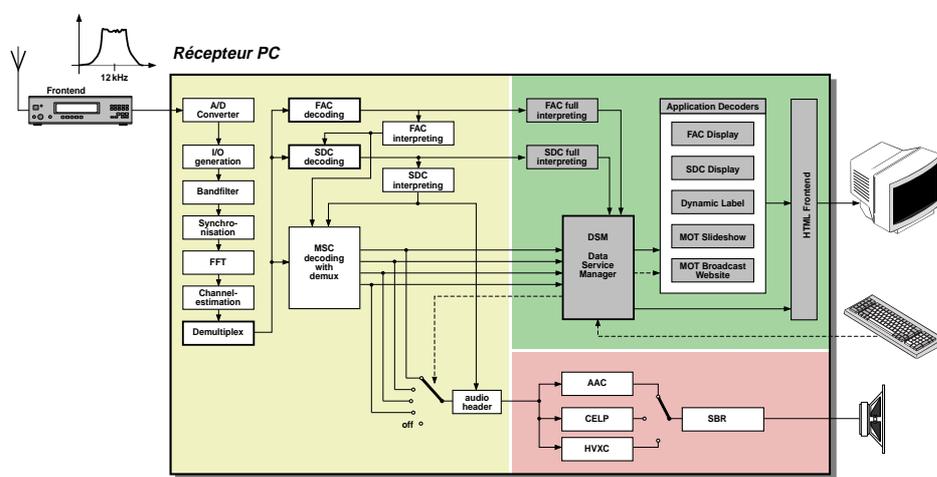


Figure 2. Diagramme de principe du logiciel de réception de DRM (Institut Fraunhofer pour les circuits intégrés).

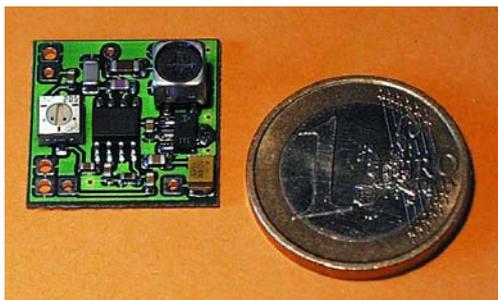


Figure 3. La platine « Mixer Chip » sert à modifier un récepteur mondial pour l'adapter à DRM (Sat-Service Schneider).

désirent et en ont le moyen. L'étude débute à la fin de cette année-ci. Les participants ont à leur disposition une version allégée du logiciel de réception (« FhG Software Radio », cf. **figure 2**) de manière à pouvoir capter les émissions de test et apprécier le système. Le logiciel permet non seulement de décoder les émissions, mais aussi d'évaluer la qualité de réception et en fournir des rapports à DRM. Pour y participer, il faut disposer d'un PC puissant et un récepteur ondes

courtes de haute voire de très haute qualité (p. ex. AOR 7030 ou Yaesu FRG 100). Il doit subir une modification pour lui ajouter une petite platine de 20 x 20 mm, le *Mixer Chip* de la **figure 3**, qui produit une fréquence intermédiaire de 12 kHz, nécessaire à la commande de la carte sonore du PC. En outre, le récepteur doit être capable de recevoir une bande passante de 10 kHz, parce que les filtres ordinairement utilisés ne laissent passer que 6 kHz, ce qui est insuffisant pour exploiter la totalité du signal de DRM.

Il est possible de se procurer les platines nécessaires auprès du Sat-Service de Schneider, par exemple, un distributeur établi en Saxe. Avec la version LC (cf. **figure 4**), on peut exécuter la modification pour moins de 40 €, avec un oscillateur à quartz, cela coûte le double environ. Celui qui ne désire pas bricoler dans l'appareil peut confier le travail à un professionnel. Schneider propose aussi un récepteur complètement équipé à partir de 799 €. Le logiciel de réception coûte 60 € de plus.

Récepteur

Évidemment, à long terme, ce n'est pas une solution d'écouter les émissions de DRM par le truchement d'un PC. Lors de l'IBC en septembre à Amsterdam, on a sorti, en présérie, un récepteur DRM (**figure 5**), conçu par Coding Technologies, une firme germano-suisse, dont la majorité du personnel avait travaillé auparavant à l'Institut Fraunhofer, en collaboration avec la BBC et le fabricant AFG. Les fonctions que le PC remplissaient sont reprises dans un module supplémentaire (représenté en diagramme fonctionnel à la **figure 6** Tri-media DSP).

D'autres firmes travaillent également au récepteur DRM. Ainsi le Sat-Service de Schneider va également introduire sur le marché son propre mini-récepteur (diagramme fonctionnel à la **figure 7**).

Lancement en juin

Le début officiel des émissions de DRM devrait coïncider avec la Conférence radio administrative mondiale, qui se tiendra en juin de l'année prochaine. D'ici là, les premiers récep-

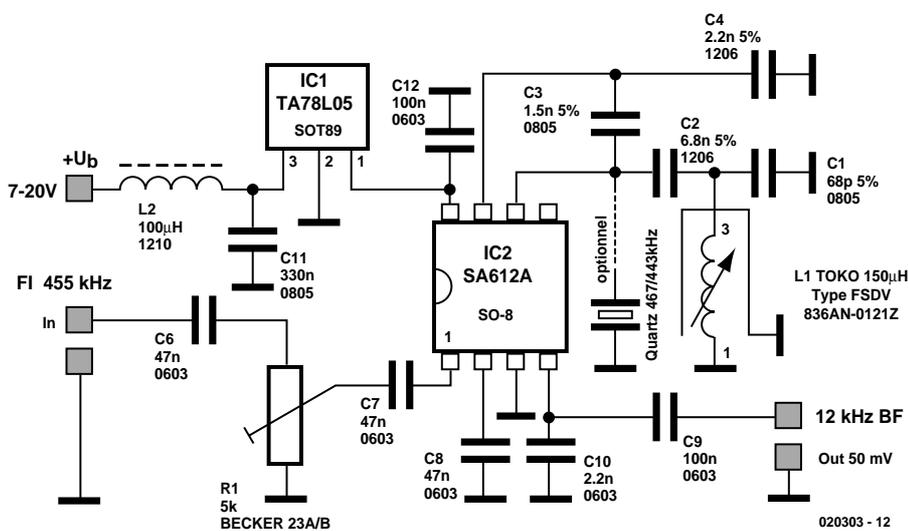


Figure 4. Schéma du mélangeur LC (Sat-Service Schneider).



Figure 5. Le prototype de récepteur DRM (Coding Technologies).

Liens Internet pour DRM :

- www.drm.org/indexdeuz.htm
(DRM généralités)
- www.mercom.co.uk/site3/terr2.htm
- www.drm.org/system/globsample.htm
- webapp.etsi.org/action%5CMV/MV20020927/es_201980v010201m.pdf
(fichier .pdf des spécifications les plus récentes)
- www.drmx.org (pour, entre autres, l'inscription au projet de test DRM)
- www.rnw.nl/realradio/html/drm.html
(Page DRM de la station néerlandaise Radio Netherlands Wereldomroep)
- www.drm-info.de (page Internet privée concernant ce thème)
- www.CodingTechnologies.com
(récepteur DRM)
- www.iis.fraunhofer.de/dab/products/drmreceiver/index.html (FhG-Software-Radio)
- www.sat-schneider.de (récepteur DRM, composants pour la modification de récepteur)

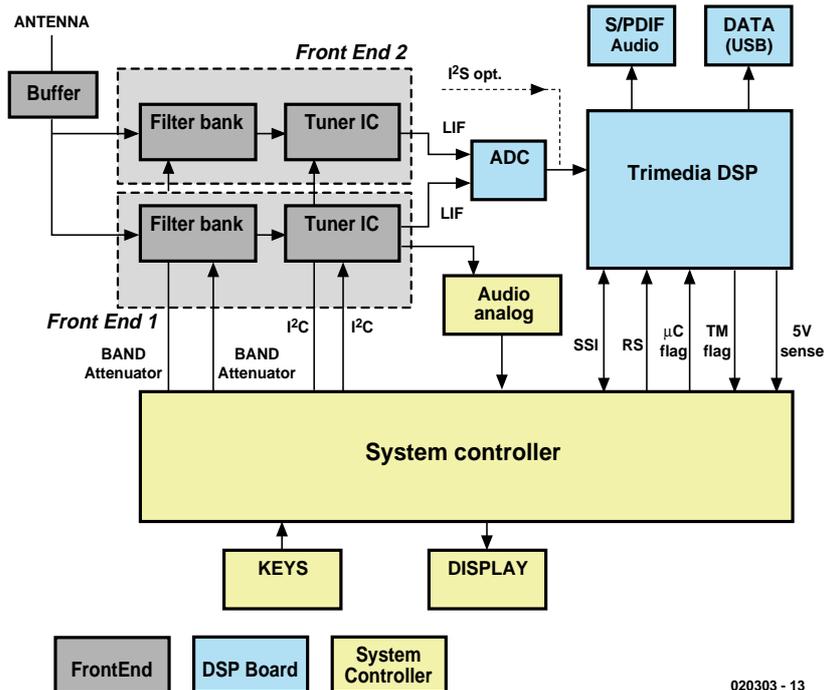


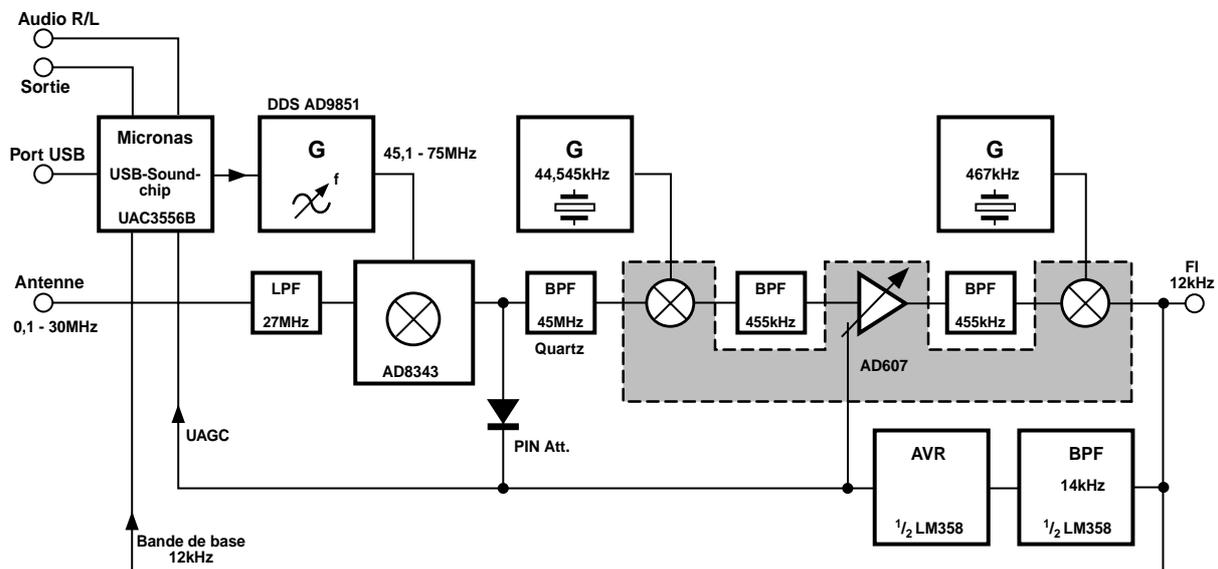
Figure 6. Diagramme fonctionnel d'un récepteur DRM (Coding Technologies).

teurs produits en série seront disponibles dans le commerce. Dans un premier temps, les prix seront encore élevés, avant que la production de masse ne soit réellement lancée. Après quoi, les nouveaux récepteurs devraient se vendre à peine plus cher que ceux d'aujourd'hui. Ils devront permettre la réception en analogique et en numérique, le passage au tout numérique se prépare

en douceur, pour se terminer au plus tôt dans quinze ans, probablement encore après dans les pays moins favorisés. La plupart des grandes stations émettrices en ondes courtes et de nombreuses plus petites ont manifesté leur intérêt à DRM. Les installations modernes de transmission se prêtent aisément, à peu de frais, à la transformation qui leur permettra de

diffuser en DRM également. Les ondes moyennes peuvent aussi connaître une renaissance grâce à l'amélioration de la qualité sonore. Aux dernières nouvelles, Mega Radio, la chaîne allemande très populaires chez les jeunes et qui « collectionne » depuis des années les fréquences OM dans presque tous les Länder, aurait planifié la couverture en qualité numérique de la totalité de l'État fédéral.

(020303)



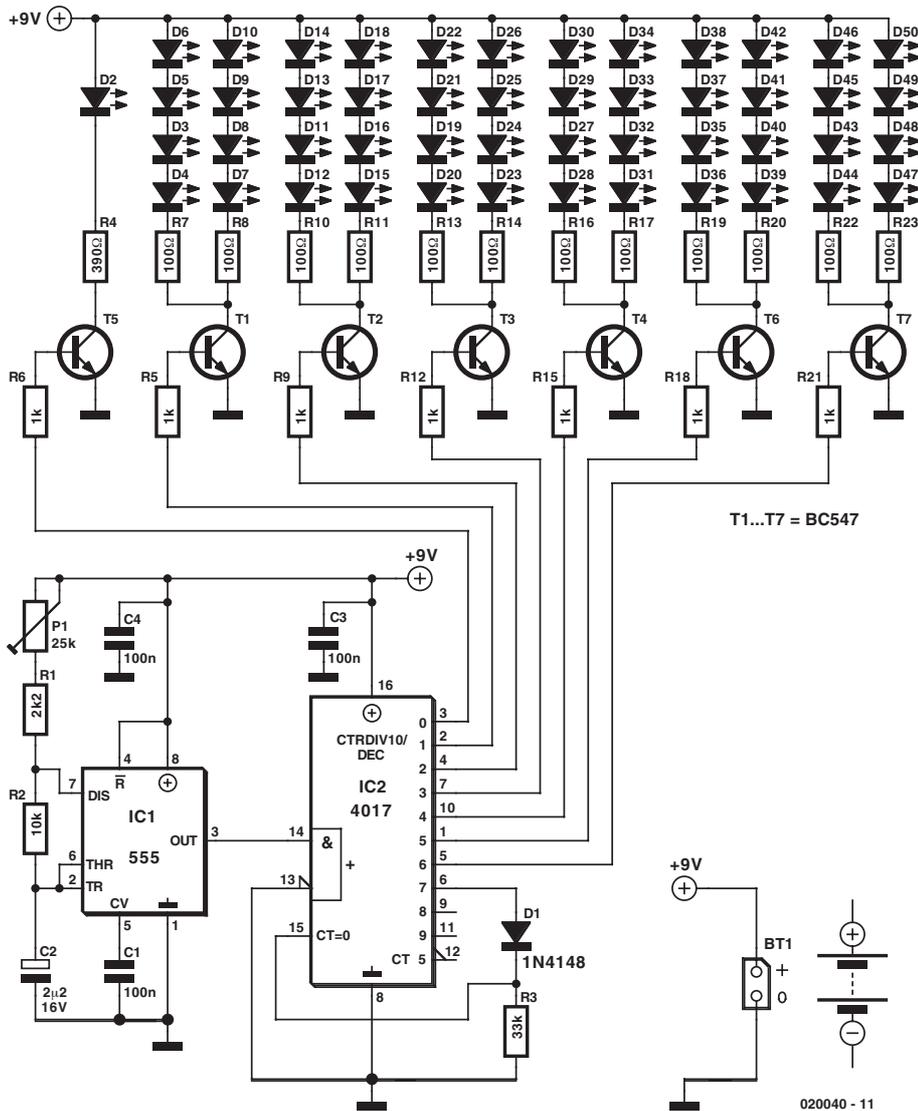
020303 - 14

Figure 7. Diagramme fonctionnel du mini-récepteur DRM (Sat-Service Schneider).

Étoile de Noël

A. Rossiusw

Ce montage produit un effet lumineux tout ce qu'il y a de plus rudimentaire, mais que l'on pourra utiliser, pour peu que l'on adopte une disposition adéquate des LED, en vue de réaliser un objet d'ornementation pour Noël.



Les LED sont disposées en forme d'étoile. On aura tout d'abord allumage de la LED centrale puis des 8 LED constituant l'« anneau » central, D3 à D10, puis l'anneau suivant et ainsi de suite jusqu'à ce que l'anneau extérieur constitué par D43 à D50 se soit allumé. Ce cycle reprend alors au début. L'ajustable P1 permet de jouer sur la vitesse de déroulement, le tempo, de ce processus.

Liste des composants

Résistances :

- R1 = 2kΩ
- R2 = 10 kΩ
- R3 = 33 kΩ
- R4 = 390 Ω
- R5,R6,R9,R12,R15,R18, R21 = 1 kΩ
- R7,R8,R10,R11,R13,R14,R16,R17, R19,R20,R22,R23 = 100 Ω
- P1 = ajustable 25 kΩ

Condensateurs :

- C1,C3,C4 = 100 nF
- C2 = 2µF/16 V radial

Semi-conducteurs :

- D1 = 1N4148
- D2 à D50 = LED rouge
- T1 à T7 = BC547
- IC1 = 555
- IC2 = 4017

Divers :

- pile compacte 9 V avec connecteur à pression

Figure 1.

