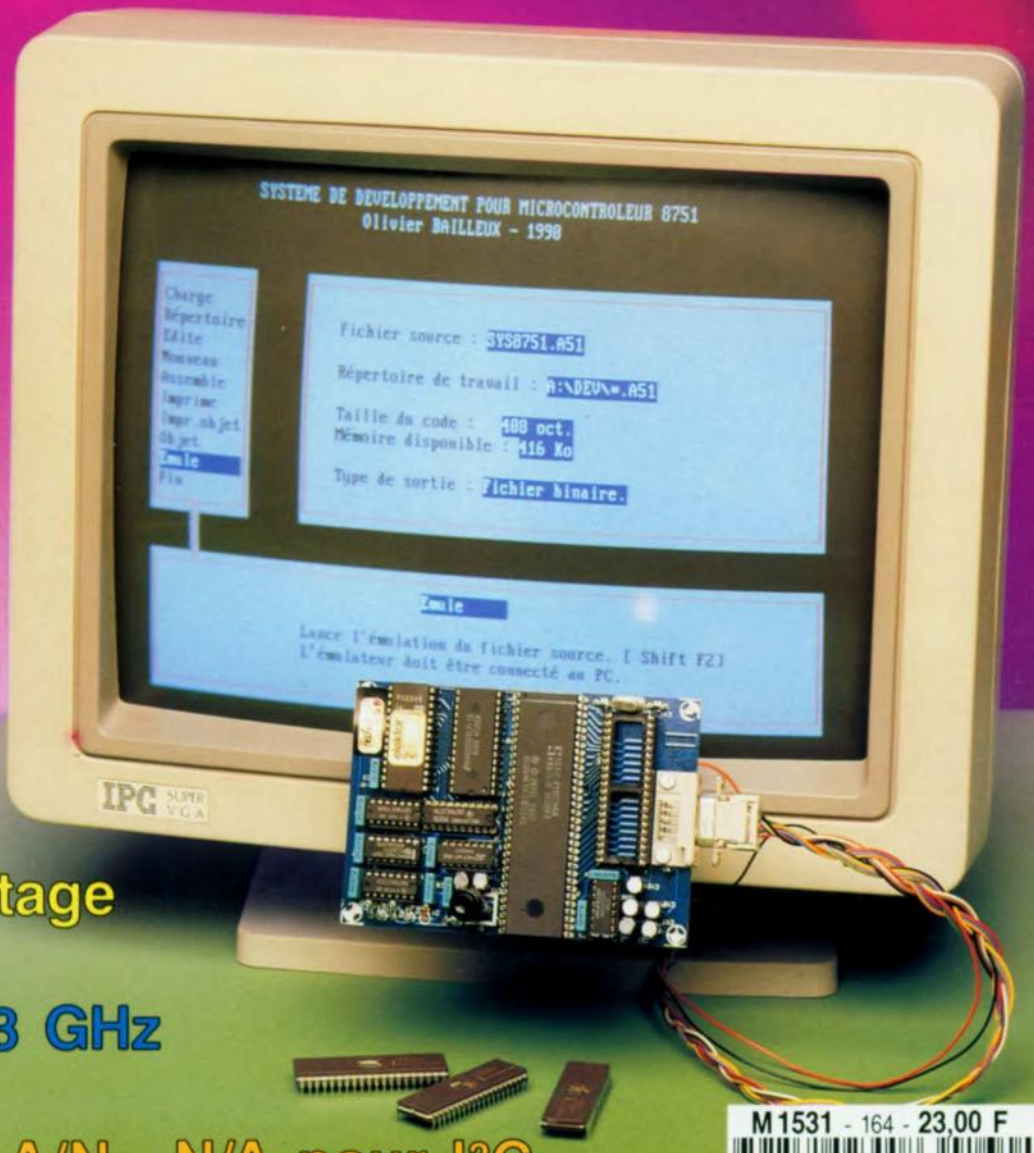
n° 164 - février 1992

ELEKTOR

l'électronique créative

émulateur de 8751



commutateur audio/vidéo électronique

convertisseur RS-232 – A/N

témoin d'écrêtage

prédiviseur 1,3 GHz

convertisseur A/N – N/A pour I²C

ELEKTOR 168 FB 7,80 FS mensuel

M 1531 - 164 - 23,00 F

SOMMAIRE

n°164
février 1992

Les 2 thèmes principaux de ce numéro sont la Micro-Informatique et l'Audio, Vidéo & Musique. Mais nous avons également abordé les domaines de la Mesure et des Hautes Fréquences –cette dernière est quelque peu tirée par les cheveux, nous en sommes conscients. 2 des montages-phares de ce numéro, un émulateur de 8751 et un programmateur universel, font partie de la première catégorie évoquée.

Services

18 elektor software service (ESS)
18 liste des circuits imprimés (EPS)
44 circuits imprimés en libre service

Informations

35 elekture: Windows 3 initiation
36 cours μC-8051 & assembleur (IV)
M. Ohsmann
42 logiciels: The Realizer, logiciel de «dessins» de logiciel
43 tort d'Elektor: télégradateur
53 chip select: AD 620
53 tort d'Elektor: tampon pour imprimante 1 ou 4 Moctets
54 améliorer le MCS® BASIC-52 V1.1 (II)
correction de problèmes apparaissant lors de multiplications
Z. Stojsavljevic & D. Mudric

RÉALISATIONS

Micro-informatique

21 programmateur universel pour PC (I)
pour EPROM, EEPROM et microcontrôleurs
48 convertisseur A/N–N/A et E/S pour I²C
mesurer, commuter, réguler avec un PC
56 émulateur de 8751
un puissant outil de développement
O. Bailleux

Audio-Vidéo & Musique

26 commutateur audio/vidéo électronique
universel et extensible à loisir
70 témoin d'écrêtage
avec indication –3dB additionnelle

Hautes-Fréquences

32 prédiviseur 1,3 GHz
UHF en entrée, TTL en sortie
P. Esser

Mesure

62 convertisseur RS-232 – A/N
interface pour ordinateur pour le 7109
B. Zschocke & A. Arnold

elektor infocarte — Table des matières 3 — Circuits intégrés tts types/linéaires/TTL, Circuits intégrés numériques/analogiques

CIRCUITS INTEGRES LINEAIRES N°
9 ampli opérationnel caractéristiques (II) 160
10 TDA 7053 187
11 NE/SA604A 189
12 UC1524/2524/3524 192
13 AD712J 194
14 TDA1525 circuit de commande de volume & de tonalité 195
15 SAA3049 décodeur de télécommande IR 198
16 MC34064 détecteur de sous-tension 199

TTL N°
11 compteur/diviseur par n programmable 4059 191
12 74HCT5555 193

CIRCUITS INTEGRES NUMERIQUES N°
1
2 TLC1541 convertisseur A/N 10 bits 171
4 PIC 16C54/5/6/7 174
5 SAB 0529/0529G 175
6 ZN428-E-8/J-8 Convertisseur N/A 177
7 GAL16V8 185
8 SPG8650 oscillateurs à quartz programmables 197
9PCD8584 contrôleur de bus I²C 200

CIRCUITS INTEGRES ANALOGIQUES N°
1 HA574A convertisseur A/N à 12 bits 167
3 DAC0808 convertisseur N/A 8 bits 172

MICRO-CONTROLEURS N°
1 8751(X)H Intel/AMD/NEC 178
2 8032 AH Intel 180
3 8035/39/40/48/49/50AH/8748/49H Intel 182

COMPOSANTS DISCRETS N°
17 bobines à air (I) bobines longues 169
18 bobines à air (II) bobines courtes 176
19 BAT 81/82/83/85/86 179
20 TIC206A . . . E,M,N,S 190

APPLICATIONS N°
12 adaptation de niveau entre les familles logiques 150

MELANGEURS DE FREQUENCE N°
1 MCL SBL-1 mélangeur à double symétrisation 184

elektor-infocartes

elektor infocarte — Table des matières 4 — Standards (Normes) Information générale

STANDARD (NORMES) N°
17 la lumière unités relatives à la lumièreis100
18 normes TV CCIR 625 lignes syst. L, B/G/H 102
19 fusibles sous verre tubulaireis 104
20 code des couleurs des bobines 105
21 symboles de programmation 106
22 prise Péritel 108
23 brochages des connecteurs pour cartouches MSX 120
24 brochage des connecteurs MSX 121
25 codage des varistors 127
26 connecteur IEC 625/ connecteur IEEE 488 129
27 fusibles 152
28 interface Centronics 162
29 code des couleurs pour diodes à faibles signaux 165
30 symboles opératoires logiques tables de vérité 166

INFORMATION GENERALE N°
21 la modulation 101
22 les effets thermo-électriques 103
23 filtres actifs RC1 109
24 filtres RC actifs 2 110
25 l'indice d'acidité (pH) 111
26 l'énergie solaire 112
27 le son 113
28 les formes d'ondes 114
29 le vent 115
30 antennes dipôles 119
31 la résonance de circuits LC 126
32

elektor infocarte | Table des matières 3 | Circuits intégrés tts types/linéaires/TTL Circuits intégrés numériques/analogiques

CIRCUITS INTEGRES LINEAIRES N°
9 ampli opérationnel caractéristiques (II) 160
10 TDA 7053 187
11 NE/SA604A 189
12 UC1524/2524/3524 192
13 AD712J 194
14 TDA1525 circuit de commande de volume & de tonalité 195
15 SAA3049 décodeur de télécommande IR 198
16 MC34064 détecteur de sous-tension 199

TTL N°
11 compteur/diviseur par n programmable 4059 191
12 74HCT5555 193

CIRCUITS INTEGRES NUMERIQUES N°
1
2 TLC1541 convertisseur A/N 10 bits 171
4 PIC 16C54/5/6/7 174
5 SAB 0529/0529G 175
6 ZN428-E-8/J-8 Convertisseur N/A 177
7 GAL16V8 185
8 SPG8650 oscillateurs à quartz programmables 197
9PCD8584 contrôleur de bus I²C 200

CIRCUITS INTEGRES ANALOGIQUES N°
1 HA574A convertisseur A/N à 12 bits 167
3 DAC0808 convertisseur N/A 8 bits 172

MICRO-CONTROLEURS N°
1 8751(X)H Intel/AMD/NEC 178
2 8032 AH Intel 180
3 8035/39/40/48/49/50AH/8748/49H Intel 182

COMPOSANTS DISCRETS N°
17 bobines à air (I) bobines longues 169
18 bobines à air (II) bobines courtes 176
19 BAT 81/82/83/85/86 179
20 TIC206A . . . E,M,N,S 190

APPLICATIONS N°
12 adaptation de niveau entre les familles logiques 150

MELANGEURS DE FREQUENCE N°
1 MCL SBL-1 mélangeur à double symétrisation 184

elektor-infocartes

elektor infocarte | Table des matières 4 | Standards (Normes) Information générale

STANDARD (NORMES) N°
17 la lumière unités relatives à la lumièreis100
18 normes TV CCIR 625 lignes syst. L, B/G/H 102
19 fusibles sous verre tubulaireis 104
20 code des couleurs des bobines 105
21 symboles de programmation 106
22 prise Péritel 108
23 brochages des connecteurs pour cartouches MSX 120
24 brochage des connecteurs MSX 121
25 codage des varistors 127
26 connecteur IEC 625/ connecteur IEEE 488 129
27 fusibles 152
28 interface Centronics 162
29 code des couleurs pour diodes à faibles signaux 165
30 symboles opératoires logiques tables de vérité 166

INFORMATION GENERALE N°
21 la modulation 101
22 les effets thermo-électriques 103
23 filtres actifs RC1 109
24 filtres RC actifs 2 110
25 l'indice d'acidité (pH) 111
26 l'énergie solaire 112
27 le son 113
28 les formes d'ondes 114
29 le vent 115
30 antennes dipôles 119
31 la résonance de circuits LC 126
32

elektor infocarte

Table des matières 3

Circuits intégrés pour μ-ordinateurs
Information en ts genres

CIRCUITS INTEGRES POUR μ-ORDINATEURS N°
9 ACIA 6551 107
10 processeur graphique EF 9365/66/67 (I) 116
10 processeur graphique EF 9365/66/67 (II) 117
10 processeur graphique EF 9365/66/67 (III) 118
11 convertisseur A/N-N/A rapides UVC 3100/3101 (ITT) 122
12 EPROM Brochage (2) 123
13 RAM statique Brochage (2) 124
14
15 RAM dynamiques brochage (2) 125
16 le format Motorola Exorciser 137
17 le format Intel Intellec 8/DMS 144
18 Tektronix format hexadécimal 145
19 Z80 PIOs généralités/brochage 154
20 EPROM brochage (3) 157
21 PPI 8255 159
22 UART 8251 168
23 UART AY-3-1015D 170
24 DRAM (1) 256 Kbits x 9 183
25 DRAM (2) 1 Mb x 9 186
26 DRAM (3) 4 Mb x 9 196

LE MINITEL 1B N°
1 standard Télétel mode Vidéotex
résumé des commandes 140
2 standard télétel mode mixte
standard téléinformatique 141
3 clavier au standard téléinformatique6 143

TELECOMMANDE IR N°
1 MV500/MV601 émetteur/récepteur télécommande IR 181

INFORMATION SPECIFIQUE N°
1 interface sérielle alimentation à 8751 142
2 jeu de caractères IBM 146
3 SESAME 148
4 Q4: module de commande MIDI 149
5 Jeu de caractères IBM (2) 153
6 EDiTS: commandes par l'interface RS 232 156
7 EPROM du circuit de clavier MIDI universel 158
8 les codes modem selon Hayes 164
9 Standard vidéo IBM PC 188

elektor-infocartes

elektor infocarte

Table des matières 4

Information générale

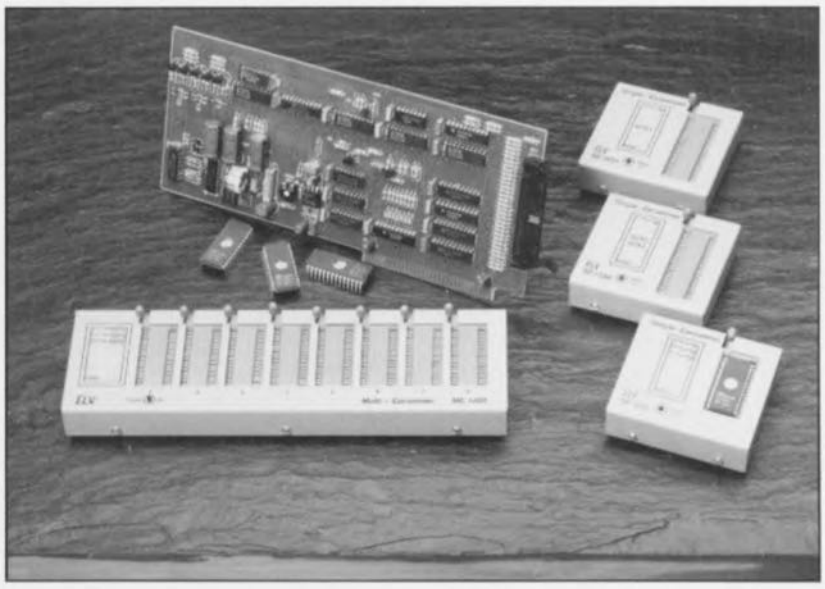
N°
33 la réactance des condensateurs 128
34 filtres pour HP passe-haut du second ordre
filtres pour HP passe-bas du second ordre 130
35 théorème des filtres- généralités
théorème de la superposition 131
36 filtres pour H.P. passe-haut du 3ème ordre
filtres pour H.P. passe-bas du 3ème ordre 132
37 les réseaux théorème.de Thévenin 133
38 filtres pour HP passe-haut du 4ème ordre
filtres pour HP passe-bas du 4ème ordre 134
39 les transformations de Blakesley
les théorèmes de réciprocité 135
40 le serveur d'Elektor
MEMENTO MINITEL 3615 + ELEKTOR 136
41 filtres pour HP passe-tout
filtres pour HP série du 1er ordre
42 filtres pour HP compensation de HP 139
43 (I) tables de conversion hexadécimal <-> décimal 147
43 (II) tables de conversion hexcimal <-> décimal 147
43 (III) tables de conversion hexcimal <-> décimal 147
44 connecteurs pour caméra vidéo 151
45 symboles mathématiques & unités de mesure (1) 155
46 symboles mathématiques & unités de mesure (2) 161
47 constantes de temps RC/relations entre la tension,
l'intensité et la puissance 163
48 unités légales (3) thermique/mécanique 173

description d'un kit ELV

programmateur universel pour PC

1e partie

pour microcontrôleurs, EEPROM et EPROM jusqu'à 2 Mbits



Le montage décrit ici permet la programmation de la quasi-totalité des EPROM disponibles sur le marché, de la 2716 à la 272001, c'est-à-dire d'une capacité allant de 2 Koctets à 256 Koctets, celle de certaines EEPROM et celle des microcontrôleurs des séries 8748 et 8751.
Ses caractéristiques techniques en font une réalisation semi-professionnelle qui n'a pas à rougir face à ses concurrents notablement plus chers. La connexion d'un module de programmation simultanée (*Gang Adapter*) –ce n'est plus «le gang des 4», mais «le gang des 8»– permet la programmation simultanée d'un maximum de 8 EPROM. Le logiciel à menus déroulants associé à cette réalisation devrait répondre aux besoins des amateurs les plus exigeants.