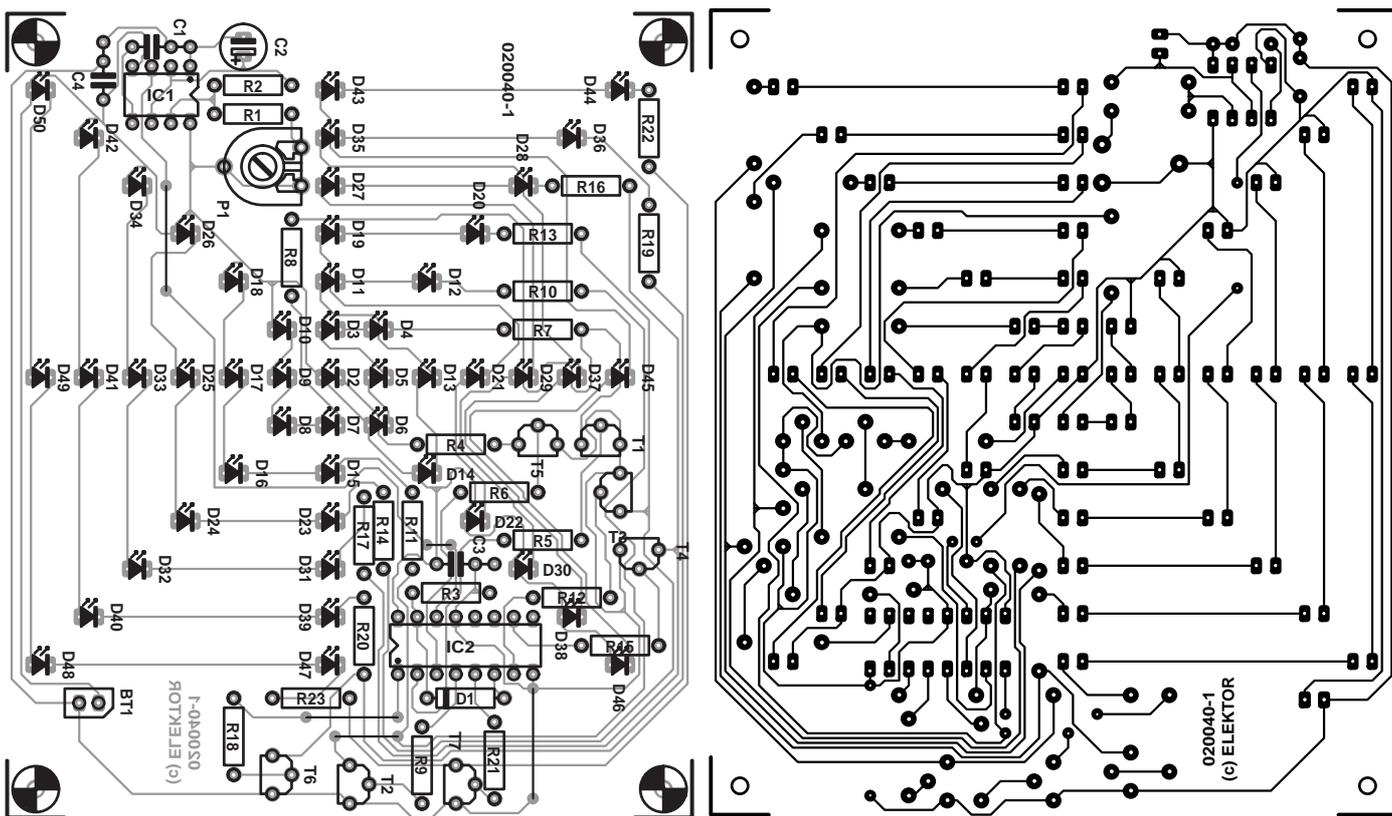
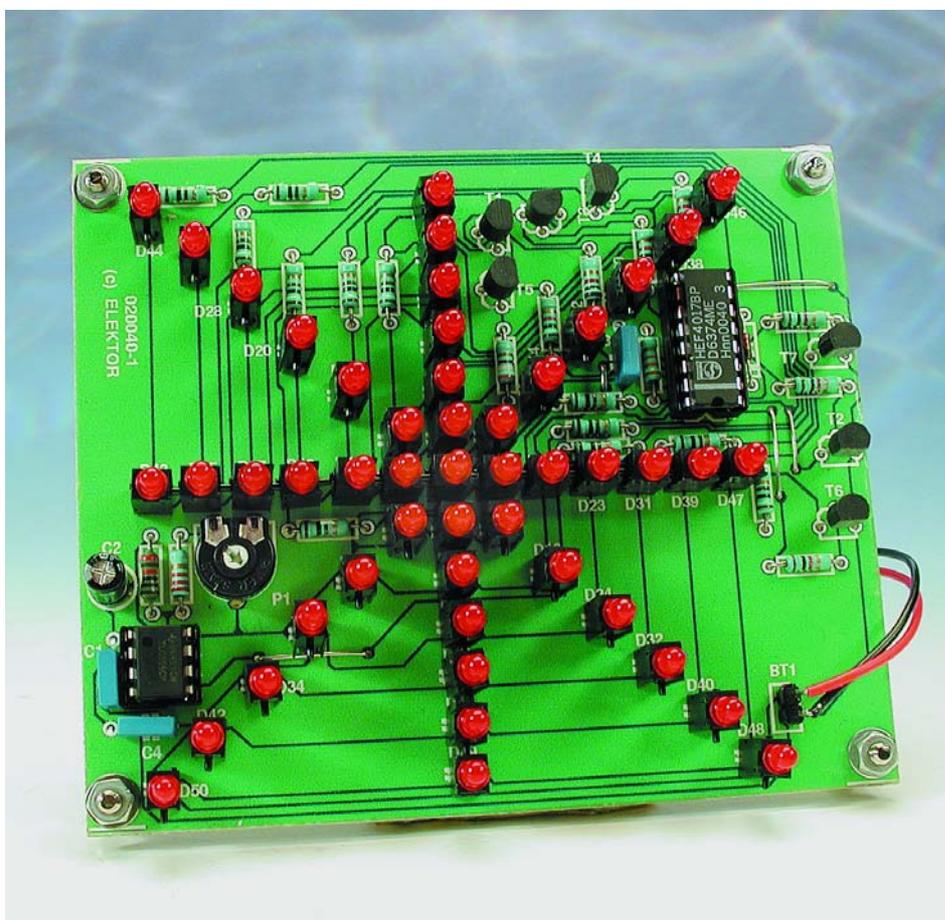


Figure 2.

Le fonctionnement de l'électronique n'est guère plus compliqué que l'histoire des LED dont nous venons de donner le chronodiagramme visuel. Un temporisateur du type 555 monté en multivibrateur astable sert de source de signal d'horloge pour un compteur décimal, IC2. Après une remise à zéro (reset) la sortie Q0 est active, chaque impulsion d'horloge entraînant une incrémentation du compteur. Le transistor-tampon correspondant passe en conduction et les 2 chaînes de LED qu'il commande d'allument. La 7ème impulsion d'horloge produit une remise à zéro et l'ensemble du processus recommence à partir de la sortie Q0. Un ornement de Noël ne vit pas uniquement par son fonctionnement mais a également beaucoup du chic de son aspect. Ceci est l'une des raisons qui nous ont amené à concevoir un dessin de platine sur lequel les LED viendront de monter en lignes et carrés parfaitement symétriques.

(020040)



Allô ! Maman ? Ici bébé !

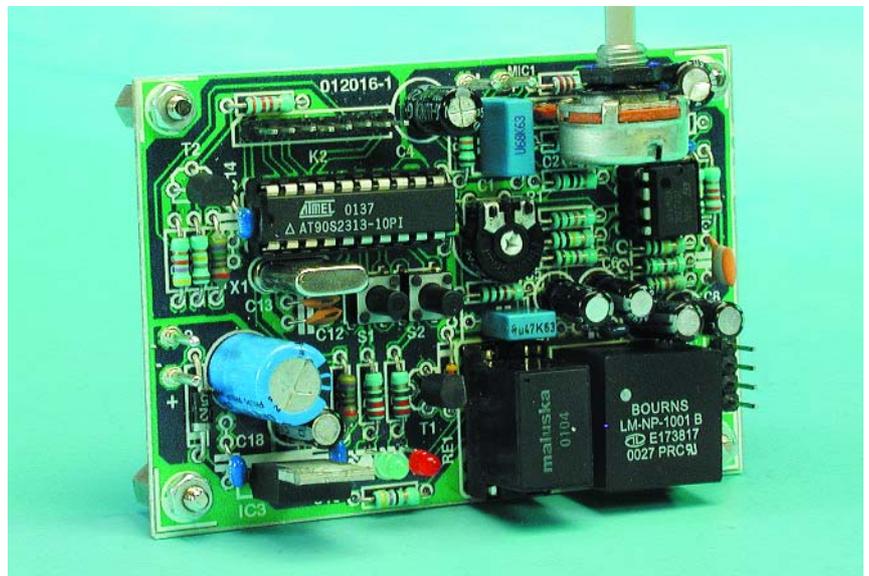
Projet : Thomas Finke

Les interphones de surveillance de chambre d'enfant souffrent de deux inconvénients marquants : leur portée est faible et ils produisent des perturbations électriques. Autant ne pas apporter de contribution délibérée à la « purée de pois » EM, surtout dans le voisinage immédiat d'un nouveau-né. Ici, les cris de bébé n'induisent... que la composition d'un numéro de téléphone.

Des interphones à sens unique pour la surveillance d'une chambre d'enfant, il en existe de multiples versions. Certaines utilisent le réseau électrique, d'autres les ondes hertziennes comme canal de communication, chaque fois avec les inconvénients qui en résultent. La portée des modèles radio est restreinte, contrairement à ceux qui utilisent le réseau électrique intérieur, pour autant que le récepteur soit branché sur la même phase. Mais aussi bien l'un que l'autre sont sensibles aux parasites et en outre, ils en répandent dans l'environnement. Le téléphone spécial pour chambre d'enfant présenté ici évite ces inconvénients. Il se branche à une prise du téléphone et détecte, à l'aide d'un microphone, quand le bébé se met à pleurer. Aussitôt, il compose un numéro enregistré, celui du portable des parents, par exemple. Nul besoin de décrocher, puisque le numéro de l'appelant s'affiche, suivi d'un message d'alerte. Aucun risque d'ignorer que votre présence est souhaitée au chevet de bébé.

Évaluation du bruit

Les cris de bébé atteignent le montage de la **figure 1** par le truchement du microphone à électret MIC1, lequel est alimenté à travers R1, avec l'appoint du découplage introduit par R7 et C4. Le condensateur C1 transfère la partie alternative du signal à basse fréquence à l'entrée inverseuse de l'amplificateur opé-



rationnel IC2.A. Avec le TS922IN, nous avons affaire à un double amplificateur de chez ST, à excursion totale en sortie, disponible chez Farnell. Il permet également de travailler avec un gain de boucle très élevé, exprimé par $1 - R6/R5$. En outre, la tension d'alimentation de la puce peut être assez haute. On retrouve sur le curseur de P2 le signal audio, peut-être pas d'une très haute qualité, mais là n'est pas

le but. C2 empêche l'oscillation à haute fréquence.

Le diviseur de tension formé par R2, R3, P1, R4, associé à C3 en amortissement, ramène à 2,5 V la composante continue du signal. Voilà qui permet d'alimenter l'amplificateur opérationnel sous une tension unique.

Le signal atténué par P2 est maintenant appliqué à un comparateur qui, contrairement à ce que l'on pourrait

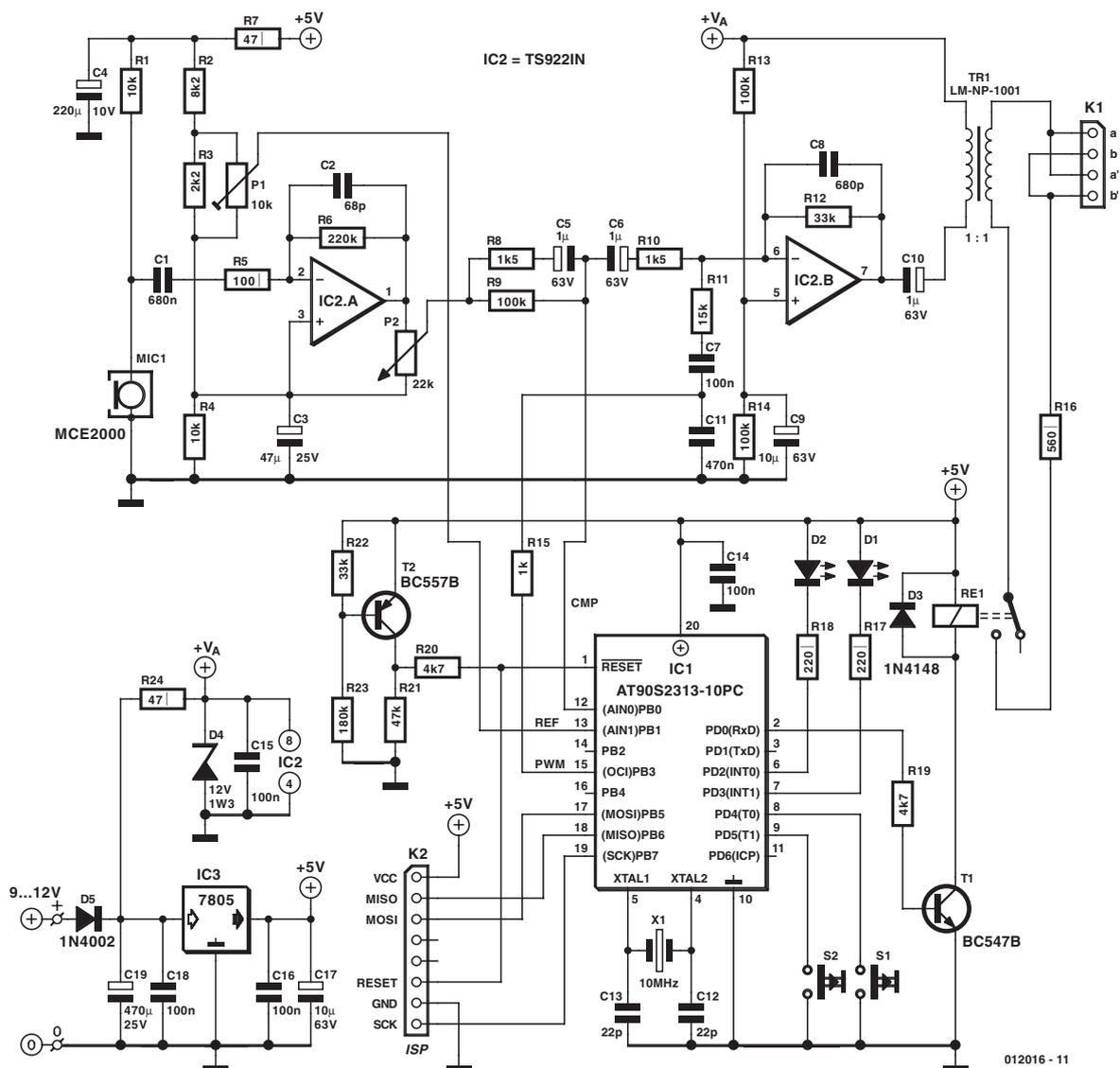


Figure 1. Les électroniques analogique et numérique travaillent de concert.

imaginer, n'est pas IC2.B, mais le microcontrôleur AT90S2313 lui-même. Ses deux lignes PB0/AIN0 (broche 12) et PB1/AIN1 (broches 13) peuvent en effet être appelées à servir de comparateur analogique. Comme le montre le schéma de la structure interne du contrôleur à la **figure 2**, ces comparateurs ne sont pas dotés de sortie vers l'extérieur, ils servent à produire une interruption à l'intérieur. Exactement ce dont nous avons besoin ici.

À l'entrée AIN1 du comparateur se trouve une tension de référence REF, réglée par P1, et sur AIN0, le signal à basse fréquence CMP. Si le signal audio dépasse la tension de référence, une interruption aura lieu sur le contrôleur. Le logiciel transforme

alors AIN0 d'entrée en sortie, qu'il maintient à zéro. Le signal audio s'en retrouve court-circuité (R8 constitue une limitation de courant) et remplacé par un signal DTMF (en modulation de largeur d'impulsion) sur la ligne. La production de ce signal est expliquée dans l'encadré. Tant pour l'audio que pour les impulsions, un autre amplificateur, IC2.B, procure encore un gain modéré, deux fois pour le DTMF et une dizaine pour le signal audio. Ici aussi, le diviseur fait de R13 et R14/C9 amène la polarisation à la moitié de la tension d'alimentation (de VA), pour un fonctionnement correct. La bande passante de chaque étage amplificateur est de 10 kHz, ce qui nous satisfait largement.

Occultation

La protection par occultation (*brown out*) empêche le microcontrôleur de prendre des initiatives désordonnées quand sa tension d'alimentation est trop basse, lors de la mise en ou hors service, par exemple. Une précaution indispensable, si l'on pense, entre autres, que le numéro d'appel à composer se trouve dans l'EEPROM : il serait malencontreux de permettre au contrôleur l'accès en écriture quand il est « en manque » ! Plutôt que de faire appel à l'incontournable CI prévu à cet effet, nous avons opté pour une solution discrète, confiée au transistor T2, le mérite de l'originalité en revient d'ailleurs à Atmel, qui publie le circuit dans sa note d'application. Bon marché et facile à trouver dans le commerce, que demander de plus ? Le circuit d'occultation actionne tout bêtement la ligne

de mise à zéro du microcontrôleur aussi longtemps que la tension d'alimentation ne satisfait pas au minimum requis.