Assez de parlottes, venons-en aux faits.
On trouve aujourd'hui sur le marché un nombre de plus en plus grand de programmateur de toutes sortes : programmateurs d'EPROM sériels ou parallèles, programmateurs d'EEPROM, programmateurs de microcontrôleurs, etc, etc. Des versions «amateur» les plus simples, capables de programmer un, voire deux types d'EPROM, aux modèles les plus professionnels dont le prix se situe bien au-delà des possibilités de nombreux amateurs, la palette est richement fournie.
Si nous optons quand même, en dépit de l'existence de cette foison d'appareils du commerce, de vous proposer une telle réalisation, nous sommes bien conscients que nous devons vous proposer un montage au rapport performances/prix intéressant, voire étonnant.

Le concept orienté vers l'avenir de ce nouveau **programmateur d'EPROM universel pour PC** convient tout à la fois, de par ses possibilités, aux applications industrielles, et de par son coût relativement abordable, à une utilisation par l'amateur, catégorie d'utilisateurs auxquels Elektor est particulièrement attaché. Le logiciel accompagnant ce montage peut se targuer d'associer un confort «moderne» à une efficacité «redoutable» qui le prédestine donc à une utilisation fréquente.

Outre les EPROM archi-connues actuelles (voire dépassées) qui vont de la 2716 à la 27512, ce programmateur est également en mesure de programmer les EPROM de 1 et 2 Mbits. L'utilisateur couvre ainsi la quasi-totalité des EPROM utilisées aujourd'hui. Le système est en outre conçu de telle manière que le jour où il le faudra, il permettra également la programmation des EPROM de 4 et 8 Mbits (512 Koctets et 1 Moctet) ainsi que celle des versions à 16 bits. Une réalisation pouvant affronter sans crainte les rigueurs du temps qui passe.

Il existe pour les EPROM à 8 bits un adaptateur pour programmation multiple permettant de programmer simultanément jusqu'à 8 EPROM du même type.
Des adaptateurs additionnels permettent la programmation des microcontrôleurs des séries 8748 et 8751.

Dans ce premier article nous nous proposons de vous décrire l'installation de la carte et d'approfondir le détail du logiciel, de manière à vous permettre de vous faire une bonne idée de quoi il retourne lorsque vous déciderez de vous lancer dans cette réalisation.
Dans le second article nous nous intéresserons à la réalisation de la carte, un circuit imprimé double face à trous métallisés et connecteur, enfichable dans tout connecteur d'extension pour PC. Tous les composants importants trouvent place sur cette platine qui est reliée au support de programmation proprement dit via un câble plat.
L'adaptateur à support comporte lui aussi une platine, petite dans le cas de la version mono-EPROM, mais notablement plus longue pour la version à 8 supports. Tous les adaptateurs à support FIN TEXTOOL trouvent place dans un boîtier métallique.

L'installation de la carte

Peut-être faites-vous partie des rares amateurs (fortunés) de réalisations électroniques achetant leurs «kits» tout faits. Si tel est le cas, vous pourrez sauter une bonne partie du texte à venir, puisque la seule étape de cette description qui pourra vous intéresser sera la mise en place de la carte dans l'ordinateur et le mode d'emploi de son logiciel.

Comme le suggère son nom, le **programmateur universel pour PC** comporte une carte encartable dans un PC et un ou plusieurs modules de programmation à utiliser en fonction du type de composant à programmer.
L'adaptateur –puisqu'il n'en est toujours utilisé qu'un seul à la fois– est relié à la carte principale par l'intermédiaire d'un câble plat à 40 conducteurs. Le logiciel est fourni sur une disquette 5''¼ de 360 Koctets. Comme de coutume, cette disquette comporte un fichier **READ.ME** qui propose les informations les plus récentes concernant cette réalisation. Il est vital de commencer par lire ce fichier.

Avant de pouvoir remplir la tâche pour laquelle il a été conçu, le **programmateur universel pour PC** doit, veuillez ne pas nous en vouloir de le préciser, être enfiché dans un connecteur d'extension d'un IBM-PC ou Compatible –travaillant sous le SED MS-DOS3.2 ou une version plus récente. On copiera ensuite le logiciel dans un nouveau répertoire à créer sur le disque dur, ce que l'on réalise en entrant l'instruction suivante :

<A: install>.

Le programme d'installation demande le répertoire à créer; il le crée ensuite, si tant est que celui-ci n'existe pas encore et y copie le logiciel de programmation.

Avant de procéder à l'implantation de la carte dans l'ordinateur il est important, plus que cela, impératif, de savoir si le domaine d'adresses prévu, à savoir $300 à $307 n'est pas déjà occupé par une autre carte d'extension. Un conflit d'adressage se traduit le plus souvent par un mauvais fonctionnement du matériel.
Une fois lancé par l'instruction **<ELVPROM>** le logiciel commence par s'assurer de la présence de la carte du programmateur universel et, le cas échéant, signale son absence éventuelle par un message d'erreur.

En règle générale on trouve les adresses occupées par un montage encartable additionnel dans le manuel qui l'accompagne; armé de ces informations, il ne vous devrait pas être trop difficile de trouver une adresse libre dans le domaine en question.
Le **tableau 1** donne les adresses possibles et libres. Lors de cette recherche, le logiciel sera d'une aide précieuse sachant qu'il a été écrit de manière à refuser les domaines d'adresses illicites (ceux du disque dur, du *timer*, du contrôleur DMA, de la carte graphique), pour la simple et bonne raison qu'un tel adressage s'est souvent déjà traduit par des conséquences graves, telles qu'un crash du système voire pire encore par la perte de données !

Si l'on envisage de modifier le domaine d'adresses prévu à l'origine, il faudra, avant d'avoir implanté la carte, lancer le logiciel. En raison de l'absence de la carte, le programme fournira le message d'erreur suivant «Il n'y a pas de matériel. Quitter avec <ESC>». Il faudra alors quitter le programme et lancer le programme **<PROMINST>** qui permet de modifier l'adresse de base de la carte et d'effectuer un certain nombre d'autres opérations.

La mise en place de la carte du programmateur universel pour PC se fera en respect des recommandations du manuel normalement fourni avec le PC.

Voici, en résumé, les étapes à suivre pour cette implantation.
– Couper le courant, et débrancher tous les câbles,
– Selon le cas, ouvrir le capot ou dévisser la face arrière du PC (bien souvent on a 5 vis disposées à proximité immédiate du coffret de l'appareil... mais c'est vrai, vous avez sans doute déjà ouvert votre machine à maintes reprises),
– Choisir un connecteur libre; la carte se contente d'un connecteur à 8 bits,
– Dévisser la plaquette de protection anti-poussière montée dans la face arrière,
– Définir, le cas échéant le domaine d'adresses sur la carte, – Enficher avec toutes les précautions nécessaires la carte dans le connecteur d'extension choisi (l'embase tournée vers l'extérieur du coffret) et la fixer,
– Après une dernière vérification, refermer le coffret de l'ordinateur et reconnecter les câbles.
La mise sous tension se traduit alors, si tant est que l'on ait installé le logiciel et que la machine refonctionne correctement, par l'apparition à l'écran du menu général.

Il n'est pas nécessaire de procéder à quelque autre installation que ce soit, sachant que le logiciel reconnaît automatiquement le type de carte graphique et le modèle de moniteur présents.

Mode d'emploi

Une fois l'installation effectuée

Caractéristiques techniques
- Réalisation encartable,
- Pour IBM-PC, XT, AT et compatibles,
- Peut programmer toutes les EPROM de la 2716 à la 27C2001,
- Permet de programmer les EEPROM,
- Programmation possible des microcontrôleurs 8748/49/51,
- Existe en version mono et multi-EPROM,
- Logiciel puissant et confortable,
- Sélection possible de la tension de programmation,
- Prévu pour la programmation des futures EPROM de 4 et 8 Mbits.

Tableau 1. Adresses pouvant être libres sur un PC
Les domaines d'adresses non mentionnés sont occupés par des composants matériels et autres cartes «vitales».

$200 à $207	Manche de commande (Joystick)
$208 à $20F	non documenté [1]
$218 à $277	non documenté [1]
$278 à $27F	LTP2 [2]
$280 à $2F7	non documenté [1]
$2F8 à $2FF	COM2 [2]
$300 à $31F	cartes d'extension
$320 à $35F	non documenté [1]
$360 à $36F	Réseaux Locaux
$370 à $377	non documenté [1]
$378 à $37F	LPT1 [2]
$380 à $3AF	non documenté [1]
$3F8 à $3FF	COM1 [2]

[1] Le système d'exploitation MS-DOS ne donne pas d'autre information concernant ces domaines d'adresses. En cas d'utilisation de l'un de ces domaines, on le fera avec les précautions de rigueur.
[2] Il est formellement déconseillé d'utiliser l'une de ces adresses sachant que le BIOS et autres programmes utilitaires s'attendent à trouver à cet endroit des interfaces standard.

avec succès, l'instruction <**ELV-PROM**> permet de démarrer le logiciel du **programmateur universel pour PC** –ou encore, si l'on a installé **DOSBATCH** pour utiliser le **testeur de circuits intégrés** ou le **testeur de semi-conducteurs**, on pourra démarrer le logiciel en optant pour le point de menu correspondant.
On voit alors apparaître un menu déroulant à plusieurs rubriques successives qui permettra la sélection de l'opération requise. Nous n'allons pas entrer dans le détail de ce programme. Son confort et sa structure sont suffisamment clairs pour que tout possesseur de PC puisse d'en tirer sans égratignure. L'écran est divisé en 2 parties : les 2/3 supérieurs constituent la zone de travail, le 1/3 inférieur la zone d'information. La zone de travail visualise les possibilités et le choix de l'utilisateur, la zone d'information fournit des indications supplémentaires concernant les cheminements (*path*), domaines d'adresses et autres informations concernant le déroulement de certains processus.

Le logiciel a prévu 4 types de champs pour permettre une communication interactive :
– un *champ de saisie*, qui demande, par exemple, le nom du fichier. Le processus est interrompu par <**ESC**>, une action sur <**RETURN**> se traduisant par la prise en compte de l'élément concerné.
– un *champ de menu :* à cet endroit, l'utilisateur se voit proposer un certain nombre d'options (type d'EPROM par exemple, comme l'illustre la **figure 1**).
– un *champ d'aide :* le programme indique que l'exécution de l'opération demandée par l'utilisateur peut avoir des résultats autres que ceux requis –écrasement d'un fichier déjà présent dans le tampon par le chargement d'un nouveau fichier par exemple.
– un *champ d'indication d'erreur :* on y lit un message d'erreur signalant l'impossibilité de poursuivre un processus donné.
Pour vous permettre de bien comprendre le concept à la base du logiciel nous allons examiner les différentes possibilités offertes par le menu.

Le sous-menu «File»

On a regroupé, sous ce menu déroulant, toutes les opérations faisant appel à des fichiers, de la lecture de fichiers présents sur le disque dur jusqu'à l'écriture sur disquette d'un fichier modifié ou non en mémoire. Voici les options dont on dispose :

Load Permet de charger en mémoire un fichier présent à l'origine sur disque (quel qu'il soit). Le programme demande de confirmer le cheminement, le nom et l'extension. Lors de la première ouverture de cette option, on se voit proposer une option définie à l'origine par <**Options/Default File**>|. Les «jokers» sont admis lors de la recherche d'un fichier. Un champ du menu visualise les fichiers répondant aux critères de recherche définis. Le programme utilise son propre répertoire comme cheminement tant que celui-ci n'a pas été redéfini par l'utilisateur par modification du <**Path**>.

Le programme reconnaît, à l'aide de l'extension du fichier, s'il s'agit d'un fichier programmable tel quel (extension ***.COM**, ***.BIN**) voire s'il peut être converti. Dans ce dernier cas, un menu additionnel propose un certain nombre de conversions, récapitulées dans le **tableau 2**.

Lors de la conversion en code machine le logiciel procède à une comparaison entre la somme de vérification et son propre résultat et indique une éventuelle différence dans la zone d'affichage des messages d'erreur.
De par leur structure, les fichiers ***.EXE** ne sont pas programmables et ne sont donc pas acceptés. Il faudra donc procéder à une conversion d'un tel fichier en un fichier ***.COM** en s'aidant du programme **EXE2BIN** (**EXE**cutable **to BIN**ary).

Tableau 2.
Possibilités de conversion de fichiers sous <Load>

pas de conversion
Intellec 8/MDS
Intel MCS86
Motorola S1
Motorola Exormax
Motorola S3
Tektronix

Save Sauvegarde le contenu de la mémoire sur le disque dur. Outre le nom à donner au fichier (où les «jokers» ne sont pas admis) le programme demande également l'adresse de début et de fin du programme. Une fois ces informations disponibles, le fichier est sauvegardé comme une recopie de la mémoire.

Path Permet de modifier le cheminement en cours d'utilisation. Comme indiqué plus haut, il existe un cheminement par défaut, qui est celui du sous-répertoire dans lequel se trouve le programme.

Delete Sert à effacer le fichier mentionné dans le champ de menu concerné et correspond donc à

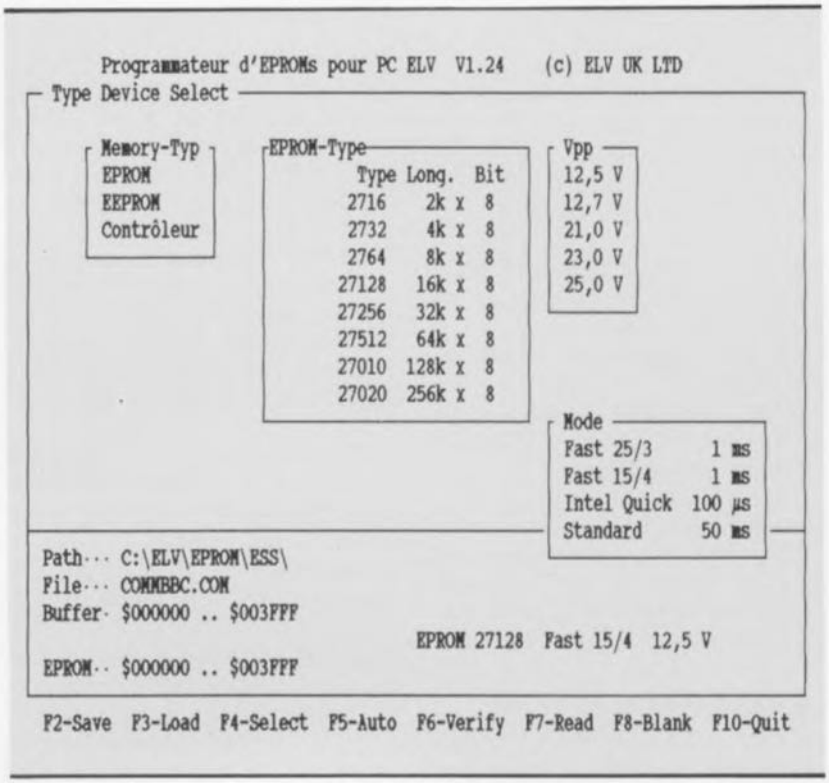


Figure 1. Recopie de l'un des écrans de choix des éléments de programmation. Comme on le voit, l'utilisateur peut opter pour l'un des 3 types de «mémoire» possibles : EPROM, EEPROM et microcontrôleur.

l'instruction classique **DEL** de MS-DOS. Pour éviter tout risque inutile, il n'est possible ici d'effacer, à l'inverse de ce que permet MS-DOS, qu'un seul fichier à la fois, car en cas d'effacement involontaire un **Undelete** automatique –à l'aide de PCTools ou XTREE– d'un fichier **.COM** ou **.EXE** réparti sur plusieurs «clusters» devient impossible.

Rename Permet de donner un nouveau nom au fichier indiqué dans le champ de menu. À nouveau, cela correspond à l'instruction **REN** classique de MS-DOS, avec la même remarque qu'il n'est possible de modifier le nom que d'un fichier à la fois. Il est licite d'effectuer des opérations de transfert (Move) entre répertoires du lecteur actif.

Le sous-menu Type

Ce menu déroulant concerne toutes les actions ayant regard à un type d'EPROM.

Device Select Propose différents types de mémoire programmables, EPROM, EEPROM et microcontrôleur. Une fois un type de composant choisi, on verra apparaître les différents composants reconnus par le logiciel dans chacune des catégories en question ainsi que la configuration mémoire correspondante. S'il existe plusieurs modes de programmation pour le dit composant on verra apparaître une (ou plusieurs) nouvelle(s) fenêtre(s) avec, le cas échéant, la tension de programmation dans la première et, dans la seconde, le mode de programmation. Le nom du fichier présent en mémoire et la taille de mémoire utilisée sont indiqués dans le champ d'information, comme l'illustre la recopie d'écran de la figure 1.

Device List Donne la possibilité de choisir le type de composant en fonction de sa dénomination. Il existe à cet intention un fichier de caractéristiques. Si le type en question n'est pas prévu, il est possible de repasser à l'option **EPROM/Device Select**.

Module Select Exige un type d'adaptateur de programmation différent si tant est que l'on n'ait pas procédé à une sélection de type automatique.

Le sous-menu EPROM

Ce sous-menu concerne les opérations nécessaires pour la lecture et la programmation d'une EPROM.