Numérotation

Le cordon de la fiche téléphonique se raccorde à notre montage sur K1. Le microcontrôleur peut provoquer la prise de ligne, auquel cas le courant de ligne circulera entre a et b via R16 et le transformateur TR1, un modèle standard de Bourns, assez facile à trouver. Au primaire, nous avons le microcontrôleur qui produit, par modulation d'impulsion, le signal DTMF nécessaire à la numérotation. L'appareil ne débranche jamais le téléphone connecté à la même prise, comme un télécopieur ou un modem. On peut donc utiliser un relais (à lame souple) simple, sans contact inverseur, et même commander les tonalités DTMF du téléphone en circuit.

Un peu de tout

Les touches S1 et S2 commandent toutes les fonctions, les LED D1 et D2 reflètent toutes les situations. Cela conduit à une manipulation un rien ascétique, surtout quand il s'agit d'introduire le numéro à appeler. Mais comme le but du jeu n'est pas d'en changer comme de chemise, investir dans un clavier numérique et un écran LCD nous a semblé hypothétique à rentabiliser.

Classiquement, la fourniture d'énergie passe par un adaptateur secteur (entre 9 et 12 V) une diode (D5) à interdire la mauvaise polarité, un condensateur réservoir de lissage et un régulateur de tension fixe de +5 V. Comme l'amplificateur opérationnel va s'alimenter directement sur l'adaptateur, il vaut mieux s'assurer qu'il ne délivre pas plus de 12 V. La diode D5 a pour mission de protéger des pointes de surtension éventuelles, certaine-

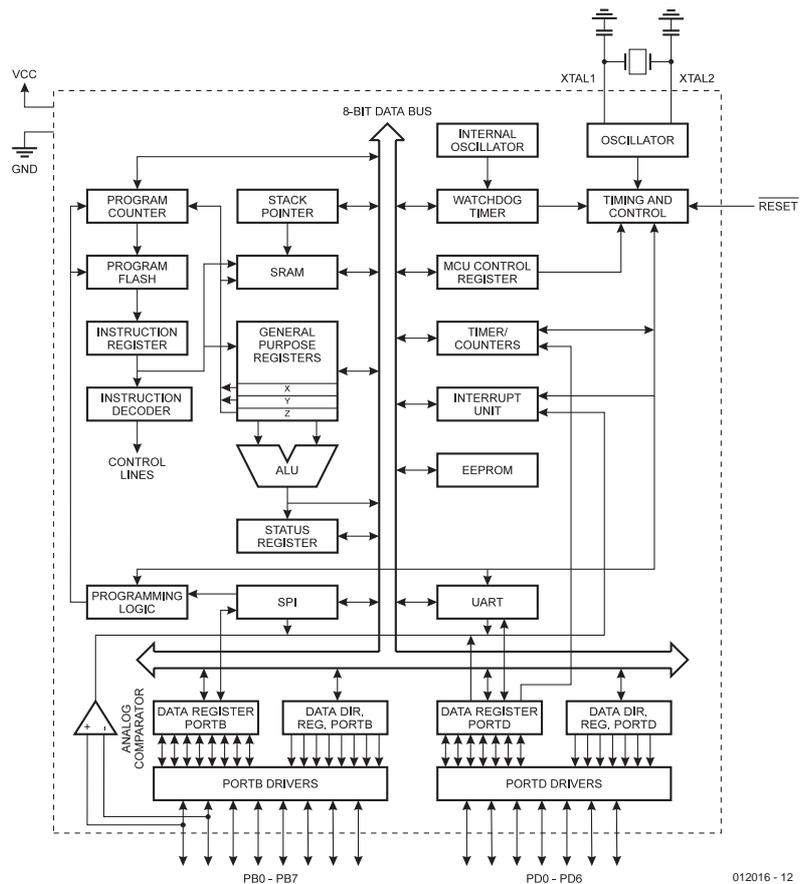


Figure 2. Le contrôleur est équipé d'origine d'un comparateur intégré.

ment pas de dilapider à longueur de temps une tension excessive. On aurait pu envisager une alimentation par les fils du téléphone, mais une bonne séparation galvanique à l'égard de ces tensions relativement hautes est préférable, dès le moment où l'objet de toutes les attentions est un bébé.

La fréquence d'horloge du contrôleur, à 10 MHz, semble fort élevée, mais c'est le prix à payer pour pouvoir générer les fréquences DTMF avec une précision suffisante. À condition de disposer du câble de programmation approprié, on peut programmer le montage en circuit en passant par K2. Son brochage cor-

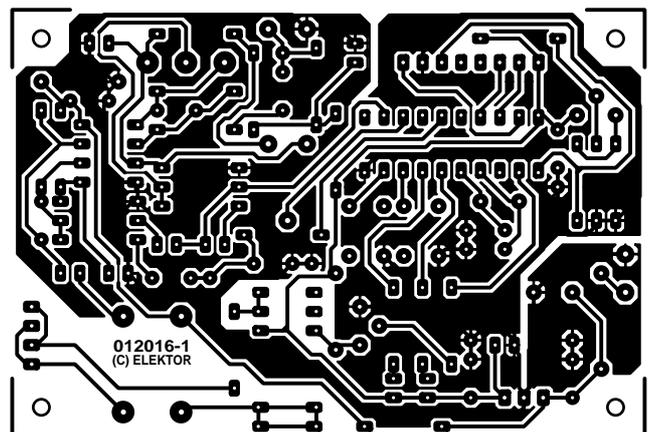
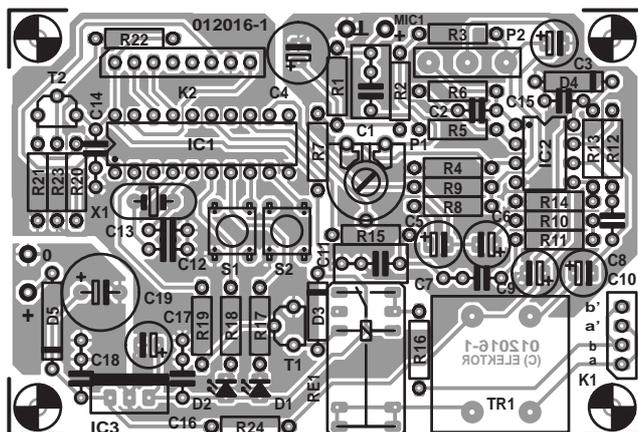


Figure 3. Les plans de la platine à simple face pour le téléphone de bébé.

Synthèse des signaux DTMF

Les générateurs DTMF simples (comme le TP5088 de NS) sont devenus pratiquement introuvables. Les nouvelles puces débordent de fonctions accessoires dont nous n'avons que faire. On les trouve difficilement et elles réclament de nombreuses lignes de commande (le MT80444 utilise encore le signal $\Phi 2$ de l'ère préhistorique du 6502). La tentation était donc grande de fabriquer le signal par logiciel.

DTMF est l'abréviation de *Dual-Tone MultiFrequency signaling*, signalisation à double tonalité. Chaque chiffre est représenté par la superposition de deux signaux, dont les fréquences sont choisies parmi quatre tonalités basses pour l'un, parmi quatre aiguës pour l'autre. La tolérance sur ces huit fréquences n'est que de 1,5 % pour que le central téléphonique puisse les identifier.

Le contrôleur Atmel utilisé ne dispose pas d'un convertisseur N/A, mais le générateur d'impulsions modulées en largeur (PWM) s'acquitte aussi bien de la tâche. Sa résolution est de 8 bits et il travaille à partir d'une fréquence de 19,6 kHz, soit $f_{CK} / 510$. La largeur du signal PWM est fixée pour chaque période dans une valeur consignée dans une table. Le spectre d'un tel signal présente, outre les pointes sur les fréquences prescrites des tonalités sinusoïdales, une pointe à $f = 0$ (le signal n'est en effet pas exempt de composante continue, mais les condensateurs C7 et C11 sont là pour cela). D'autres harmoniques y sont aussi présentes, mais un filtre passe-bas simple, composé de R15 et C11, les atténue suffisamment pour que le central n'entende que les tonalités de numérotation. La différence d'amplitude requise entre le groupe des fréquences hautes et celui des basses est prise en compte dans la table des sinus, directement. Lorsqu'on ne peut pas atteindre la précision de fréquence nécessaire avec une seule période enregistrée, on en utilise deux ou trois, si bien qu'au total, la table des sinus couvre environ 200 octets.

Liste des composants

Résistances :

R1, R4 = 10 k Ω
 R2 = 8k Ω
 R3 = 2k Ω
 R5 = 100 Ω
 R6 = 220 k Ω
 R7, R24 = 47 Ω
 R8, R10 = 1k Ω
 R9, R13, R14 = 100 k Ω
 R11 = 15 k Ω
 R12, R22 = 33 k Ω
 R15 = 1 k Ω
 R16 = 560 Ω
 R17, R18 = 220 Ω
 R19, R20 = 4k Ω
 R21 = 47 k Ω
 R23 = 180 k Ω
 P1 = ajustable 10 k Ω horizontal
 P2 = potentiomètre 22 k Ω linéaire, ne pas utiliser de version mono

Condensateurs :

C1 = 680 nF
 C2 = 68 pF
 C3 = 47 μ F/25 V axial
 C4 = 220 μ F/10 V axial
 C5, C6, C10 = 1 μ F/63 V axial
 C7, C14 à C16, C18 = 100 nF céramique
 C8 = 470 pF
 C9, C17 = 10 μ F/63 V axial
 C11 = 470 nF

C12, C13 = 22 pF
 C19 = 470 μ F/25 V axial

Semi-conducteurs :

D1 = LED 3 mm rouge
 D2 = LED 3 mm verte
 D3 = 1N4148
 D4 = diode zener 12 V/1W3
 D5 = 1N4002
 T1 = BC547B
 T2 = BC557B
 IC1 = AT90S2313-10PC (programmé **EPS 012016-41**)
 IC2 = TS922IN (ST - chez Farnell)
 IC3 = 7805

Divers :

K1 = embase mâle à 2 rangées de 7 contacts
 K2 = embase mâle à 2 rangées de 8 contacts
 S1, S2 = bouton-poussoir 6 x 6 mm
 Re1 = relais sub-miniature 16 x 119 x 11,5 mm avec contact SPDT tel que, par exemple Maluska FRS1B-S DC 5V, (5 V, 56 Ω , Conrad 505188
 X1 = quartz 10 MHz (à $C_{load} = 32$ pF parallèle)
 Tr1 = transducteur Ligne (Bourns LM-NP-1001 B)
 MIC1 = capsule micro à électret tel que, par exemple, MCE2000 (Monacor)

respond à celui prévu par Lattice pour ses ispCPLD. On peut ainsi, avec le même câble, programmer des puces de Atmel et de Lattice. Si vous comptez vous procurer le micro-contrôleur préprogrammé, vous pouvez franchement vous passer de K2 (et du cordon).

Construction, alignement et emploi

L'agencement de la platine à simple face, garnie des composants, est illustré par la **figure 3**. Le montage se révèle d'une compacité telle qu'il sera aisé de tout mettre dans une boîte en plastique qui ne le sera pas moins.

On commence par poser tous les composants, du plus petit au plus grand, puisque des ponts de câblage, il n'y en a aucun, pas plus que de singularités à considérer lors du travail de soudage. Mieux vaut choisir des poussoirs dotés d'un bouton assez long, question de ne pas aller lutiner l'électronique en cours de programmation.

Par souci de sécurité, la capsule de microphone sera logée dans le boîtier, mais après l'avoir perforé, de manière à entendre les cris de bébé ! Le cordon téléphonique, on l'ancrera fermement au coffret à l'aide d'un passe-câble.

Après les vérifications d'usage, arrive le moment de mettre l'appareil sous tension. La LED verte s'allume comme témoin de fonctionnement. On peut alors désactiver l'appareil d'une pression sur le poussoir S1 et la LED verte s'éteint. Si, en période d'activité, un bruit est détecté, la LED verte se met à clignoter. Profitons-en pour régler le gain sur P2 de manière à ce qu'en présence d'un signal audio puissant, comparable aux cris du bébé, IC2.B ne soit pas saturé. C'est facile à établir à l'aide d'un oscilloscope. On peut alors régler la sensibilité avec P1. N'importe quel bruit ne déclenche pas l'alarme, il faut qu'il dépasse une certaine fréquence et dure un temps suffisant. Fréquence et durée sont inscrits en EEPROM et qui le souhaite peut les modifier. Lorsque ces conditions sont remplies, Re1 s'enclenche, ce qui revient à décrocher le combiné. La LED rouge s'éclaire en permanence. Ensuite, le microcontrôleur émet les tonalités d'appel qui correspondent au numéro mémorisé et après quelques secondes d'attente, le temps de deux à trois sonneries, il raccroche. L'appareil reste dans cet état pour un temps encore, avec la LED rouge clignotante, pour éviter d'émettre des appels répétitifs. Puis finalement, il en revient à la situation de départ.

À l'aide !

Lorsque l'appareil est actif, une pression sur S2 provoque la composition immédiate du

numéro programmé. En parallèle sur S2, on peut monter un contact de relais ou le collecteur ouvert d'un autre appareil, un système d'alarme ou un récepteur de radio, pour qu'il puisse également déclencher la numérotation.

Saisie du numéro d'appel

Programmer le numéro qui devra être composé automatiquement n'est pas précisément commode, mais puisqu'il restera mémorisé dans l'EEPROM, même lorsque l'appareil est éteint, on ne devra s'y soumettre que rarement. Une pression longue (>2 s) sur S1, avec l'appareil en fonctionnement, le fait passer en mode de programmation, signalé par la LED rouge allumée et la verte éteinte. On inscrit alors le premier chiffre en appuyant le nombre de fois voulu sur S1 ; on n'y touche pas s'il s'agit d'un zéro. On confirme et on passe au deuxième chiffre en poussant sur S2. Reste à « pianoter » les autres chiffres de la même manière. Enfin, une autre pression longue sur S1 met un point final à la programmation. Il est permis de donner dix impulsions pour un ou plusieurs des chiffres, cela provoquera une pause pendant la numérotation, ce qui peut s'avérer nécessaire s'il faut commuter sur un autre central ou obtenir une ligne extérieure. En cas de faute de frappe, il faut exercer deux pressions longues sur S1 et recommencer depuis le début.

Interro !

À l'issue d'une telle séance de comptage, rien ne remplace une répétition générale. Une longue pression sur S2 et le contrôleur va se

Adresse	Fonction	par défaut
\$00-\$1F	Numéro d'appel en code ASCII, \$00 est le caractère de terminaison	
\$20	Fréquence-seuil au-delà de laquelle le bruit est classé pleurs de bébé ($f_s = X \cdot 38 \text{ Hz}$)	10
\$21	Longueur de la durée de la fenêtre de surveillance en $2^X \cdot 6,5 \text{ ms}$	10
\$22	Il faut autant d'intervalles « bruyants » à raison de 6,5 ms dans la fenêtre de surveillance pour un déclenchement de l'alarme	80
\$23	Intervalle de temps séparant la saisie de numéro et la remise en place du combiné (sonneries) $X \cdot 6 \text{ s}$ (environ)	4
\$24	L'appareil reste inactif après chaque saisie pendant $X \cdot 16 \text{ s}$ avant qu'il n'y ait, en cas de bruit permanent, recombinaison du numéro	10
X est le contenu de l'emplacement de mémoire		

voir contraint de réciter sa leçon. La LED rouge s'allume en continu. La verte, par ses clignotements, nous énumère le nombre d'unités que totalise le premier chiffre mémorisé, marque une pause et se répète. Si c'est un zéro, elle reste éteinte, naturellement. Nous passons au chiffre suivant avec S2. Après l'énoncé du dernier chiffre, la LED rouge s'éteint et la verte s'allume. S'il a bien retenu sa leçon, le montage en sait assez pour veiller sur bébé.

Programmation

Le logiciel qui anime le circuit est complètement écrit en C. Les

constantes (seuil de fréquence de reconnaissance du son, valeurs temporelles) restent modifiables sans compilateur, puisqu'elles se trouvent dans l'EEPROM du microcontrôleur. C'est évidemment là aussi que s'écrit le numéro d'appel et il est possible de le mémoriser directement si l'on dispose du cordon de programmation.

Si l'on maintient enfoncés S1 et S2 lors de la mise sous tension, les valeurs prévues par défaut s'inscrivent dans l'EEPROM.