Auto Enchaîne automatiquement les opérations «**Blanktest**», «**Program**» et «**Verify**». Le logiciel demande les adresses de début, de fin et de début de mémoire. Il ne reste plus ensuite qu'à mettre l'EPROM dans le support, à refermer le levier, pour que la programmation se fasse automatiquement. Toute détection d'erreur se traduit par l'arrêt du processus de programmation.

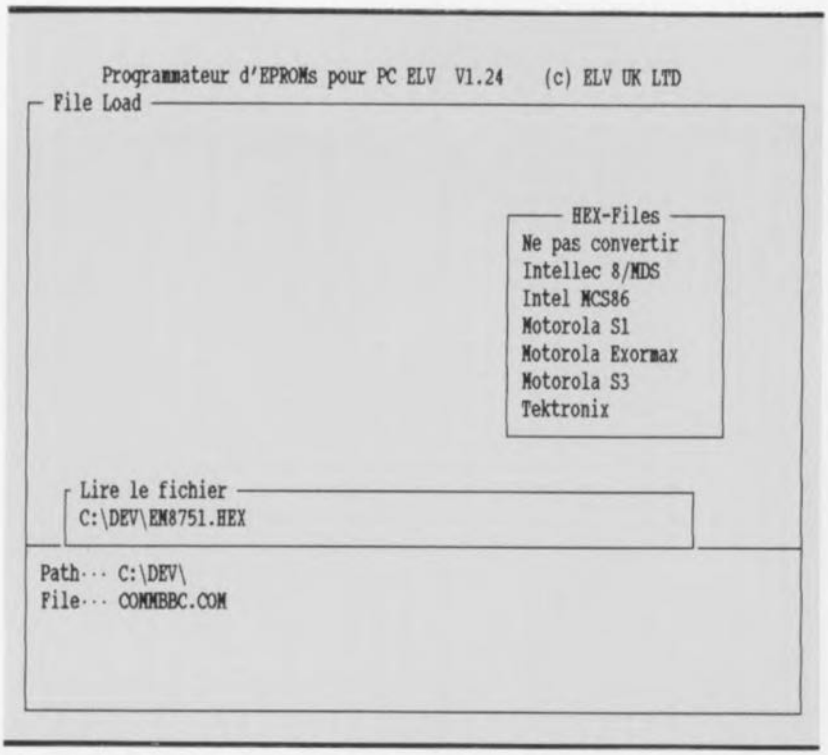


Figure 2. Sélection de l'un des modes de conversion possibles vers le format binaire pour EPROM. Cette fonction de conversion est effectuée directement par le logiciel.

Blanktest Vérifie que la mémoire implantée dans le support présente bien le caractère de virginité prévu pour le type de composant concerné ($FF pour une EPROM par exemple).

Program Programme, dans la mémoire implantée dans le support, de l'adresse de début à l'adresse de fin indiquée, les données présentes en mémoire en utilisant le processus choisi dans la fenêtre de **Type**.

Verify Vérifie que les données écrites dans la mémoire implantée dans le support correspondent, dans les limites des adresses de début et de fin définies par l'utilisateur, aux données présentes en mémoire.

Read Sert à effectuer une lecture des données stockées dans la mémoire du composant implanté dans le support, dans le domaine défini par les adresses de début et de fin.

Clear Sert à l'effacement des EEPROM (notez bien le double EE !). Il ne saurait être question d'essayer d'effacer une EPROM en faisant appel à cette fonction.

Le sous-menu Data

Ce sous-menu rassemble les opérations servant à la modification des données présentes en mémoire. Sachant que les opérations **Edit**, **Fill** et **Move** comportent des risques non négligeables d'erreurs «opératoires», il est possible de les «mettre sous les verrous» à l'aide de l'option **Data Modify** du sous-menu **Options**.

Dump Visualise le contenu d'un fichier à l'écran. Il n'est cependant pas possible de modifier les données qu'il contient.

Edit Permet de modifier les données présentes en mémoire. On visualise à chaque fois 256 octets dans la fenêtre; on peut ensuite modifier ces données soit sous format binaire, soit sous format ASCII. Les octets modifiés sont marqués pour être parfaitement identifiables. Ceci fait, l'utilisateur peut choisir de procéder ou non à la modification définitive des données du fichier.

Fill Place, dans chacun des emplacements de mémoire d'un domaine d'adresses défini par une adresse de début et une adresse de fin, un caractère défini par l'utilisateur.

Move Déplace un domaine de mémoire compris entre une adresse

de début et une adresse de fin vers un nouveau domaine commençant à une adresse dite de destination.

Compare Compare un domaine de données défini par une adresse de début et une adresse de fin avec un domaine en mémoire commençant à l'adresse de destination. Les différences détectées sont visualisées.

Le sous-menu Options

Ce menu rassemble les opérations spécifiques concernant l'installation du programme, évoquées brièvement dans les lignes précédentes. Passons-le rapidement en revue :

Default File Propose un fichier standard ou des «jokers» (tels que ***.BIN**). **Default Conv** visualise l'état de la barre de menu lors de l'ouverture du sous-menu de conversion. L'accès en est accéléré.

Default Prom Donne l'état de la barre de menu lorsque le sous-menu de **Type** (et d'autres éventuellement) est ouvert. Il est possible de cette manière de faciliter le choix d'un type d'EPROM souvent utilisé.

Data Modify Bloque ou débloque l'accès à certaines opérations à risque : **Edit**, **Fill** et **Move**.

Save Options Permet de sauvegarder sur disque les nouvelles options définies par l'utilisateur, options stockées dans le fichier de configuration **.CFG**.

Le sous-menu Quit

Cette option permet, de la même manière que <**ESC**> du menu, de quitter le programme. Les fichiers ouverts sont refermés, les domaines des mémoire occupés dynamiquement (par des données) ainsi que d'éventuels fichiers créés pour pouvoir travailler avec des EPROM de grande capacité sont effacés.

Après cet article de présentation, nous nous intéresserons dans l'article du mois prochain à l'aspect électronique de cette réalisation et, dans un troisième et dernier article, à sa construction et son mode d'emploi.

```
Programmateur d'EPROMs pour PC ELV  V1.24    (c) ELV UK LTD
                          File View
C:\COMMBBC.COM

000000  4C 2F 81 4C  48 80 C2 1A  01 43 4F 4D  4D 55 4E 49  L/ûLHÇ┬→GCOMMUNI
000010  43 41 54 4F  52 00 31 2E  30 35 00 28  43 29 31 39  CATOR 1.05 (C)19
000020  38 33 20 53  70 65 63 69  61 6C 69 73  74 20 53 6F  83 Specialist So
000030  66 74 77 61  72 65 20 50  72 6F 64 75  63 74 73 20  ftware Products
000040  4C 74 64 00  50 32 00 FF  20 48 C9 20  F0 20 C9 20  Ltd P2   H╒ ≡ ╒
000050  D0 20 4C B2  B6 C9 20 D0  1E 4C E7 87  98 48 8A 48  ╨ L▒╢╒ ╨▲L╥çÿHèH
000060  A2 00 B1 F2  29 DF DD 20  80 D0 18 C8  E8 E0 20 90  ó ▒≥)▀ Ç╩↑╚Φα É
000070  F1 B0 14 68  AA 68 A8 68  28 60 68 AA  68 A8 68 A9  ±▒¶h¬hèh(`h¬hèh-
000080  00 28 60 C9  2E D0 EC A9  00 85 00 85  21 85 22 68   (`╒.╨∞- à à!à"h
000090  AA 48 A9 8E  20 F4 FF 20  1C 8A 45 72  72 6F 72 3A  ¬H-Ä ⌠ ¬èError:
0000A0  20 00 60 20  97 80 A0 21  B1 FD F0 22  20 EE FF C8   ` ùÇá!▒²≡" ∈ ╚
0000B0  4C A8 80 20  E7 FF 4C 7E  81 A0 00 B1  FD C8 85 28  L¿Ç τ L~ûá ▒²╚à(
0000C0  C9 21 D0 23  20 21 88 20  94 B5 4F 53  20 65 72 72  ╒!╨# !ê ö╡OS err
0000D0  6F 72 3A 00  A9 20 20 EE  FF A0 21 B1  FD F0 22 20  or: ⌐  ∈ á!▒²≡"
0000E0  EE FF C8 4C  DB 80 20 21  88 A5 27 C9  21 D0 23 A9  ∈ ╚L╫Ç !êÑ'╒!╨#⌐
0000F0  00 85 18 85  19 A4 2A F0  1D 85 2A AA  20 CE FF 4C  à↑à↓ñ*≡◆à*¬ ╬ L

Pg↕  <ESC>-Exit    F9-GoTo
```

Figure 3. La fonction <Edit> du logiciel du programmateur universel pour PC en prenant pour exemple un fichier de données typique.

SIOUX

Patrick JEAN-ELIE

J'ai le plaisir de vous envoyer un logiciel nous dit Mr Jean-Elie dans sa lettre, que j'ai baptisé SIOUX et qui est dédié à la carte SCALP et aux systèmes de la famille MCS-51, tournant sur Atari-ST.

Ce logiciel comporte:
* un éditeur de texte pleine-page,
* une émulation de console série entièrement paramétrable (vitesse de 600 à 19 200 bauds),
* une fonction d'envoi et exécution d'un programme BASIC depuis l'éditeur ou depuis un fichier vers la carte SCALP.

Le dialogue avec la carte peut sortie en même temps sur l'écran et sur une imprimante connectée à l'Atari.

Il contient également une fonction de désassemblage de tout l'espace mémoire du microcontrôleur, dont le résultat peut être affiché, imprimé, ou sauvegardé dans un fichier. Plusieurs formats d'affichage permettent la récupération du désassemblage pour édition, ou pour recompilation. On peut aussi désassembler un fichier binaire, produit de la fonction Dump.

Cette fonction Dump permet d'afficher, d'imprimer ou de sauvegarder tout ou partie de la mémoire (le fichier résultant pourra être désassemblé).
Des applications MIDI peuvent être développées, le choix de l'interface est possible, entre la RS-232 et les prises MIDI. Une option d'exécution de programmes Atari sans quitter SIOUX permet aux inconditionnels de travailler avec leur éditeur favori, ou aux autres de lancer un compilateur, ou un langage... (en fonction de la mémoire disponible).

L'environnement initail est restauré au retour.

Les menus déroulants et boîtes de dialogue sont inédits, pour la convivialité.
Ce programme n'a qu'une prétention didactique. Des extensions sont prévues (Fonction Renum, et peut-être un compilateur...). Il est disponible au prix de 300 FF à l'adresse ci-dessous.
Intéressés ?

NdlR : Il arrive souvent à la rédaction plusieurs logiciels «amateur», permettez-nous l'utilisation de cet adjectif, par mois; il nous faut faire un choix, et si l'un d'entre eux nous semble digne d'intérêt, nous en parlerons dans cette rubrique.
Il nous faut cependant reconnaître que nous ne pouvons pas tester ce type de logiciel dans ses moindres détails. Il existe de nombreux logiciels «Shareware» dans le monde du PC...

Patrick JEAN-ELIE
693, Avenue de Mazargues A2
13009 Marseille

commutateur audio/vidéo électronique

universel et extensible

Plus le nombre d'appareils audiovisuels faisant leur apparition dans les salons et autres chambres à coucher devient important, plus leur interconnexion pose un problème délicat, de plus en plus souvent insoluble. On veut pouvoir transférer, d'un appareil à l'autre, non seulement les signaux audio mais aussi et surtout les signaux vidéo.
Un magnétoscope devrait, par exemple, pouvoir envoyer ses images simultanément vers 2 téléviseurs, alors qu'il devrait en outre être possible à l'utilisateur de pouvoir, avec ce même magnétoscope, enregistrer le son en provenance soit de l'un des téléviseurs soit de la chaîne audio stéréophonique. Le commutateur audio/vidéo électronique proposé ici permet toutes les manipulations évoquées plus haut, et bien d'autres; il a en outre l'avantage d'être extensible, vu que l'on peut augmenter le nombre des entrées jusqu'à ce que l'on dispose d'autant d'entrées qu'il est nécessaire.

Il est dépassé le temps où chaque foyer ne disposait que d'un seul téléviseur, après n'avoir possédé, pendant de longues années, qu'un seul poste de radio –qui d'entre vous se souvient encore des postes à galène ?

Les chaînes audio stéréophoniques, les magnétoscopes, les téléviseurs multiples, le caméscope –il ne faudrait pas exagérer quand même– le lecteur de DAN portable ou le magnétophone DAT, on ne sait plus très bien où arrêter l'inventaire.

C'est lorsque l'on envisage d'interconnecter plusieurs de ces appareils que l'on constate avec amertume que les choses sont loin d'être aussi simples que l'on croyait, en particulier en raison du fait qu'il faut procéder à la commutation tant de signaux BF (basse fréquence), les signaux audio, que de signaux HF (haute fréquence vous l'aurez sans doute deviné), les signaux vidéo. C'est alors que l'on se rend compte des services que pourrait rendre un boîtier de commutation permettant le «dispatchage» de signaux audio et de signaux vidéo. Et c'est tout justement là la raison de l'existence du **commutateur audio/vidéo** faisant l'objet de cet article. Il peut en outre servir également pour le branchement de 2 appareils, voire plus, à une entrée vidéo d'un téléviseur ou d'un moniteur vidéo. Si l'on se trouve en présence d'un modèle de téléviseur relativement ancien, cette possibilité ne manquera pas de s'avérer fort intéressante.

La commutation de signaux vidéo (d'une bande passante de 5,5 MHz, voire plus) est plus délicate que celle de signaux audio BF (20 kHz). Un rotacteur standard n'est sans doute pas l'organe de commande le plus adapté à la commutation de tels signaux. Ceci explique que le fabricant de pommes frites, pardon de chips, Philips, ait lancé sur le marché un circuit intégré spécialement conçu à cet effet, capable de commuter simultanément et électroniquement 2 signaux audio (stéréo) et 2 signaux vidéo.

La **figure 1** donne le synoptique de la structure interne du circuit intégré concerné, à savoir le TDA8440. Bien que ce circuit intégré ait été conçu à l'origine pour être commandé par l'intermédiaire du bus I²C, il est également possible de le mettre à contribution en en effectuant la commande à l'aide de niveaux logiques classiques.

C'est la raison pour laquelle les entrées de sélection S0, S1 et S2 remplissent une double fonction. Ces 3 lignes permettent d'une part la sélection de l'une des 7 adresses potentielles disponibles sur le bus I²C. Il devient possible ainsi de connecter jusqu'à 7 de ces circuits intégrés en parallèle au bus I²C. D'autre part, une mise simultanée de ces 3 entrées au niveau logique haut fait passer le circuit intégré en mode «non bus I²C» (*not I²C bus mode*). Lorsque le circuit se trouve dans ce mode de fonctionnement il est possible de le commander à

Caractéristiques techniques de la partie audio :

■ Distorsion harmonique (20 Hz à 20 kHz) :	<0,004%
■ Distorsion d'intermodulation harmonique (carré 3,15 kHz, sinus 15 kHz) :	<0,002%
■ Diaphonie (séparation entre canaux) :	70 dB
■ Interaction audio (canal 1 – canal 2) :	– 80 dB
■ Interaction vidéo/audio :	< – 90 dB
■ Variation d'amplitude (20 Hz à 20 kHz) :	+ 0/ – 0,2 dB
■ Gain audio (1 kHz) :	– 0,26 dB

Il a été tenu compte, lors des mesures, d'une bande passante audio de 20 Hz à 20 kHz et, pour la source de signal, d'une impédance de 600 Ω.

l'aide de signaux numériques standard. La commutation des entrées se fait ici par l'intermédiaire des entrées SDA et SCL.

L'entrée SDA permet, en mode non bus I²C, la commutation entre la source audio/vidéo 1 (application du 12 V à l'entrée SDA) et la source audio/vidéo 2 (mise à zéro de cette entrée). L'entrée SCL permet le choix du facteur d'amplification (gain) de l'amplificateur vidéo. L'application à l'entrée SCL d'un niveau logique haut (+12 V) se traduit par un gain de 2, sa mise à zéro ramène ce gain à l'unité.

En cas de mise en parallèle de plusieurs TDA8440, on pourra utiliser l'entrée OFF pour mettre hors-fonction les circuits inactifs.
L'activation du TDA8440 se fait par mise au niveau logique bas de l'entrée OFF; inversement, la mise au niveau logique haut de cette entrée rend le circuit intégré inactif, faisant passer toutes ses sorties à haute impédance.

L'électronique

Nous avons fait en sorte, pour garantir à ce montage une flexibilité optimale, que le **commutateur audio/vidéo électronique** (et bien entendu aussi la platine dessinée à son intention) puisse être commandé tant à l'aide de boutons-poussoirs (touches Digitast par exemple) que par l'intermédiaire du bus I²C.
Nous en avons en outre fait un système modulaire de sorte que l'on pourra lui ajouter le nombre d'entrées additionnelles requises. En pratique, cela signifie qu'il est possible de connecter un nombre important de sources à la prise Péritel (SCART) d'un téléviseur par exemple.