(012016)

LM3812/LM3813/LM3814/LM3815

Mesure de courant avec sortie MLI

Fonctions spéciales
Analogique

ELEKTOR

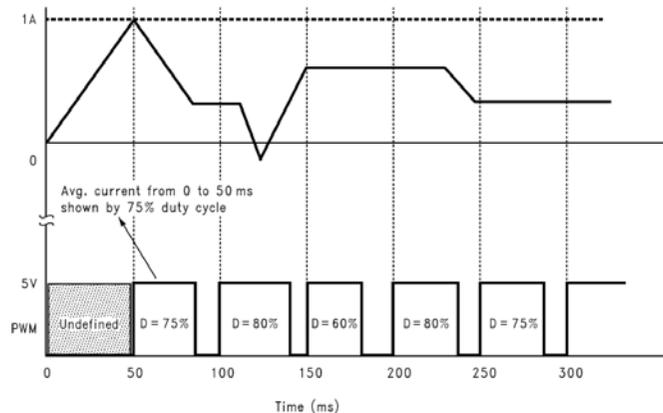
INFOCARTE 12/2002

Tableau de référence:

Courant	MLI	Courant	MLI
1,00	95,5	-1,00	4,5
0,95	93,2	-0,95	6,8
0,90	90,9	-0,90	9,1
0,85	88,6	-0,85	11,4
0,80	86,4	-0,80	13,6
0,75	84,1	-0,75	15,9
0,70	81,8	-0,70	18,2
0,65	79,5	-0,65	20,5
0,60	77,3	-0,60	22,7
0,55	75,0	-0,55	25,0
0,50	72,7	-0,50	27,3
0,45	70,5	-0,45	29,5
0,40	68,2	-0,40	31,8
0,35	65,9	-0,35	34,1
0,30	63,6	-0,30	36,4
0,25	61,4	-0,25	38,6
0,20	59,1	-0,20	40,9
0,15	56,8	-0,15	43,2
0,10	54,5	-0,10	45,5
0,05	52,3	-0,05	47,7
0,00	50,0	0,00	50,0

Dans le cas des types 7 A, le courant est à multiplier par 10 (!).

Timing Diagram



LM3812/LM3813/LM3814/LM3815

Mesure de courant avec sortie MLI

Fonctions spéciales
Analogique

ELEKTOR

INFOCARTE 12/2002

Fabricant :

National Semiconductor

www.national.com

www.national.com/ds/LM/LM3812.pdf

www.national.com/ds/LM/LM3814.pdf

Caractéristiques techniques :

- Très faibles pertes par shunt interne de 0,004 Ω
- Ne requiert pas de détecteur externe
- Tension d'alimentation : 2 à 5,25 V
- Erreur maximale $\pm 2\%$ à température ambiante (erreur de détecteur comprise)
- Courant de repos faible en mode *shutdown* (2,5 μ A)
- Mesure précise par convertisseur A/N delta/sigma avec intervalle d'échantillonnage de 50 ou de 6 ms
- Sortie MLI (PWM) compatible microcontrôleur avec information d'intensité et de sens de courant
- Faible sensibilité aux écarts de température
- Suppression interne des pics de courant, même lors de la mise en fonction

Description :

Le LM381x est un circuit de mesure de courant qui permet une mesure de courant précise en toute simplicité et ce avec des pertes extrêmement faibles au niveau de la résistance de shunt interne d'une valeur typique de 0,004 Ω . Les types LM3812 et LM3814 conviennent à la mesure de potentiels d'alimentation positifs, les types LM3813 et LM3815 étant eux prévus pour les potentiels négatifs.

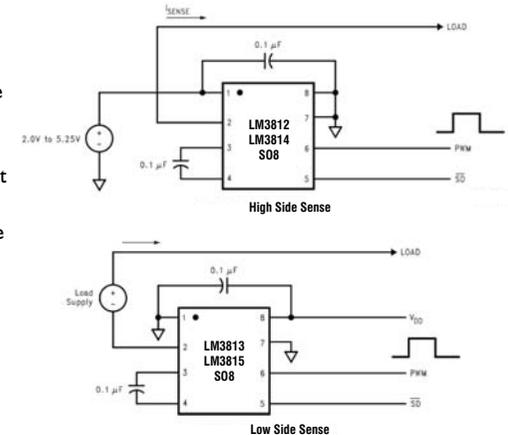
La mesure de précision est l'affaire d'un CAN (convertisseur analogique/numérique) delta-sigma qui moyenne le courant sur une durée de, selon le mode, 50 ms (LM3812/LM3813) ou 6 ms (LM3814/LM3815), de façon à ne pas tenir compte des crêtes de courant.

Ces circuits sont dotés d'une sortie modulée en largeur d'impulsion (MLI = PWM pour *Pulse Width Modulation*) qui donne l'intensité du courant et sa direction. La ligne *shutdown* peut servir, par

exemple, à éviter des déclenchements intempestifs à l'application de la tension d'alimentation ou pour faire passer le LM381x en sommeil lorsque l'on choisit le mode économie d'énergie. Tous les circuits de cette série sont disponibles pour un courant max de ± 1 A (-1.0) ou ± 7 A (7.0).

Application typique :

Fusible électronique, Elektor 10/2002



Types et boîtiers :

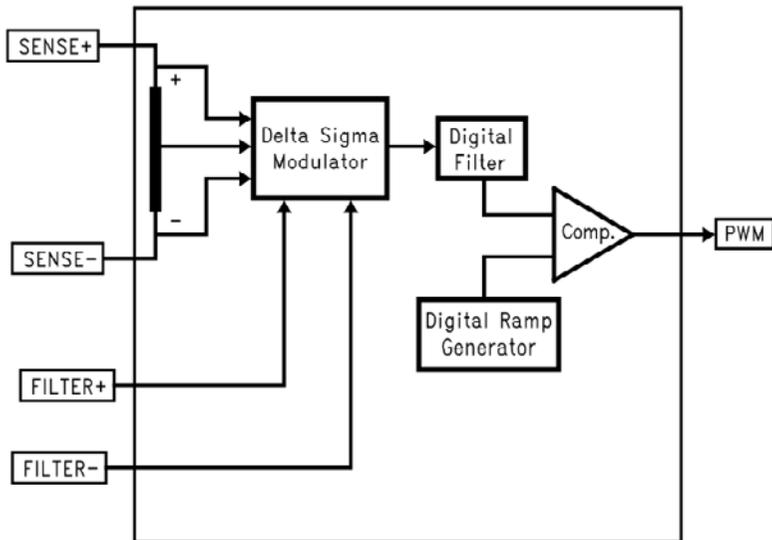
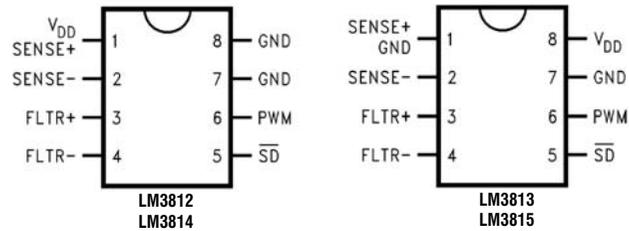
Dénomination	Plage	Intervalle	Méthode
LM3812M-1.0	± 1 A	50 ms	High-side
LM3812M-7.0	± 7 A	50 ms	High-side
LM3813M-1.0	± 1 A	50 ms	Low-side
LM3813M-7.0	± 7 A	50 ms	Low-side
LM3814M-1.0	± 1 A	6 ms	High-side
LM3814M-7.0	± 7 A	6 ms	High-side
LM3815M-1.0	± 1 A	6 ms	Low-side
LM3815M-7.0	± 7 A	6 ms	Low-side

Tous ces types sont proposés en boîtier SO-8

Brochage et structure interne :

Broche

- | | | | |
|---|-----------------|---|---|
| 1 | SENSE+ | Connexion chaude (<i>high-side</i>) du shunt interne (VDD/GND)* | (au positif de la tension d'alimentation sur les LM3812/14, à la masse sur les LM3813/15) |
| 2 | SENSE- | Connexion froide (<i>low-side</i>) du shunt interne | |
| 3 | FLTR+ | Entrée du filtre | |
| 4 | FLTR- | Entrée du filtre | |
| 5 | \overline{SD} | Shutdown (actif au niveau bas) | |
| 6 | PWM | Sortie MLI (PWM) | |
| 7 | GND | Masse | |
| 8 | GND/VDD | Positif de la tension d'alimentation sur les LM3813/15, masse sur les LM3812/14 | |



Caractéristiques électriques: tension d'alimentation 5,0 V, C de couplage = 1 μ F et C de filtrage = 100 nF, $T_j = -40$ à $+125$ °C

Symbole	Paramètre	Conditions	Min.	Typ.	Max.	Unité
IACC	Précision du courant moyen	LM3812-1.0/3813-1.0, à 0,9 A	0,864	0,9	0,936	A
		LM3812-7.0/3813-7.0, à 2,5 A	2,350	2,5	2,650	A
		LM3814-1.0/3815-1.0, à 0,9 A	0,850	0,9	0,950	A
		LM3814-7.0/3815-7.0, à 2,5 A	2,288	2,5	2,712	A
en	Bruit de sortie effectif	LM3812-1.0/3813-1.0		2		mA
		LM3812-7.0/3813-7.0		20		mA
		LM3814-1.0/3815-1.0		12		mA
		LM3814-7.0/3815-7.0		120		mA
I_{Q1}	Courant de repos	Fonctionnement normal, $\overline{SD} = \text{High}$		100	160	μ A
I_{Q2}	Courant de repos	Fonctionnement shutdown, $\overline{SD} = \text{Low}$		2,5	10	μ A
D_{RES}	Résolution MLI	LM3812/3813		0,1		%
		LM3814/3815		0,8		%
t_s	Intervalle d'échantillonnage	LM3812/3813	40	52	80	ms
		LM3814/3815	4	6	10	ms
f_p	Fréquence MLI	LM3812/3813	12,5	20	25	Hz
		LM3814/3815	100	160	250	Hz
V_{TH}	Valeur de seuil supérieure pour \overline{SD}		1,8	1,2		V
V_{TL}	Valeur de seuil inférieure pour \overline{SD}			1,3	0,7	V
V_{OH}	Niveau haut du signal MLI	Courant de charge 1 mA, $2 V \leq V_{DD} \leq 5,25 V$	$V_{DD} - 0,2$	$V_{DD} - 0,05$		V
V_{OL}	Niveau bas du signal MLI	Courant de charge -1 mA, $2 V \leq V_{DD} \leq 5,25 V$		0,04	0,2	V
P_1	Résistance de shunt	$I_{SENSE} = 1 A$		0,004		Ω

Capacimètre

À timer CMOS

Projet : Maximilian A. Lange

Un appareil de mesure simple destiné à élucider les inscriptions cryptiques quasiment illisibles que l'on trouve sur les condensateurs modernes.



La miniaturisation des composants électroniques n'a pas que des avantages. Si, il y a quelques années à peine par exemple, il y avait sur les condensateurs suffisamment de place pour y indiquer la valeur, la tolérance, le fabricant, la polarité et bien d'autres informations encore, on ne découvre plus aujourd'hui qu'un code vague s'il n'est pas illisible indiquant la provenance du composant. La seule option restant alors est de mesurer chaque composant avant son implantation. Cette vérification n'implique pas l'utilisation d'un instrument de précision, on pourra se contenter, pour déterminer approximative-

ment une capacité, d'un instrument simple et bon marché tel le capacimètre proposé ici dans la série « mini-projet ».

Une paire de temporisateurs

Un coup d'oeil sur le schéma du capacimètre de la **figure 1** permet de découvrir une paire de 555C. Il s'agit d'un circuit temporisateur (*timer*) tout ce qu'il y a de plus classique, compatible broche à broche et

fonctionnellement avec le NE/LM555 qu'il n'est sans doute plus nécessaire de vous présenter. Le **tableau 1** en donne la table de vérité. La différence majeure se situe au niveau du suffixe « C » : la consommation de courant des entrées **THRESHOLD**, **RESET** et **TRIGGER** est, avec ses quelques pico-ampères, qu'elle n'a pratiquement pas d'effet sur le processus de charge du condensateur.

Les 2 temporisateurs, dont on retrouve le synoptique de la structure interne en **figure 2**, travaillent selon 2 modes différents, IC1 en tant que multivibrateur astable alors que IC2 travaille en multivibrateur monostable. IC1 génère à sa sortie (broche 3) des brèves impulsions négatives de 25 μ s espacées de 65 ms environ. Ces impulsions ont pour fonction de déclencher le second temporisateur, IC2, de manière à démarrer un processus de mesure.

C'est le temporisateur IC2 qui est chargé de la mesure de la capacité inconnue. Au « repos », l'entrée de décharge (*DIScharge*, broche 7) constitue une connexion à la masse et décharge ainsi le condensateur dont on veut connaître la capacité. Le processus de mesure démarre à l'arrivée du signal de déclenchement (*trigger*) sur l'entrée de déclenchement (*TRigger*, broche 2). Le FET de décharge (*DISCHARGE* de la figure 1) bloque alors et le condensateur débute sa charge au travers de la résistance (R2 à R7) à laquelle il se trouve connecté par le biais du rotacteur S1 (cf. **tableau 2**). La fin de l'im-

pulsion de déclenchement reste sans effet, la sortie et le FET de décharge gardent leur état tant que la tension aux bornes du condensateur et de ce fait celle appliquée à l'entrée de seuil (*THReshold*) reste en deçà des 2/3 de la tension d'alimentation.

Au bout de $t_H = 1,1 \cdot R \cdot C_X$ ce niveau de seuil est atteint. Ceci a pour effet de faire basculer la sortie vers un niveau bas et la ligne de décharge (DIS) redécharge le condensateur.

Les choses restent à nouveau figées jusqu'à l'arrivée de l'impulsion de déclenchement suivante. Ce processus se répète jusqu'à ce que l'on coupe l'alimentation de l'appareil de mesure. Le rapport impulsion/pause, que l'on appelle aussi le rapport cyclique, du signal est en relation directe avec la capacité du condensateur.

Nous avons vite fait le tour du reste de l'électronique. Pour éviter de nous compliquer la vie nous avons utilisé, pour l'affichage, un galvanomètre à bobine mobile de 100 μA , mais on pourra également utiliser un voltmètre numérique (placé en calibre 2 V).

Pour éviter que l'aiguille de l'instrument de mesure ne batte la mesure au rythme du rapport cyclique, la combinaison R9/C5 lissent le signal. La tension continue correspondant au rapport cyclique est dérivée au curseur de P2. La diode zener D1 force cette tension à un niveau donné de sorte que l'affichage n'est pas influencé par l'état de la pile. Nous savons maintenant tout sur le principe de mesure. Il reste cependant à remplir un certain nombre de conditions si nous voulons obtenir des résultats convenables et, dans une certaine mesure, fiables.

- La durée de période de la mesure (65 ms dans le cas présent) ne doit

SI	R _{Mess}	Plage de mesure*
1	22 M Ω	1 nF
2	2,2 M Ω	10 nF
3	220 k Ω	100 nF
4	22 k Ω	1 μF
5	2,2 k Ω	10 μF
6	220 Ω	100 μF

* dans le cas d'un affichage de MMN, la valeur est toujours le double

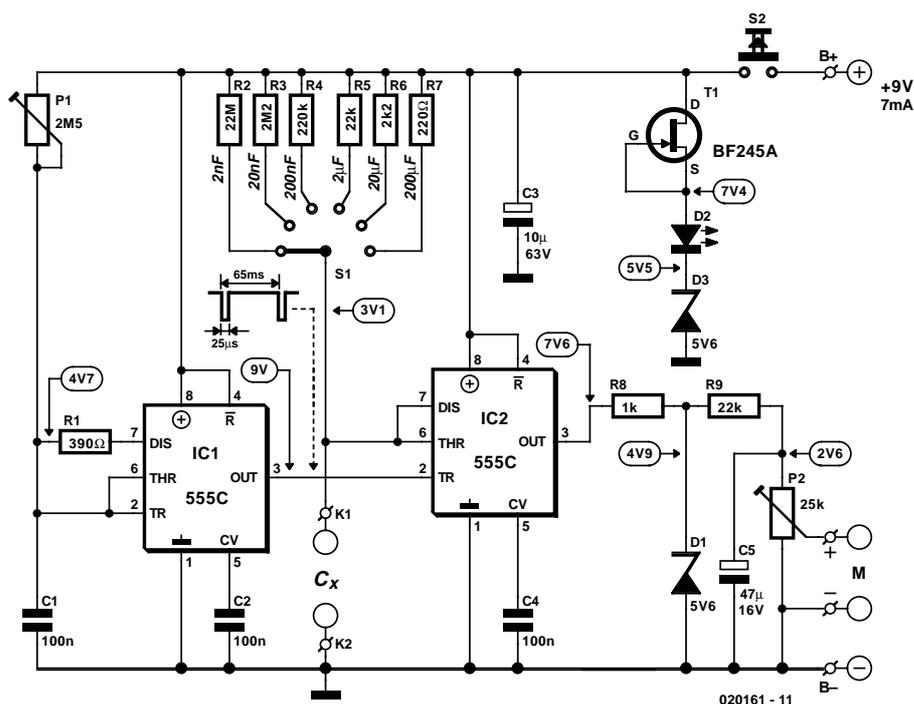


Figure 1. L'électronique d'une extrême simplicité de notre capacimètre repose sur une paire de circuits temporisateurs.

Tableau 1. Table de vérité du 555

THRESHOLD (Broche 6)	TRIGGER (Broche 2)	RESET (Broche 4)	OUTPUT (Broche 3)	DISCHARGE SWITCH (Broche 7)
indifférent	indifférent	bas	bas	activé
> 2/3 · V+	> 1/3 · V+	haut	bas	activé
< 2/3 · V+	< 1/3 · V+	haut	stable	stable
indifférent	< 1/3 · V+	haut	haut	désactivé

pas être trop longue vu que sinon, de par la faible durée relative de l'état haut, la moyenne de la tension de sortie ne serait que très faible, ce qui se traduirait par un

débattement très faible de l'aiguille de l'instrument de mesure.

- Bien plus grave encore, la durée de période ne doit pas être trop courte. On a en effet dans ce cas-là apparition d'une nouvelle

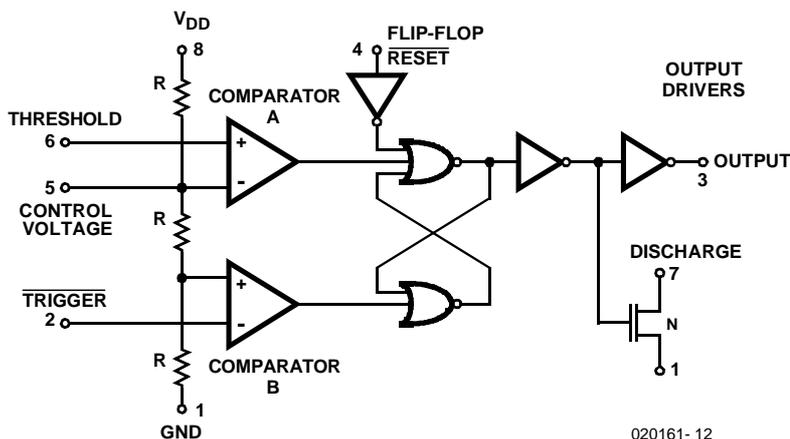


Figure 2. Structure interne du 555.

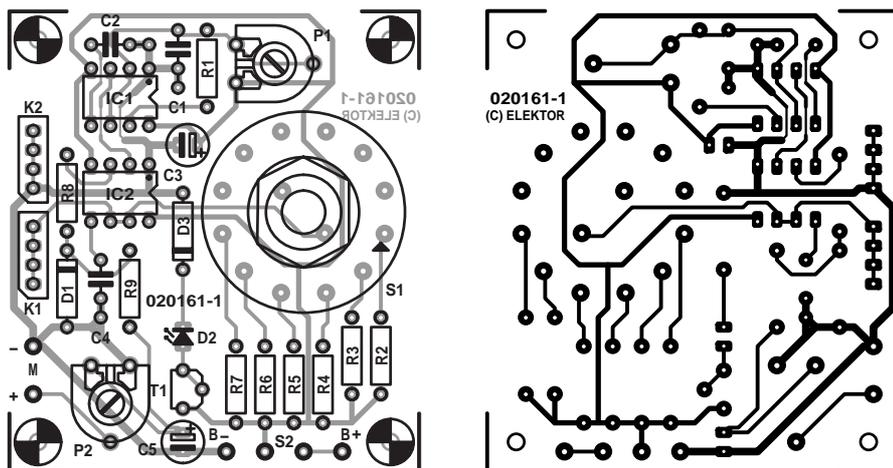


Figure 3. Dessin des pistes et sérigraphie de la platine du capacimètre.

impulsion de déclenchement avant que la mesure ne soit terminée. Cet état de faits se traduit par la perte d'impulsions et partant un résultat de mesure erroné. Vous pouvez provoquer ce type d'erreur en jouant sur P1 de manière à augmenter la fréquence du multivibrateur astable. On verra brusquement l'affichage tomber à la moitié de la valeur affichée précédemment.

– Si les durées de période et de mesure se trouvent bien dans le moins du monde rapport requis mais qu'elles sont trop longues, (si l'on travaille par exemple à de l'ordre des dixièmes de seconde), la paire R8/C5 n'est plus en mesure de lisser suffisamment la tension de sortie pour éviter le tressaille-

ment de l'aiguille de l'instrument.

Relevons un petit détail au passage : T1, D2 et D3 constituent un dispositif rudimentaire de visualisation de l'état de la pile. De par la présence de la diode zener D3, la LED est rehaussée à 7,4 V, de sorte que cette LED s'allume lorsque la pile est encore correctement chargée. Dès que la tension aux bornes de la pile chute en dessous de cette valeur, la LED D2 s'éteint. Le transistor T1 travaille en source de courant classique.

Liste des composants

Résistances :

- R1 = 390 Ω
- R2 = 22 M Ω
- R3 = 2M Ω
- R4 = 220 k Ω
- R5, R9 = 22 k Ω
- R6 = 2k Ω
- R7 = 220 Ω
- R8 = 1 k Ω
- P1 = ajustable 2M Ω
- P2 = ajustable 25 k Ω

Condensateurs :

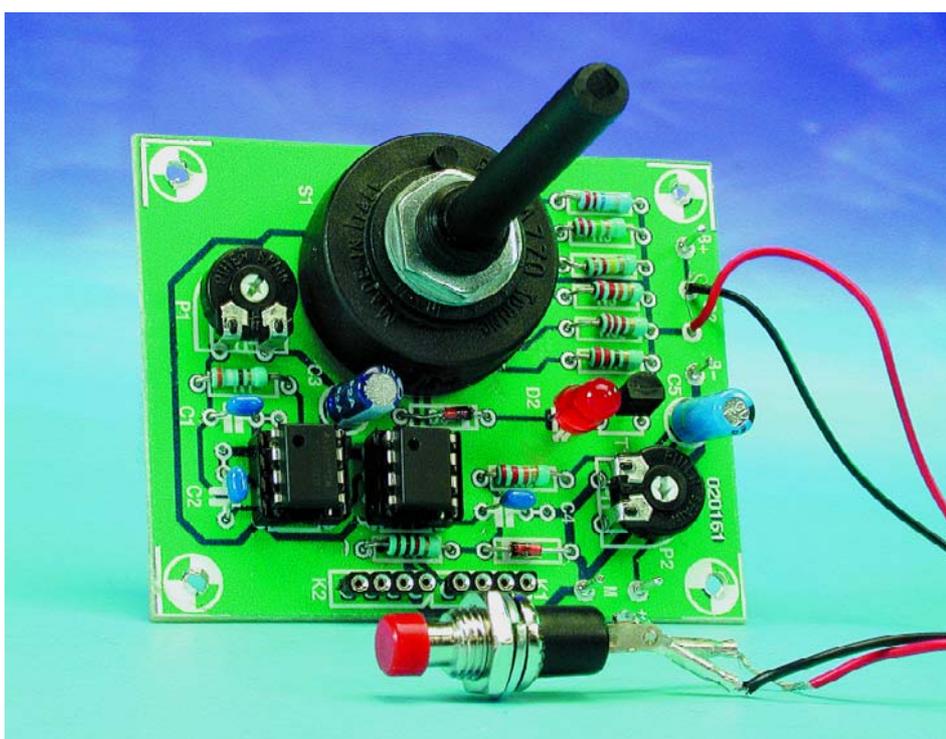
- C1, C2, C4 = 100 nF
- C3 = 10 μ F/63 V axial
- C5 = 47 μ F/16 V axial

Semi-conducteurs :

- D1, D3 = diode zener 5V6/500 mW
- D2 = LED rouge à haut rendement (2 mA)
- T1 = BF245A ou BF256A
- IC1, IC2 = 555C (555 CMOS)

Divers :

- S1 = rotacteur à 2 circuits/6 positions
- S2 = bouton-poussoir unipolaire à contact travail
- K1, K2 = barrette de 4 contacts en tulipe
- Boîtier de 60 x 101 x 26 mm avec compartiment pour pile bouton
- Galvanomètre à bobine mobile 100 μ A



Réalisation, étalonnage, utilisation

La mise en place des composants sur le circuit imprimé, dont on retrouve le dessin des pistes et la sérigraphie de l'implantation des composants en figure 3 est disponible au téléchargement depuis le site Web d'Elektor (www.elektor.fr), n'a rien de bien délicat. Tous les composants (bouton-poussoir, pile et galvanomètre) qui ne prennent pas directement place sur la platine y seront reliés par le biais des picots qu'elle comporte. Vu que la longueur des câbles de connexion exerce une influence très néfaste sur le résultat de mesure, il faudra connecter le condensateur directement aux contacts prévus à cet effet sur la platine. Ces 2 contacts, K1 et K2, prennent la forme phy-

sique de 2 morceaux de barrette autosécable à contacts tulipe (dorés le cas échéant). Comme le montre un coup d'oeil à la platine, l'écartement maximal des contacts est de 20,32 mm. Si ceci ne suffit pas on pourra utiliser pour K2 un morceau plus long d'embase autosécable dont les contacts additionnels auront été reliés au point de contact commun de K2. Il faudra, avant de pouvoir implanter la platine dotée de ses composants et testée dans un coffret que l'on aura pourvu d'une face avant tirée du dessin représenté en **figure 3**, procéder à l'étalonnage du circuit.

Le montage comporte 2 points d'étalonnage. L'ajustable P1, qui permet de jouer sur la fréquence de mesure, est mis en butée vers la gauche, l'ajustable P2 étant lui positionné à mi-course. On connecte ensuite le galvanomètre (ou le multimètre), introduit un condensateur de 100 nF à la tolérance la plus faible possible dans les contacts K1/K2 et on met le rotacteur S1 en calibre 200 nF. Il ne reste plus ensuite qu'à connecter la pile et à appuyer ensuite sur le bouton-poussoir S2. Le débattement de l'aiguille devrait être minime. On tourne

ensuite doucement l'ajustable P1 vers la droite jusqu'à ce que le débattement de l'aiguille de l'instrument diminue brutalement de moitié (nous avons vu plus haut pourquoi !). Ce phénomène observé, nous revenons doucement en arrière jusqu'à atteindre (de l'ordre de 90%) du débattement observé auparavant. Par action sur P2 on règle l'affichage à 100 mA (ou 1,000 V en cas de connexion d'un multimètre en fonction voltmètre). Ceci termine la procédure d'étalonnage.

Comme nous le disions dans l'introduction, cet appareil n'a pas la prétention d'être un instrument de précision. Il connaît inévitablement une erreur de mesure de quelques pour cents. L'utilisation, pour R2 à R7, de résistances de précision n'ayant pas d'effet sensible sur la précision, on pourra se contenter de résistances à tolérance de 1%. La fonction du présent capacimètre est de permettre l'identification de condensateurs de la série E6, dont les facteurs sont 1, 1,5, 2,2, 3,3, 4,7 et 6,8. Chaque valeur vaut 1 fois et demie environ la valeur qui la précède (en fait plus exactement $\sqrt[6]{10} = 1,47$). Dans ces conditions, notre capacimètre devrait être en mesure de déterminer

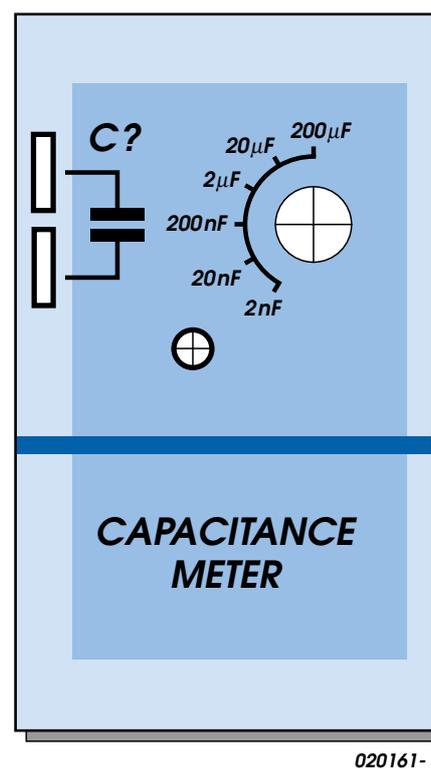


Figure 4. Exemple de dessin de face avant pour le capacimètre.

indubitablement la valeur d'un condensateur de la série-E6.

(020161)

USB pour tous

Partie 2 : pilote à réaliser soi-même

Markus Müller et Christian Ehmer

L'utilisation des pilotes de périphériques a été décrite dans le numéro précédent d'Elektor. Passons à la façon de modifier soi-même un pilote de périphérique de Cypress. Vous n'avez besoin que de quelques programmes pour vous lancer, même si vous n'avez encore jamais travaillé avec Visual Studio de Microsoft. Chaque étape est décrite dans ses moindres détails.

Le Gestionnaire de périphériques de Windows indique sous *Contrôleur USB* les entrées de tous les périphériques USB connectés à cet instant. La **figure 1** montre un périphérique USB équipé de la puce Cypress sans EEPROM et un espion des données USB pour BinTerm.

Il est presque impossible pour un simple mortel de réaliser un pilote de périphérique à partir de zéro. Cypress, heureusement, ne propose pas que la version à installer du pilote *EZ-USB*, mais en offre aussi le code source.

Le pilote de périphérique comporte 2 parties : le fichier INF contenant les informations de configuration et le fichier SYS du pilote proprement dit. Le fichier INF peut être édité avec un simple éditeur de texte. Il faut disposer d'un compilateur C++ pour créer le fichier SYS.

Une méconnaissance totale de la programmation en C ne constitue pas un handicap insurmontable. Vous devez disposer de *Microsoft Visual Studio 6*, du kit de développement de pilotes *DDK (Driver Development Kit)* pour Windows 2000 et bien entendu du code source du pilote de périphérique Cypress.

Installez *Visual Studio* ainsi que les autres modules sans modifier les chemins d'installation ! Pour le pilote de périphérique, il faut disposer du *EZ-USB Development Kit* avec lequel vous avez déjà fait connaissance et que vous avez déjà installé (voir la première partie de l'article).

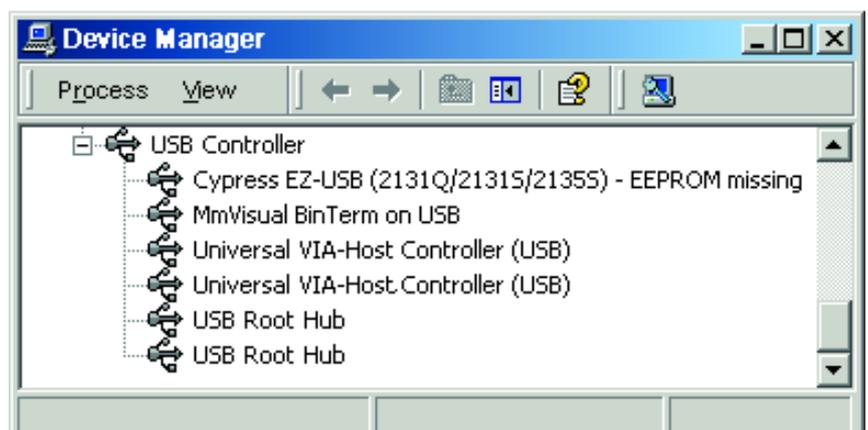


Figure 1. Un pilote « maison » dans le Gestionnaire de périphériques Windows.

Spécifications

Spécifications système :

Windows 2000 ou XP, accès à Internet.

Outils/Programmes

MmVisual BinTerm a/c Version 2.2.2421

Delphi 6 de Borland ou Visual Basic de Microsoft

EZ-USB Development Kit de Cypress Semiconductor

Microsoft Visual C++ 6

Microsoft DDK2000–Driver Development Kit pour Windows 2000

Matériel

Une puce AN2131SC de Cypress fonctionnelle pour USB avec EEPROM ou le montage BinTerm pour USB.

Ce dernier montage vous sera proposé dans l'un des prochains numéros d'Elektor. BinTerm pour USB convient idéalement aux applications expérimentales vu que l'EEPROM ne comporte pas de code de programme. BinTerm charge tous les programmes dans le système lors de l'exécution.

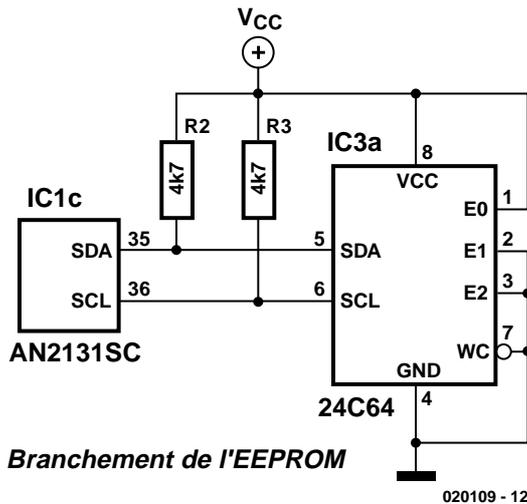


Figure 2. Circuit de l'extension EEPROM.

Informations de configuration

Le fichier INF est indispensable pour l'installation de pilotes de périphériques sous Windows. Il contient les informations indiquant comment, où

et quoi installer. Il faut adapter le fichier à vos propres besoins pour éviter que le pilote de périphérique de Cypress entre en conflit avec le votre. Les parties à modifier sont indiquées en gras dans le **tableau 1**. Il faut tout d'abord rebaptiser le

fichier INF. Introduisez le nouveau nom sous la rubrique [xxx.Files.Inf]. Remplacez le nom de l'entreprise Cypress par le votre. La rubrique [Strings] contient les désignations affichées par le Gestionnaire de périphériques de Windows. Le pilote de périphérique que vous avez réalisé doit aussi se trouver dans un autre fichier SYS. Baptisez-le autrement, par exemple MYSYS.SYS. Remplacez toutes les références à ce fichier dans EZUSB par « MYSYS ». Attention ! Le pilote de périphérique ne doit contenir ni caractères français spéciaux ni espaces. Donc, EXCLUSIVEMENT 8 lettres ordinaires et chiffres au maximum.

Les numéros VID et PID sont particulièrement importants. Ils doivent être compris entre 1 et 65534 et représentés en hexadécimal. L'organisation www.usb.org permet à chacun (moyennant finance) de louer un numéro VID. Le produit peut être alors pourvu d'un symbole USB et le numéro est garanti unique. On ne peut cependant pas parler de location gratuite si vous voyez ce que nous voulons dire...