La **figure 2** donne le schéma complet du montage, schéma que l'on pourra diviser en 2 parties. La première partie, alimentation et amplificateur du bas de la figure 2, est commune à tous les modules et ne sera donc réalisée qu'une fois. Le reste du circuit constitue le commutateur proprement dit y compris l'électronique de commande nécessaire. Chaque unité met à disposition 2 sets d'entrées (vidéo et audio stéréo) que l'on peut interconnecter aux sorties communes. Étant donnée leur caractéristique d'impédance élevée, les sorties des différents modules peuvent être branchées en parallèle. On peut ainsi utiliser simultanément plusieurs modules de sorte qu'il est toujours possible de disposer du nombre d'entrées vidéo requis.
Il faudra cependant veiller dans ce dernier cas à ne jamais surcharger l'alimentation, qui, telle que proposée ici est capable d'alimenter simultanément un maximum de 8 modules (16 entrées vidéo).
Bien qu'il existe des versions du régulateur 7812 capables de fournir 1,5 A – ce qui permettrait l'alimentation simultanée de 12 modules – nous ne recommandons pas cette approche sachant qu'avec le radiateur proposé dans la liste des composants le régulateur risquerait d'atteindre des températures excessives en raison de l'augmentation de dissipation qu'entraînerait l'utilisation de ce type de régulateur.

Revenons à notre schéma. Bien que son fonctionnement soit relativement simple, le circuit n'en nécessite pas moins un nombre relaticement

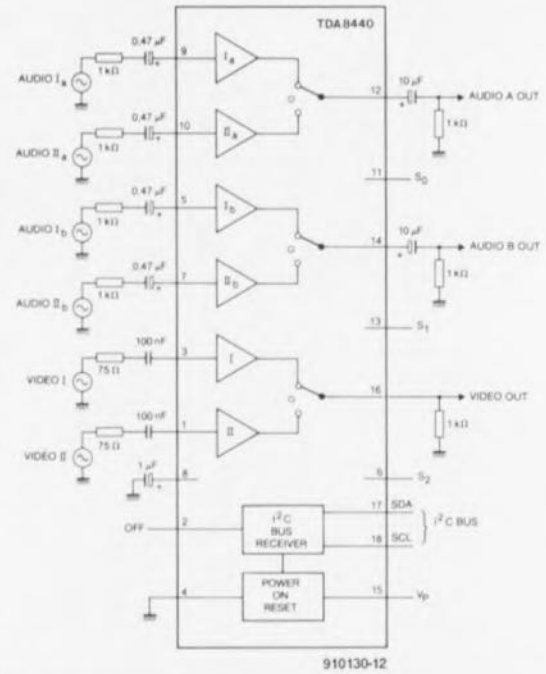
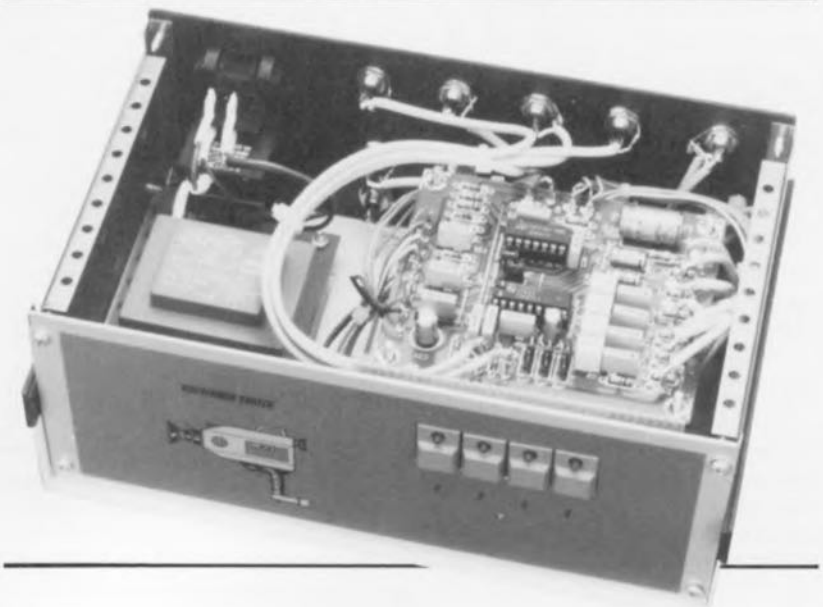


Figure 1. Synoptique de la structure interne du TDA8440, un commutateur audio/vidéo en un seul circuit lancé sur le marché par Philips.

important de composants. Cette situation est due pour une bonne part au grand nombre de composants nécessaires à la terminaison optimale, c'est-à-dire à l'obtention de l'impédance terminale correcte, des entrées vidéo et audio. Il nous faut en outre 2 bascules bistables entourées de quelques composants discrets pour pouvoir faire travailler le TDA8440 en mode «non bus I²C».

Si l'on veut utiliser le TDA8440 dans ce mode, il faut mettre les cavaliers de court-circuit **JP2** à **JP5** dans la position représentée sur le schéma.
Le cavalier **JP1** sera placé dans la



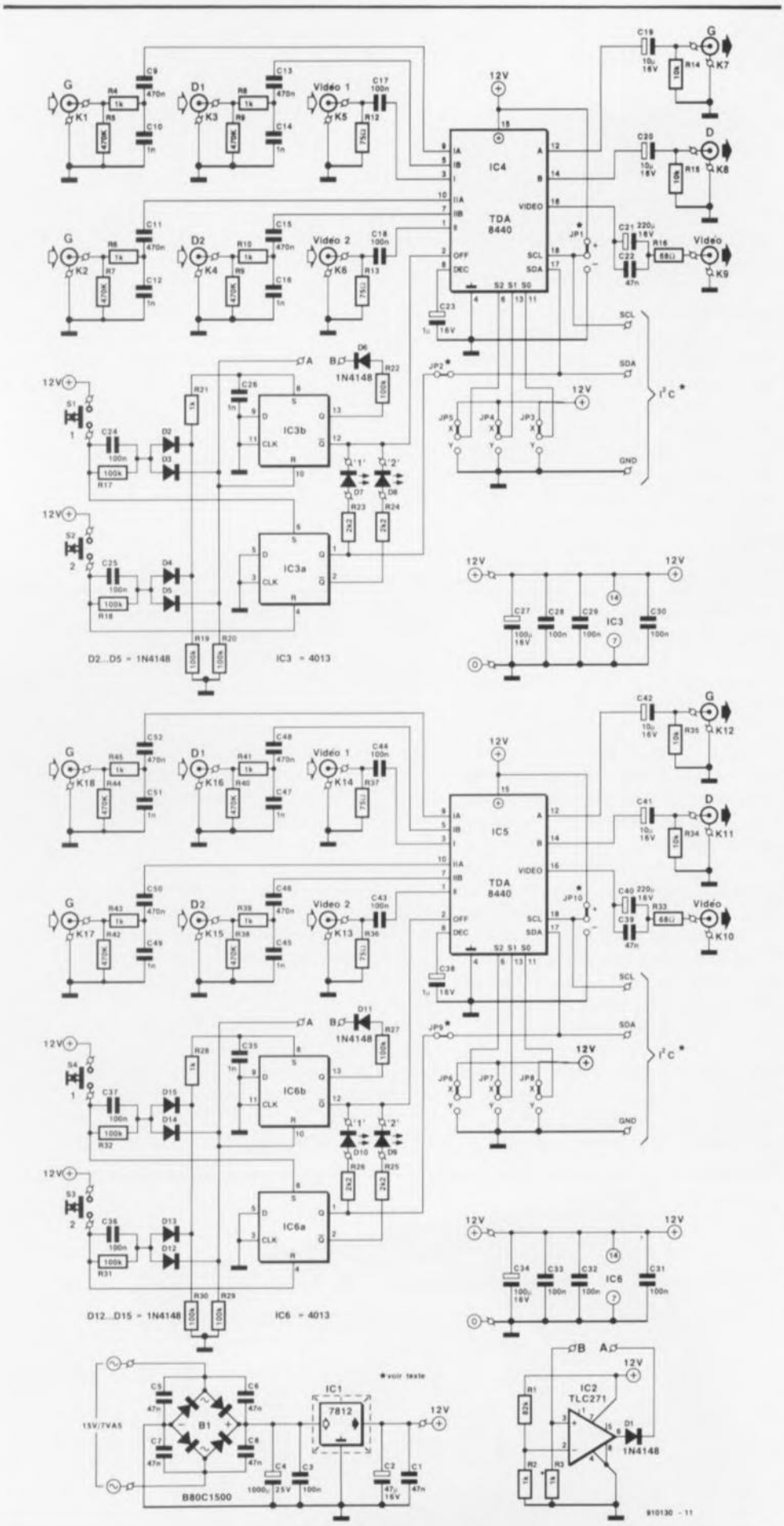


Figure 2. L'électronique du commutateur audio/vidéo dans toute sa simplicité. Nous aurions fort bien pu nous contenter de ne vous proposer que la moitié du circuit sachant que la seconde est parfaitement identique à la première. Chaque entrée comporte une connexion vidéo et une connexion audio stéréophonique.

position correspondant au facteur d'amplification requis pour l'amplificateur vidéo.
Cette remarque vaut bien entendu également pour la seconde partie du circuit basée elle sur IC5.