Les VID et PID de notre solution sont « inventés », ce qui pourrait provoquer des conflits avec les appareils d'autres fournisseurs. Attri-

Tableau 1

```
[Version]
provider=%Cypress%
[Manufacturer]
%Cypress%=Cypress
[Cypress]
%USB\VID_0547&PID_2131.DeviceDesc%=EZUSB.Dev, USB\VID_0547&PID_2131
[DestinationDirs]
EZUSB.Files.Ext = 10,System32\Drivers
EZUSB.Files.Inf = 10,INF
[EZUSB.Dev]
CopyFiles=EZUSB.Files.Ext, EZUSB.Files.Inf
AddReg=EZUSB.AddReg
[EZUSB.Dev.NT]
CopyFiles=EZUSB.Files.Ext, EZUSB.Files.Inf
AddReg=EZUSB.AddReg
[EZUSB.Dev.NT.Services]
Addservice = EZUSB, 0x00000002, EZUSB.AddService
[EZUSB.AddService]
DisplayName = %EZUSB.SvcDesc%
ServiceBinary = %10%\System32\Drivers\ezusb.sys
[EZUSB.AddReg]
HKR,,NTMPDriver,,ezusb.sys
[EZUSB.Files.Ext]
ezusb.sys
[EZUSB.Files.Inf]
ezusbw2k.inf
[Strings]
Cypress="Cypress Semiconductor"
USB\VID_0547&PID_2131.DeviceDesc="Cypress EZ-USB (2131Q/2131S/2135S) - EEPROM missing"
```



Figure 3. Test du pilote de l'appareil avec BinTerm.

buez 8B16 au numéro VID et A001 au numéro PID.

VID (Vendor ID), PID (Product ID) et DID (Device ID) sont 3 mots de données de 16 bits par lesquels la puce Cypress s'annonce au système d'exploitation. Le pilote est recherché, installé et chargé sur la base des numé-

ros VID et PID.

Il vous faut une EEPROM pour employer vos propres VID et PID avec la puce Cypress. La **figure 2** contient l'extension du circuit de la partie 1 permettant de « voir » une EEPROM de plus de 8 Koctets. La

séquence de nombres suivante sera gravée dans l'EEPROM à partir de l'adresse 0 :

```
B0h      Octet initial
          Reprise des ID
16h, 8Bh VID = 8B16h
01h, A0h  PID = A001h
01h, 00h DID = 0001h
```

Lorsque le périphérique USB est raccordé, la puce Cypress s'annonce au système d'exploitation avec la nouvelle identification pour que Windows installe le pilote correspondant.

Tableau 2

```
NTSTATUS DriverEntry( IN PDRIVER_OBJECT DriverObject, IN PUNICODE_STRING RegistryPath)
{
    NTSTATUS ntStatus = STATUS_SUCCESS;
    PDEVICE_OBJECT deviceObject = NULL;
    : :
    DriverObject->MajorFunction[IRP_MJ_DEVICE_CONTROL] = Ezusb_ProcessIOCTL;
    : :
    DriverObject->DriverExtension->AddDevice = Ezusb_PnPAddDevice;
    : :
}
```

Tableau 3

```
NTSTATUS Ezusb_CreateDeviceObject(IN PDRIVER_OBJECT DriverObject,
                                IN PDEVICE_OBJECT *DeviceObject, LONG Instance)
{
    NTSTATUS ntStatus;
    WCHAR deviceLinkBuffer[] = L"\\DosDevices\\Ezusb-0";
    UNICODE_STRING deviceLinkUnicodeString;
    WCHAR deviceNameBuffer[] = L"\\Device\\Ezusb-0";
    : :
    deviceLinkBuffer[18] = (USHORT) ('0' + Instance);
    deviceNameBuffer[14] = (USHORT) ('0' + Instance);
}
```

Tableau 4

```
NTSTATUS Ezusb_ProcessIOCTL(IN PDEVICE_OBJECT fdo, IN PIRP Irp)
{
    : :
    switch (ioControlCode)
    {
        case IOCTL_Ezusb_VENDOR_REQUEST: // = $00222014
            length = Ezusb_VendorRequest (fdo, (PVENDOR_REQUEST_IN) ioBuffer);
            if (length)
            {
                Irp->IoStatus.Information = length;
                Irp->IoStatus.Status = STATUS_SUCCESS;
            } else
            {
                Irp->IoStatus.Status = STATUS_SUCCESS;
            }
            break;
    }
```

Tableau 5

```

case IOCTL_EZUSB_GET_DRIVER_VERSION:
{
    PEZUSB_DRIVER_VERSION version = (PEZUSB_DRIVER_VERSION) ioBuffer;
    if (outputBufferLength >= sizeof(EZUSB_DRIVER_VERSION))
    {
        version->MajorVersion = EZUSB_MAJOR_VERSION;
        version->MinorVersion = EZUSB_MINOR_VERSION;
        version->BuildVersion = EZUSB_BUILD_VERSION;
        Irp->IoStatus.Status = STATUS_SUCCESS;
        Irp->IoStatus.Information = sizeof(EZUSB_DRIVER_VERSION);
    } else
    {
        Irp->IoStatus.Status = STATUS_UNSUCCESSFUL;
    }
}
break;

```

Tableau 6

```

typedef struct _VENDOR_REQUEST_IN
{
    BYTE    bRequest;
    WORD    wValue;
    WORD    wIndex;
    WORD    wLength;
    BYTE    direction;
    BYTE    bData;
} VENDOR_REQUEST_IN, *PVENDOR_REQUEST_IN;

```

Modification des pilotes de périphériques SYS en C++

Démarrons *Visual C++ Studio*, choisissons l'option *Ouvrir un espace de travail* dans le menu *Fichier* et ouvrons le fichier *EZUSB.DSW*. Vérifions ensuite s'il est possible de créer le projet. Choisissez *Créer ezusb.sys* dans le menu *Créer (F7)*. Si tout se passe bien, le message *0 erreur(s), 0 avertissement(s)* apparaît dans la boîte de dialogue. Si les chemins d'installation ont été modifiés lors de l'installation, il faut modifier tous ceux qui se trouvent dans les options du compilateur et de l'éditeur de liens.

L'environnement est correctement configuré lorsque le code est créé

sans erreurs ni avertissements. Il est pratiquement impossible de changer le nom de l'ensemble du projet sous C, ce pourquoi nous n'y toucherons pas. Le fichier *SYS* créé sera baptisé à la fin *MYSYS.SYS*.

Les fichiers individuels

Le fichier le plus important se nomme *ezusbsys.c*; il contient l'intégralité du code de communication entre le pilote de périphérique USB sous Windows et l'application. Il n'est pas possible de traiter du code source C de Cypress dans son intégralité.

Le système d'exploitation charge le fichier *SYS*, puis des pointeurs sont placés au point d'entrée (*DriverEntry*) existant dans chaque pilote de

façon à permettre au système d'exploitation de sauter aux autres fonctions de l'utilisateur. *Ezusb_PnPAddDevice* permet au système d'exploitation de créer un nouvel appareil. Cette fonction appelle *Ezusb_CreateDeviceObject* pour définir le nom virtuel de l'appareil (comme COM1 ou LPT1).

La fonction *Ezusb_ProcessIOCTL* assume la tâche la plus importante : elle saute à la fonction *DeviceIoControl* de *Kernel32.dll* lors de chaque appel.

Le nom de l'appareil peut être modifié à volonté comme dans le **tableau 3**. Changez *Ezusb* en *Mysys*. Comme les 2 noms sont d'égale longueur, il est inutile de modifier les indications de longueur 18 et 14.

Le programme réalisé dans la partie précédente peut établir la liaison sous le nom `\\.\Mysys-0` au moyen de *CreateFile* dès que ce pilote a été chargé par le système d'exploitation. Le pilote peut théoriquement traiter jusqu'à 8 appareils du même type. C'est la raison pour laquelle le nom se termine par « - » suivi d'un chiffre.

Le sous-programme communiquant avec l'appareil USB (**tableau 4**) est appelé lors de chaque commande *DeviceIoControl*. *Ezusb_VendorRequest* exécute la commande « Request » destinée à l'appareil USB... mais il vaut peut-être mieux nous arrêter ici.

Toutes les commandes pouvant être exécutées par le pilote de l'appareil se trouvent dans l'instruction *switch (ioControlCode)*. Même le contrôle de la version est effectué

Tableau 7

```

#define Ezusb_IOCTL_INDEX 0x0800
#define IOCTL_Ezusb_VENDOR_REQUEST CTL_CODE(FILE_DEVICE_UNKNOWN,
        Ezusb_IOCTL_INDEX+5, METHOD_BUFFERED, FILE_ANY_ACCESS)

```

Liens

Cypress Semiconductor	www.cypress.com
MmVisual BinTerm	www.mmvisual.de
Keil Software	www.keil.com
Borland Delphi	www.borland.de
	www.borland.com
Microsoft	www.microsoft.com
USB- Organisation	www.usb.org
Braintechology	www.braintechology.de
	(achats de puces Cypress)

par DeviceIoControl comme le montre le **tableau 5**. Les informations *Version principale*, *Version secondaire* et *Numéro de build* (*Major Version*, *Minor Version*, *Build Number*) sont contenues dans le fichier *Version.h* et peuvent être modifiées à votre gré.

L'exemple de programme Delphi de la première partie contient une déclaration du type

de l'enregistrement

```
_VENDOR_REQUEST_IN.
```

Ce type est défini dans le fichier *ezusbsys.h* de Cypress (**tableau 6**). Ce fichier et les autres structures de la commande DeviceIoControl peuvent être repris directement en C et, adaptés en conséquence, en Visual Basic ou Delphi.

Toutes les commandes de *DeviceIoControl* sont définies plus bas dans le même fichier (**tableau 7**). La définition est obtenue au moyen de la macro *CTL_CODE*. Consultez de préférence *Aide* pour savoir quels nombres sont combinés et de quelle façon. Pour vérifier manuellement l'index : Le premier code commence par \$00222000, chacun des suivants est incrémenté de \$4. Par conséquent,

```
IOCTL_Ezusb_VENDOR_REQUEST
```

= \$00222014.

Compilez le code source avec F7. Changez en *MYSYS.SYS* le nom du fichier *EZUSB.SYS* produit par la compilation. Copiez le fichier INF modifié et *MYSYS.SYS* sur une disquette. Vous voilà en possession d'une disquette de pilote pour puce Cypress avec les numéros VID et PID que vous avez attribués.

Raccordez au PC l'appareil USB avec les identifications VID et PID correspondantes. Windows identifie automatiquement un nouvel appareil.

Testez le pilote de l'appareil avec BinTerm (**figure 3**). Vous pouvez enregistrer le nom virtuel de l'appareil sous l'onglet USB-Test et vérifier que la communication avec la puce Cypress s'effectue correctement.

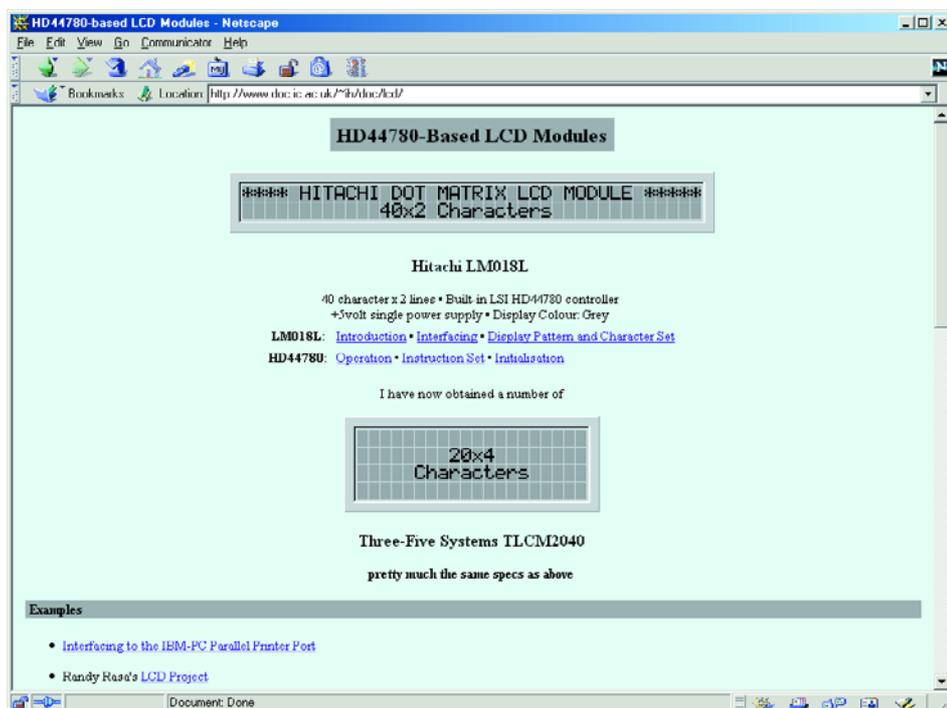
(020109-II)

Les affichages LCD

Détails techniques et applications

Harry Baggen

De plus en plus de montages électroniques, ceux à base de microcontrôleurs en particulier, utilisent des petits affichages LCD alphanumériques. Leur prix a sensiblement chuté et leur commande est devenue chose courante. Leur utilisation implique cependant que l'on en connaisse le brochage et les autres caractéristiques techniques. On trouve heureusement sur Internet une mine de données concernant les affichages LCD, sans même parler de leurs applications.



Dès que l'on envisage la réalisation du moindre système à microcontrôleur, d'un terminal ou de tout autre montage dit « intelligent » on est tenté d'utiliser, pour l'affichage (si tant est qu'il soit nécessaire), comme solution standard, un affichage alphanumérique. Cela n'a rien de bien étonnant vu que ces composants sont devenus abordables, relativement faciles à connecter et à piloter. La plupart des petits modules LCD disponibles sur

le marché utilisent le même contrôleur, peu importe qu'il s'agisse d'un affichage à 1, 2 voire 4 lignes de 8, 16, 20, 24, 32 voire jusqu'à 40 caractères même. Tous ces affichages comportent un contrôleur HD44780 de Hitachi ou son équivalent. Le site **Hitachi** [1] propose la fiche de caractéristiques de ce composant au format .pdf. Vu cependant l'universalité

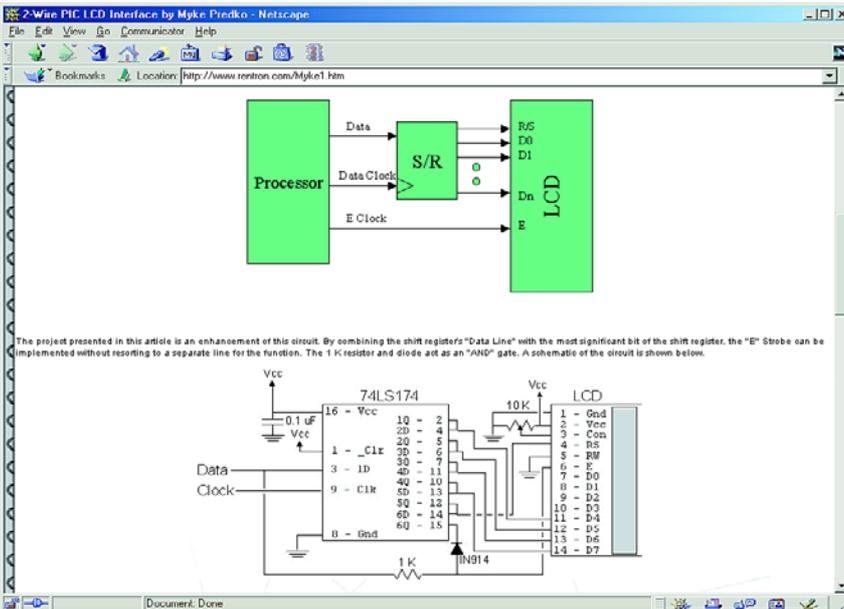
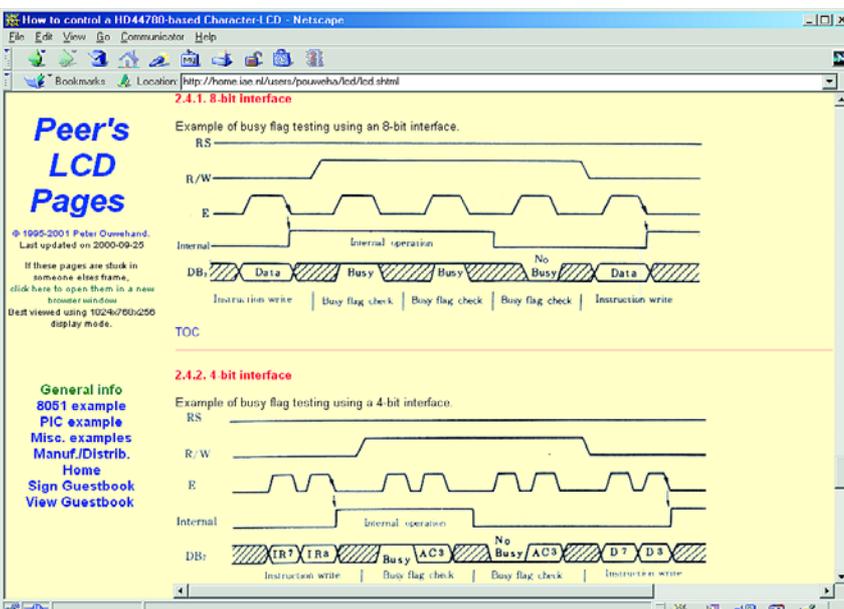
de ces affichages, nombre d'autres sites s'intéressent à eux.

La page **HD44780-Based LCD Modules** [2] de Ian Harris constitue un bon point de départ pour une quête d'informations. Il s'intéresse dans le détail aux affichages à 2 et 4 lignes, passant en revue leur interfacement, les motifs affichables et le set de caractères. Il propose en outre des liens vers des exemples d'applications et d'autres sources d'information.

Lampex [3], un fabricant de toutes sortes d'affichages LCD est une autre bonne adresse où trouver des informations concernant ce contrôleur, le brochage des affichages utilisant le HD44780 ainsi que le rétro-éclairage.

Si vous cherchez des informations sur d'autres types d'affichages non standard, nous ne pouvons que vous recommander de jeter un coup d'oeil à la page LCD de **EIO** (*Electronics Information Online*) [4]. On y trouve les fiches de caractéristiques et les spécifications de la plupart des fabricants de LCD. Le pilotage d'un affichage à base de 44780 n'a rien de bien sorcier surtout une fois que l'on aura lu quelques-unes des documentations proposées sur Internet.

Les **Peer's LCD-Pages** de Peter Ouweland [5] expliquent dans le

détail la commande d'affichages LCD à base de 44780. On y trouve même des exemples d'application pour le 8051 et le PIC.

Myke Predko [6] propose, sous le titre « **Build your own 2-Wire LCD Interface using the PIC16C84 microcontroller** » une autre application intéressante. Comme le suggère le titre, il s'agit d'une sorte de commande sérielle d'un affichage LCD par le biais d'un PIC. Autre site méritant une mention, la page **LCD Interfacing Reference** [7] de Myke.

Si vous voulez connecter un LCD à votre ordinateur vous pourrez le faire, par exemple, par le biais de son port imprimante. Le site **Beyond Logic** [8] de Craig Peacock en donne, parmi bien d'autres, un exemple (avec code-source).

Si vous ne vous sentez pas encore très à l'aise et que votre expérience du sujet n'est que très limitée, vous pourrez trouver une description exhaustive accompagnée de nombreuses illuminations d'un affichage LCD connecté par le biais d'un câble pour imprimante sur le site « **Build your own printer cable LCD Display** » [9] de Overclockers Australia. Un programme Windows bien clair permet de procéder au paramétrage de l'affichage LCD depuis son PC.

(025067)

Adresses Internet

- [1] Hitachi: <http://semiconductor.hitachi.com/hd44780.pdf>
- [2] HD44780-Based LCD Modules: www.doc.ic.ac.uk/~ih/doc/lcd/
- [3] Lampex: www.lampex.com/prod.htm
- [4] EIO LCD-pagina: www.eio.com/dataset.htm
- [5] How to control a HD44780-based Character-LCD: <http://home.iae.nl/users/pouweha/lcd/lcd.shtml>
- [6] Build your own "2-Wire LCD Interface" using the PIC16C84 microcontroller: www.rentron.com/Myke1.htm
- [7] LCD Interfacing Reference Page: www.myke.com/lcd.htm
- [8] Beyond Logic: www.beyondlogic.org/parlcd/parlcd.htm
- [9] Build your own printer cable LCD Display: http://www.overclockers.com.au/techs-tuffa_diy_lcd/

Joystick N64 au PC

Réactualisation pour jeux actuels

Kurt Schuster

Pas tous les jeux les plus récents s'accommodent du timing du logiciel original du N64C²PC, l'adaptateur permettant d'utiliser une manette de Nintendo pour piloter des jeux sur PC [1]. Cette réactualisation (*update*) du logiciel évite que l'image ne saute, frétille voire disparaisse purement et simplement.

La pierre d'achoppement est la synchronisation du microcontrôleur avec le rythme d'échantillonnage utilisé par le port de manette de jeux (le fameux *joystick port*) du PC. Nous avons besoin, pour démarrer au bon moment et dans les plus brefs délais, des 2 instructions `jnb JOYAX,*` et `jnb JOYAY,*` de la boucle principale. En effet, la version originale ne tenait compte que du signal JOYAX, ce qui se traduisait souvent par un démarrage précoce des routines de temporisation, décalage qui se visualisait sous la forme d'un écran quelque peu instable (au niveau d'un réticule ou d'un personnage par exemple). En raison du même problème, nous avons positionné, en amont de ces instructions, les boucles d'attente `jb JOYAX,*` et `jb JOYAY,*`, destinées à permettre d'attendre le début parfaitement défini d'un cycle d'échantillonnage. Une fois que l'on a reconnu indubitablement le début d'un tel cycle, les lignes de manche JOYAX, JOYAY, JOYBX et JOYBY sont forcées au niveau logique bas. Ceci a pour effet d'empêcher la charge de la composante C de paramétrage de temps dans le PC et de faire exécuter, par le programme, la routine `servicetime`, qui interroge le contrôleur du Nintendo 64 qui calcule les valeurs pour le manche A et le manche B et transmet au PC l'état du bouton de joystick. La nouveauté est ici que la valeur analogique prise en compte n'est pas, comme cela était le cas avec la version d'origine du programme, convertie en une valeur 16 bits signée, ce qui est une opération complexe et de longue durée, mais transformée immédiatement en une valeur de 8 bits. Cette opération n'a pas

Déplacement 100%	Bouton L et bouton haut de C (paramétrage par défaut)
Déplacement 66%	Bouton L et bouton droit de C
Déplacement 33%	Bouton L et bouton bas de C
Intervention de A et B	Bouton L et bouton gauche de C
C = Combiné C à 4 boutons	

de conséquence néfaste sur la précision vu que de toutes façons le contrôleur ne fournit que des valeurs à 8 bits. De manière à s'assurer que cette durée de « *service* » soit de longueur immuable, on a, parallèlement au déroulement de la routine, fonctionnement d'un temporisateur (*timer*) dont on attend l'écoulement après la routine de « *service* ». Dans la routine suivante, les différentes lignes du port de joystick sont mises, individuellement et en fonction des positions des manches de

commande, au niveau haut. Les condensateurs des multivibrateurs monostables intégrés de l'électro-

L'intervention requiert la suite d'action suivante sur les touches :

1. Appuyer sur le bouton L et le maintenir enfoncé
2. Appuyer sur le bouton C requis
3. Relâcher le bouton L
4. Relâcher le bouton du Combiné

Chronologie des processus d'un cycle d'échantillonnage :

Lignes du joystick PC	Lignes du joystick N64C ² PC
Haut : du cycle précédent	En attente d'un niveau bas
Bas : Décharge des condensateurs	En attente d'un niveau
Haut : Charge des condensateurs	Bas : bloque la charge des condensateurs
Mesure du temps de charge	Service time 250 ms, prendre la valeur au contrôleur, la traiter, fournir l'état du bouton
Mesure du temps de charge	Joytime, entre 10 et 1285 ms, puis mise des lignes au niveau haut et chargement des condensateurs
Mesure du temps de charge	Temps de consolidation de 20 ms environ
Évaluer les durées mesurées	

nique du PC se chargent et fournissent ainsi une valeur de résistance analogique au PC. Comme joytime n'est qu'une simple boucle d'attente sans interruptions, il n'y a aucune raison d'avoir à effectuer de conversion 16 bits. Autre nouveauté, la présence d'une petite boucle d'attente, en aval de joytime, qui permet le plein chargement des composantes C des réseaux.

Il faudra inévitablement, en cas d'utilisation de cette nouvelle version, procéder à une nouvelle calibration du contrôleur. On trouvera de plus amples informations à ce sujet, ainsi

que sur la technologie des ports de manettes du PC, dans l'article cité en référence [1] de la bibliographie.

Autre élément touché par cette réactualisation, le contrôle de déplacement à 3 modes. Il ne suffit plus, comme c'était le cas précédemment, pour calibrer le déplacement obtenu par le biais de la croix, d'appuyer uniquement sur le bouton L du contrôleur (situé sur l'avant à gauche d'où le L de LEFT). Nouveauté : la possibilité d'intervertir, par logiciel, l'unité analogique et la croix de commande. Cette option requiert cependant une recalibration du joystick.

Ce remplacement de programme n'a pas de conséquence sur le paramétrage des fonctions remplies par les différents boutons.

La nouvelle version du logiciel est mise à votre disposition gratuitement sur le site Internet d'Elektor (www.elektor.fr). Si vous n'avez pas d'accès à Internet, vous pouvez la commander sous la forme d'une disquette auprès des adresses habituelles (Publitronic en particulier) sous la dénomination **EPS020288-11**.

(020288)

Bibliographie :

[1] N64C2PC, Elektor n° 260, février 2000, page X-2 et suivantes

MSC1210EVM

Carte d'évaluation pour le MSC1210

Dans le cadre de notre rubrique « Outils de développement » nous vous proposons dans ce numéro une carte d'évaluation sortant des cartons de Texas Instruments, la MSC1210EVM.



La première question que l'on ne manquera sans doute pas de se poser est bien sûr : le MSC1210, qu'est-ce que cela, je n'en ai jamais entendu parler. Le MSC1210 est un CAN (Convertisseur Analogique/Numérique) delta-sigma travaillant avec une précision de 24 bits doté d'un noyau microcontrôleur du type 8051 et de jusqu'à 32 Koctets de mémoire Flash. Il possède 8 canaux d'entrée analogiques différentiels/asymétriques. Son architecture delta-sigma permet à ce composant d'afficher une résolution effective de 22 bits à un taux de données de 10 Hz. Il se laisse programmer à des taux plus importants, jusqu'à 10 kHz, mais cela bien entendu au prix d'une diminution de la résolution effective.

Outre les périphériques et fonctions standard du 8051, le MSC1210 possède un accumula-

teur 32 bits, une interface SPI (in *Site Programming Interface*), une sortie MLI (PWM = *Pulse Width Modulation*) 16 bits, de la mémoire de données en Flash, 1 280 octets de mémoire de données en RAM, 2 UART et une paire de registres DPTR, 34 broches d'Entrée/Sortie (I/O). Un composant puissant comme on le constate.

Le noyau 8051 du MSC1210 a été réactualisé, ce qui lui permet de travailler à 4 cycles d'horloge par cycle-machine. Il est doté de périphériques additionnels tels que chien de garde (*watchdog*), détection de tension de service trop faible (*brownout*), sans oublier une gestion de la consom-

mation de courant et des registres de points d'arrêt matériels. Ce composant est destiné tout d'abord aux applications industrielles et son format TQFP-64 ne le prédestine guère aux expérimentations par l'amateur.

Le kit arrive dans un carton de bonnes dimensions, les différents composants bien amortis par de la mousse. Il comporte tout ce dont on a besoin pour partir à la découverte du MCS1210. Comme l'illustre la photo en début d'article, on y trouve :

– La carte d'évaluation proprement dite avec un MCS1210Y5, une RAM

statique de 128Koctets de Cypress (avec plus de 6,6 millions de transistors !), un CAN, le DAC8531 et d'un champ d'expérimentation

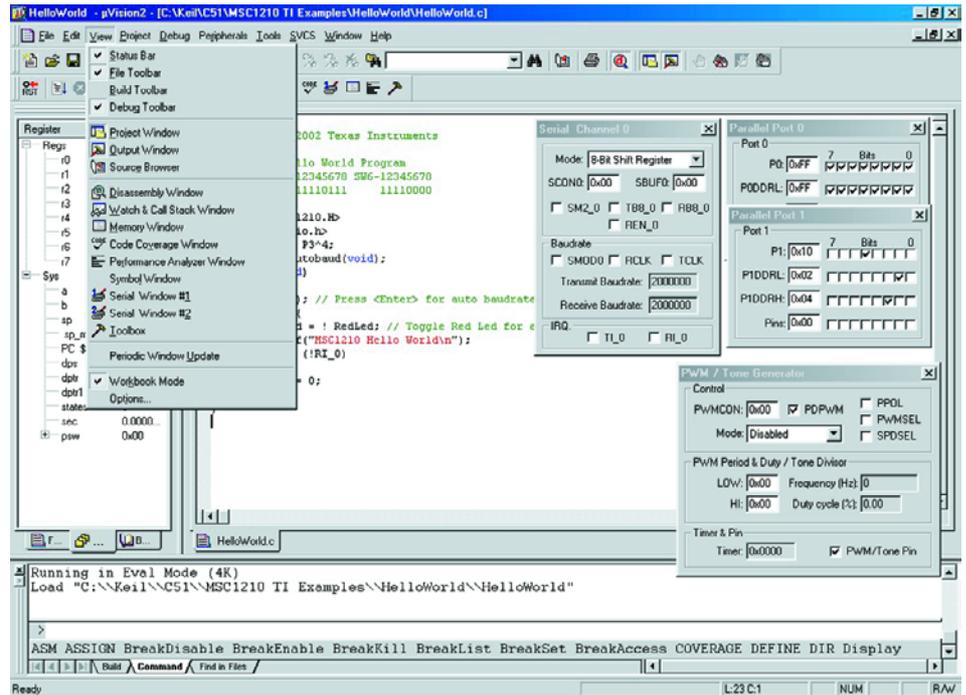
- Un câble sériel
- Une alimentation fort bien pensée puisqu'il suffit d'échanger le connecteur pour la rendre compatible avec les différentes prises secteurs d'Europe et d'ailleurs
- Un petit manuel de 36 pages, le User's Guide
- Un CD-ROM comportant les programmes et logiciels nécessaires à une première évaluation du MSC1210.

Ce CD-ROM s'avère être d'une importance capitale pour apprendre comment mettre ce kit de développement en oeuvre.

La première étape consiste bien évidemment à lire le User's Guide. Il ne donne que les informations les plus importantes concernant l'aspect physique de la carte (cavaliers, caractéristiques techniques, etc.). Une fois que la carte est configurée (en principe on la laisse dans l'état dans lequel elle arrive), il faut installer l'environnement intégré de développement de Keil Software, un IDE fort bien structuré comme le montre la recopie d'écran ci-contre. Après avoir installé C51 et l'outil de téléchargement MSC (MSC Download Tool).

Une partie du CD-ROM comporte une sorte de « leçon de choses » réalisée à l'aide du programme Lotus ScreenCam; on voit ainsi très bien les différentes étapes à suivre avant que l'on n'ait transféré un programme vers la mémoire de la carte. Ne serait-ce que pour le cours consacré au 8051 et 8052, ce CD-ROM vaut déjà la peine d'être parcouru.

Il faut quelques essais pour saisir le principe de la compilation et du téléchargement. Attention à ne pas faire d'erreur au niveau de l'embase



Sub-D à utiliser ; c'est J5, l'embase qui se trouve le plus près du bord, qui sert à la communication entre le PC et la carte d'évaluation. Le transfert en fin de processus « Rebuilt all target files » vers le MSC1210, opération qui se fait normalement automatiquement, est visualisé par un écran visualisant le taux de transfert et le nombre de lignes et la situation instantanée. On notera que le taux de transfert maximal supporté est de 57 600 bauds.

Le domaine du site Internet de Texas Instruments consacré au MSC1210 sis à l'adresse :

<http://focus.ti.com/docs/prod/folders/print/msc1210.html>

met à disposition une quantité impressionnante d'informations et de documents à télécharger, dont un fichier .pdf de pas moins de 58 pages consacré au MSC1210. Signalons cependant, l'existence sur Internet d'une version différente de certains

des matériels accompagnant la carte d'évaluation. Ainsi, si le User's Guide accompagnant la carte d'évaluation est le SBAU073, la version sur le site est baptisée SBAU083 et date de juillet 2002 ; notons qu'elle concerne un autre modèle de carte, la MSC1210-DAQ-EVM. Il y est d'ailleurs fait référence au document cité 6 lignes plus haut. Notons que la carte d'évaluation peut également travailler avec le RIDE (*Raisonance Integrated Development Environment*) de Raisonance, téléchargeable à l'adresse : <http://www.raisonance.com/Download/index.php>.

La version actuelle de RIDE est la 6.4.27 – BN 727 du 30-08-2002. Attention, il s'agit d'un fichier de près de 25 Moctets.

La combinaison RIDE + fichier SBAU083 offre une excellente plate-forme de démarrage pour quelqu'un qui ne connaît pas toutes les ficelles de l'utilisation d'un tel outil.

En résumé, une carte d'évaluation destinée au (semi-)professionnel qui veut évaluer les possibilités du composant très performant qu'est le MSC1210 de Texas Instruments.

(027143)

Dé à microcontrôleur

La carte 89S8252 Flash en plate-forme de développement

Burkhard Kainka

Le développement et la mise au point d'un programme pour microcontrôleur requièrent l'utilisation d'un système de développement. La carte 89S8252 Flash décrite dans le numéro de décembre 2001 convient fort bien à une telle application. Même si le système-cible est un contrôleur moins performant tel que le AT89C2051, notre carte Flash permet un développement de programme rapide et confortable.

Nous avons choisi, comme exemple de développement, la mise au point d'un dé électronique à base de 89C2051, un autre microcontrôleur de l'écurie Atmel. Avouons-le de but en blanc, un dé électronique n'a rien de bien

révolutionnaire. Il n'en reste pas moins que cette application associe simplicité et facilité de compréhension de sorte que nous la considérons comme un exemple acceptable.

Les différentes étapes du développement pourront ensuite être reportées facilement à des applications plus complexes.

L'aspect matériel

La première étape consiste à connecter les diodes électroluminescentes, les LED, au microcontrôleur. Bien qu'un dé ne sache compter que jusqu'à 6, nous utiliserons les 7 LED classiques qui seront reliées à 7 des lignes du port P1 (cf. le schéma de la **figure 1**). On pourra, en cas d'utilisation de LED à haut rendement (*high efficiency*), augmenter à 470 Ω la valeur des résistances de limitation de courant. Comme nous l'expliquons dans le cours « Microcontrôleurs » les LED dotées de leur résistance de limitation sont reliées à la ligne d'alimentation positive, Vcc. La ligne de port restante sert à l'interrogation d'un bouton-poussoir. On pourra placer une réalisation expérimentale de ce montage soit directement sur le champ à pastilles présent sur la platine, soit mettre sur sa propre petite platine que l'on connectera ensuite à la carte Flash par le biais des embases et des câbles requis. Ceci termine l'étape matérielle de cette réalisation, de

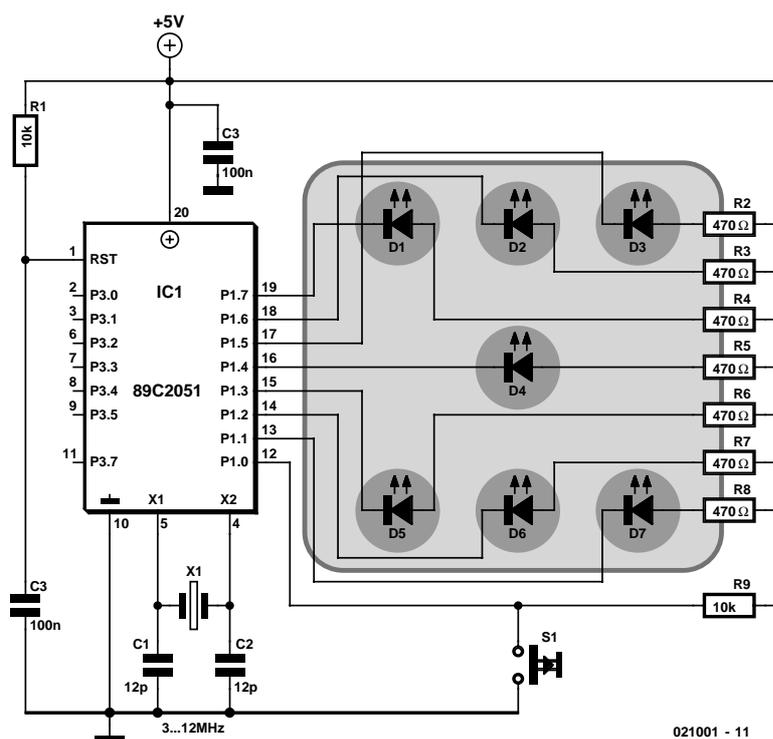


Figure 1. L'électronique du dé et sa connexion à la carte AT89S8252 Flash.

Listage 1. Attribution des positions du patron de visualisation.

```

;microcontroller dice
;
; P1.7          P1.3
;
; P1.6  P1.4  P1.2
;
; P1.5          P1.1
;
;          P1.0 = key

      mov  a,#00010001b ;output "6"
      mov  P1,a
      mov  a,#01000101b ;output "5"
      mov  P1,a
      mov  a,#01010101b ;output "4"
      mov  P1,a
      mov  a,#01101101b ;output "3"
      mov  P1,a
      mov  a,#11010111b ;output "2"
      mov  P1,a
      mov  a,#11101111b ;output "1"
      mov  P1,a
    
```

sorte que nous pouvons maintenant nous intéresser au développement du programme.

Développement de programmes

Nous allons utiliser, comme outil de développement, l'assembleur. Vu la simplicité de la tâche, c'est exactement ce qu'il nous faut. De plus, un programme en assembleur de petite taille ne requiert pas de RAM. Une fois terminé, le programme ne devrait pas poser de problème de fonctionnement sur le contrôleur petit (en ce qui concerne la taille de la mémoire au moins).

La meilleure technique consiste à travailler par petites étapes. On commencera par mettre sur papier le patron de bits servant à la commande des LED. Rien de tel, pour cette tâche, que de faire appel à une représentation binaire. La correspondance entre les 7 LED du dé et les différentes lignes de port est facile à établir. Chaque LED allumée est représentée par un « 0 ». Le bit 0 doit être forcé en permanence au niveau haut sachant que l'on utilise le port P1.0 en tant qu'entrée pour le bouton-poussoir (**listage 1**).

La vérification de la correction de la commande des LED et le programme terminé requièrent une boucle d'attente qui prend ici la forme de la routine (sous-programme) `delay`. La durée d'attente est transférée dans

l'accumulateur (**listage 2**).

Il s'est avéré, en pratique, que cette durée de temporisation était encore trop courte si l'on voulait pouvoir suivre le processus visuellement en temps réel. Ceci explique que nous ayons prévu une seconde routine, `delay2`, dont la durée dépasse légèrement la seconde. La routine `delay2` fait appel, à plusieurs reprises, à la routine `delay` (**listage 3**) et devra elle-même être appelée à l'endroit correct du programme principal. Un petit programme de test (**listage 4**) permet de visualiser lentement successivement tous les patrons d'allumage des LED du dé. Il est possible ainsi de vérifier la connexion correcte des LED. Ce test permet en outre de détecter d'éventuelles erreurs au niveau du paramétrage du patron de bits défini.

Lancer les dés par bouton-poussoir

Le processus d'interrogation du bouton-poussoir tourne dans 2 boucles. Tant qu'il n'y a pas eu d'action sur le bouton-poussoir le programme se contente d'attendre et reste partant inactif. Dès qu'il y a action sur le bouton-poussoir on a incrémentation continue de l'état de compteur attribué au dé. On utilise à cet effet le registre r1 en vue de créer cet état de comptage. le registre r1 ne doit pas prendre d'autre valeur que de 1

à 6. Il faut partant s'assurer à chaque fois que l'on ne se trouve pas en présence du chiffre 7, auquel cas il faudra réinitialiser à 1 l'état du compteur.

L'une des tâches primordiales du programme consiste à faire correspondre les différents patrons d'allumage des LED aux états que peut prendre le compteur. Ce traitement se fait à l'aide d'opérations de comparaisons et de sauts. L'instruction `cjne` (compare & jump if not equal) se charge de ce travail. On compare le contenu du registre r1 successivement aux chiffres 1 à 6. Dès qu'il est établi une égalité on la traduit par la visualisation du chiffre correspondant sur l'affichage. Tant qu'il n'a pas été établi d'égalité on a passage à la comparaison suivante. Cet ordigramme défini, il devient très facile d'écrire, comme le montre le **listage 5**, un premier programme de dé, sans prétention il est vrai.

En principe, l'objectif est atteint. Le dé fonc-

Listage 2. Définition d'une temporisation.

```

delay  mov  r7,a
delay0 mov  r6,#255
delay1 djnz  r6,delay1
      djnz  r7,delay0
      ret
    
```

Listage 3. Allongement de la temporisation.

```

main  acall  delay2
out6  mov   a,#00010001b
      mov  P1,a
out5  acall  delay2
      mov  a,#01000101b
      mov  P1,a
out4  acall  delay2
      mov  a,#01010101b
      mov  P1,a
out3  acall  delay2
      mov  a,#01101101b
      mov  P1,a
out2  acall  delay2
      mov  a,#11010111b
      mov  P1,a
out1  acall  delay2
      mov  a,#11101111b
      mov  P1,a
      sjmp main
    
```

Listage 4. Prototypage de dé en roulement continu.

```

delay2 mov  a,#255
      acall  delay
      acall  delay
      acall  delay
      acall  delay
      ret
    
```

tionne comme il le doit. Chaque action sur le bouton-poussoir se traduit par l'apparition d'un affichage ne cessant de changer pour s'arrêter sur un chiffre quelconque compris entre 1 et 6. Nous avons résumé ici le développement en 3 phases alors qu'en réalité il n'aura pas fallu moins de 20 étapes de développement pour en arriver là.

Il aura fallu à chaque fois changer le texte du code, compiler le programme et le transférer dans le microcontrôleur. Pourquoi a-t-il fallu opérer autant de fois ? Pour la bonne et simple raison que l'on se trouve à chaque fois confronté à l'une ou l'autre petite erreur. Le développement par petits pas a pour effet de permettre d'approcher petit à petit du résultat final. La carte Flash facilite cette stratégie dite du « prototypage rapide » (*rapid prototyping*). Essais et erreurs corrigées amènent, après un bon nombre de petits pas, au résultat recherché. Le test de chaque nouvelle étape de développement ne dure en fait que quelques secondes. Il est révolu le temps où il fallait, pour chaque nouvelle tentative, griller une EPROM...

Petit truc pratique

Les LED de visualisation de chiffre du dé ne font pas que traduire le fonctionnement du dé mais visualisent également le processus de téléchargement. Cela tient au fait que les lignes de port de poids fort, P1.7, P1.6 et P1.5, servent aussi à la programmation du processeur. Au cours de la programmation sérielle en site par le biais de l'interface SPI (*Serial Programming Interface*) et qui se fait selon un processus bien protocolé, l'outil de téléchargement « frétille » sur ces lignes. Ces impulsions sont visibles au niveau des LED. À l'utilisation, cette visualisation « fantôme » s'est avéré très pratique vu qu'elle visualise un téléchargement de programme s'effectuant correctement. En effet, il n'existe pas moins de 3 sources de problèmes possibles :

1. Oubli de la mise sous tension de la carte
2. Interspersion des ports COM1 et COM2
3. Le port COM choisi est déjà occupé par un autre programme.

Les LED du dé signalent en tout cas, lors d'un téléchargement, que tout se passe bien. Il n'y a pas de raison de ne pas s'offrir ce luxe dans le cadre d'autres projets. D'où la suggestion suivante : nous allons placer 2 LED dotées de leur résistance de limitation de courant sur le champ de pastilles. L'une d'entre elles visualisera la présence de la tension d'alimentation, l'autre les signaux véhiculés par la ligne de port P1.7. Grâce à elles on saura toujours où l'on en est.

Listage 5. Première version fonctionnelle du dé.

```
;dice1.asm

;microcontroller dice
;
; P1.7          P1.3
;
; P1.6   P1.4   P1.2
;
; P1.5          P1.1
;
;           P1.0 = key
;

#include 8051.H
        .org 0000H

main    mov     a,#0FFh
        mov     P1,a           ;all leds off
        mov     r1,#1         ;first number
loop    jb     P1.0,loop      ;wait for key
key     mov     a,r1          ;r1 = memory
        inc     a
        mov     r1,a
        cjne   r1,#7,out
        mov     r1,#1
out     cjne   r1,#6,out5
        mov     a,#00010001b
        mov     P1,a
out5    cjne   r1,#5,out4
        mov     a,#01000101b
        mov     P1,a
out4    cjne   r1,#4,out3
        mov     a,#01010101b
        mov     P1,a
out3    cjne   r1,#3,out2
        mov     a,#01101101b
        mov     P1,a
out2    cjne   r1,#2,out1
        mov     a,#11010111b
        mov     P1,a
out1    cjne   r1,#1,out0
        mov     a,#11101111b
        mov     P1,a
out0    jnb    P1.0,key
        sjmp   loop

        .end
```

Listage 6. Programme amélioré avec « roulement progressif ».

```
;dice2.asm

;microcontroller dice
;
; P1.7          P1.3
;
; P1.6   P1.4   P1.2
;
; P1.5          P1.1
;
;           P1.0 = key
;

#include 8051.H
        .org 0000H
```

```

main    mov    a,#0FFh
        mov    P1,a          ;all leds off
        mov    r1,#1        ;first number
loop    jb     P1.0,loop      ;wait for key
key     acall  play          ;r1=r1+r1
        mov    a,#01h
        mov    P1,a          ;all leds on
        mov    a,#5
        acall  delay
        jnb   P1.0,key

        acall  play          ;roll
        mov    a,#10
        acall  delay
        acall  play
        mov    a,#25        ;slower
        acall  delay
        acall  play
        mov    a,#50
        acall  delay        ;slower
        acall  play
        mov    a,#100
        acall  delay        ;slower
        acall  play
        mov    a,#150
        acall  delay        ;slower
        acall  play
        mov    a,#200
        acall  delay        ;slower
        acall  play
        mov    a,#250
        acall  delay        ;slower
        acall  play
        mov    a,#250
        acall  delay        ;last
        acall  delay
        acall  play        ;stop
        sjmp  loop

play    mov    a,r1          ;r1 = memory
        inc    a
        mov    r1,a          ;r1=r1+1
        cjne  r1,#7,out
        mov    r1,#1        ;reset to 1
out     cjne  r1,#6,out5
        mov    a,#00010001b ;output "6"
        mov    P1,a
out5    cjne  r1,#5,out4
        mov    a,#01000101b ;output "5"
        mov    P1,a
out4    cjne  r1,#4,out3
        mov    a,#01010101b ;output "4"
        mov    P1,a
out3    cjne  r1,#3,out2
        mov    a,#01101101b ;output "3"
        mov    P1,a
out2    cjne  r1,#2,out1
        mov    a,#11010111b ;output "2"
        mov    P1,a
out1    cjne  r1,#1,out0
        mov    a,#11101111b ;output "1"
        mov    P1,a
out0    ret

delay   mov    r7,a
delay1  mov    r6,#255
delay2  djnz  r6,delay2
        djnz  r7,delay1
        ret

        .end

```

Autre petit truc pratique

Les voies de Windows sont impénétrables, surtout lors de l'apparition d'une nouvelle version de ce système d'exploitation. Jusqu'à présent le concepteur avait une certitude : après mise en route de l'ordinateur toutes les sorties d'interface présentent un niveau logique bas. Avec Windows Me nous avons noté un autre comportement : après mise sous tension la ligne DTR commence par présenter un niveau haut. Il est probable qu'il s'agit du résultat de l'exécution de routines de test Plug&Play sophistiquées du système. Le programmeur de Microsoft a sans doute oublié ensuite de remettre le système dans l'état d'origine.

Le résultat de cette situation est que la carte Flash retourne à l'état de réinitialisation si tant est qu'elle soit encore, lors de la mise en route du PC, connectée à l'interface de ce dernier. Cela peut être la source de problèmes si l'on n'est pas conscient de cette situation. « Et pourtant, tout fonctionnait encore parfaitement hier ! ». Le débutant non averti ne manquera pas de sauter sur son fer à souder, le vieux loup commencera toujours par mettre le bon Bill (Gates) en doute si ce n'est en cause.

La solution dans ce cas-là : soit déconnecter le connecteur de l'interface, soit on réinitialise le programme de la carte ce qui a pour effet de faire revenir à l'état d'origine standard.

Améliorations

Pour l'instant, notre dé ne tourne que sur le système de développement. Il n'en reste pas moins que déjà apparaissent les premiers souhaits d'amélioration. Un « vrai » dé ne s'arrête pas brutalement, mais perd progressivement son énergie. Il est fort possible de simuler ce comportement en améliorant le programme. Il suffira de jouer sur les actions du dé et les durées de temporisation pour leur attribuer les valeurs que l'on voudra. Le comportement du dé est défini dans la routine play, la temporisation dans la routine delay. Après relâchement du bouton-poussoir le dé « roulera » 6 positions plus loin, chaque passage à la position suivante se faisant plus lentement. On est déjà plus près de la réalité. Nous nous sommes aperçus au cours du développement que, lorsqu'il tourne rapidement, le dé présente un clignotement peu professionnel lors du maintien de l'action sur le bouton-poussoir. Ceci nous a amené à changer le programme de façon à ce que les 7 LED restent allumées tant que l'on appuie sur le bouton-poussoir.

Le **listage 6** donne la dernière version (provisoirement) du programme, que l'on transférera

vers un petit système à microcontrôleur propre. Cela ne signifie pas le moins du monde qu'il n'est plus possible d'améliorer quoi que ce soit au niveau du programme. Il n'est jamais mauvais de procéder à des expériences pour améliorer ses dons de programmeur.

Construction du système-cible

Il faudra commencer, avant de pouvoir transférer le programme vers le système-cible, donner corps à l'électronique concernée. Tout ce dont on a besoin sont le contrôleur, un quartz et un condensateur de réinitialisation.

Nous avons déjà défini l'utilisation des lignes de port. Le schéma de notre réalisation est ainsi pratiquement terminé. À une petite différence près entre les microcontrôleurs dont il faudra tenir compte : les lignes P1.0 et P1.1 du AT89C2051 ne comportent pas de résistance de forçage au niveau haut (*pull up*). Ces lignes servent en effet parallèlement d'entrées du comparateur analogique. Cela ne pose pas de problème lorsqu'il s'agit de commander des LED. Pour le bouton-poussoir au

contraire il faudra ajouter une résistance pull-up externe que ne requerrait pas à l'origine le système de développement.

Le programme mis au point est transféré dans le modèle plus petit de microcontrôleur à l'aide d'un programmeur adéquat. Il ne reste plus qu'à souder les différents composants et à vérifier le fonctionnement du montage. Si tant est que l'on n'ait pas fait d'erreur de connexion notre dé devrait rouler longtemps encore.

(021001)