En cas d'action sur l'une des touches (de ce module ou sur l'une de celles de tout module additionnel) on a activation –via une porte OU (*OR*) réalisée en technologie discrète et constituée par les diodes D3 et D5 associées à la résistance R20– de la ligne de remise à zéro commune (**A**).
Toutes les bascules utilisées dans le circuit et reliées à cette ligne **A** sont alors remises à zéro de sorte que les sorties $\overline{Q}$ passent au niveau haut. Sur la seule platine du module dont un bouton-poussoir a été activé il se passe un processus additionnel. En cas d'action sur S1 ou S2, la bascule IC3b correspondante reçoit, outre une impulsion de remise à zéro, également une impulsion de positionnement (*set pulse*).

Cette impulsion de positionnement est d'une durée notablement supérieure à l'impulsion de remise à zéro générale, de sorte qu'en finale la bascule sera positionnée. Comme la sortie $\overline{Q}$ de cette bascule est reliée à l'entrée OFF de IC4, on a de ce fait activation du commutateur électronique. C'est là la raison pour laquelle seul le TDA8440 du module dont l'une des touches a été activée peut être actif. Tant qu'aucun des boutons-poussoirs n'a été actionné, c'est-à-dire immédiatement après la mise sous tension du système, tous les TDA8440 sont inactifs. Aucune des sources vidéo n'est alors reliée à la sortie du circuit.
On aura, en fonction de la touche actionnée, soit positionnement de IC3a (action sur S1), soit sa remise à zéro (action sur S2). Cette bascule détermine laquelle des sources audio/vidéo, 1 ou 2, est reliée à la sortie. Notons en outre la présence aux sorties des bascules de 2 LED chargées d'indiquer quelle est la source reliée à la sortie.

En cas d'action simultanée sur plusieurs touches, la tension appliquée à l'entrée non-inverseuse (+) du comparateur IC2 via les paires D6+R22 et D11+R27 atteint un niveau tel que l'on a à la sortie de cet amplificateur opérationnel un niveau logique haut entraînant la mise à «1» de la ligne **A**. Seule la dernière touche actionnée servira à la sélection d'une entrée.
Si l'on choisit de commander le montage par l'intermédiaire d'un

bus I²C, on supprimera les cavaliers de court-circuit **JP1** et **JP2** (**JP9** et **JP10**) et l'on implantera (ou non) les cavaliers **JP3** à JP5 (**JP6** à **JP9**) en fonction de l'adresse que l'on veut définir sur le bus I²C pour chacun des TDA8440. Comme nous le disions plus haut, il est possible ainsi de connecter un maximum de 7 modules sur un bus I²C. Si l'on opte pour cette solution, on pourra supprimer toute la logique de commande telle que les bascules, les boutons-poussoirs et les composants discrets connexes.

Pour en savoir plus en ce qui concerne les instructions de commande et l'utilisation des protocoles sur le bus I²C nous vous recommandons de consulter les recueils de caractéristiques IC12a et IC12b de Philips qui abordent tous ces sujets dans le moindre détail. Nous vous renvoyons également aux 2 articles consacrés au bus I²C publiés dans les numéros précédents d'Elektor.

Le schéma comporte la plupart des composants en double exemplaire : le premier composant prenant place sur la platine numéro 1, le second sur l'autre. Comme le montre la **figure 3**, la platine dessinée à l'intention de ce montage se subdivise en fait en trois sous-circuits imprimés, l'alimentation et 2 circuits de commutation.

La réalisation

Sachant que l'on travaille avec des signaux vidéo, il faut respecter un certain nombre de règles lors du dessin du circuit imprimé, ce qui explique la proximité de certains des composants par rapport au

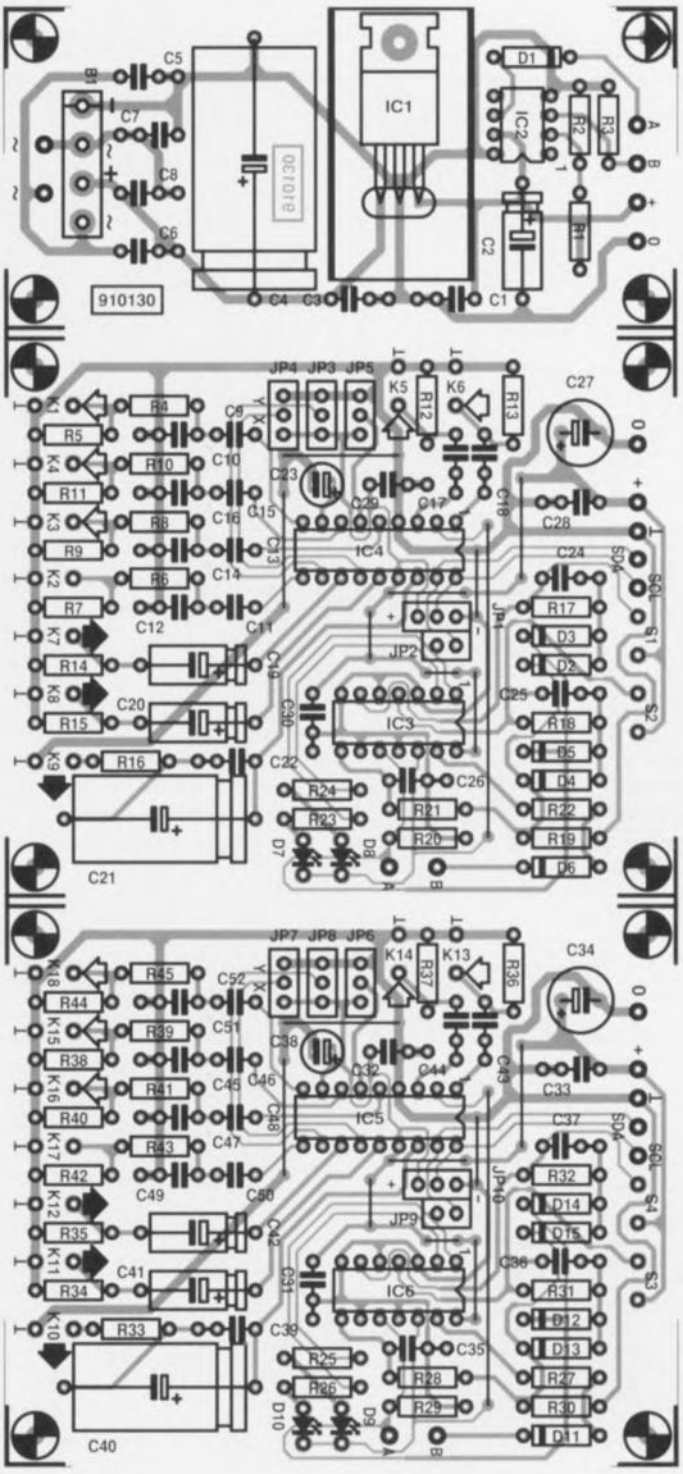


Figure 3. Représentation de la sérigraphie de l'implantation des composants du circuit imprimé dessiné à l'intention du commutateur audio/vidéo.

Liste des composants

Résistances :

R1 = 82 kΩ
R2,R3,R4,R6,R8,R10,R21,R28, R39,R41,R43,R45 = 1 kΩ
R5,R7,R9,R11,R38,R40, R42,R44 = 470 kΩ
R12,R13,R36,R37 = 75 Ω
R14,R15,R34,R35 = 10 kΩ
R16,R33 = 68 Ω
R17 à R20,R22,R27, R29 à R32 = 100 kΩ
R23 à R26 = 2kΩ2

Condensateurs :

C1,C5 à C8,C22,C39 = 47 nF céramique
C2 = 47 µF/16 V
C3,C17,C18,C24,C25,C28 à C33,C36,C37,C43,C44 = 100 nF
C4 = 1 000 µF/25 V
C9,C11,C13,C15,C46, C48,C50,C52 = 470 nF
C10,C12,C14,C16,C26,C35, C45,C47,C49,C51 = 1 nF
C19,C20,C41,C42 = 10 µF/16 V
C21,C40 = 220 µF/16 V
C23,C38 = 1 µF/16 V radial
C27,C34 = 100 µF/16 V radial

Semi-conducteurs :

D1 à D6,D11 à D15 = 1N4148
D7 à D10 = LED haute luminosité
B1 = B80C1500
IC1 = 7812
IC2 = TLC271 (Texas Instruments)
IC3,IC6 = 4013
IC4,IC5 = TDA8440 (Philips)

Divers :

JP1,JP3 à JP8,JP10 = barrette sécable à 3 contacts droits
JP2,JP9 = barrette sécable à 2 contacts droits
K1 à K18 = embase cinch fixation par écrou
S1 à S4 = touche de contact unipolaire à contact travail, Digitast par exemple
radiateur en U pour IC1

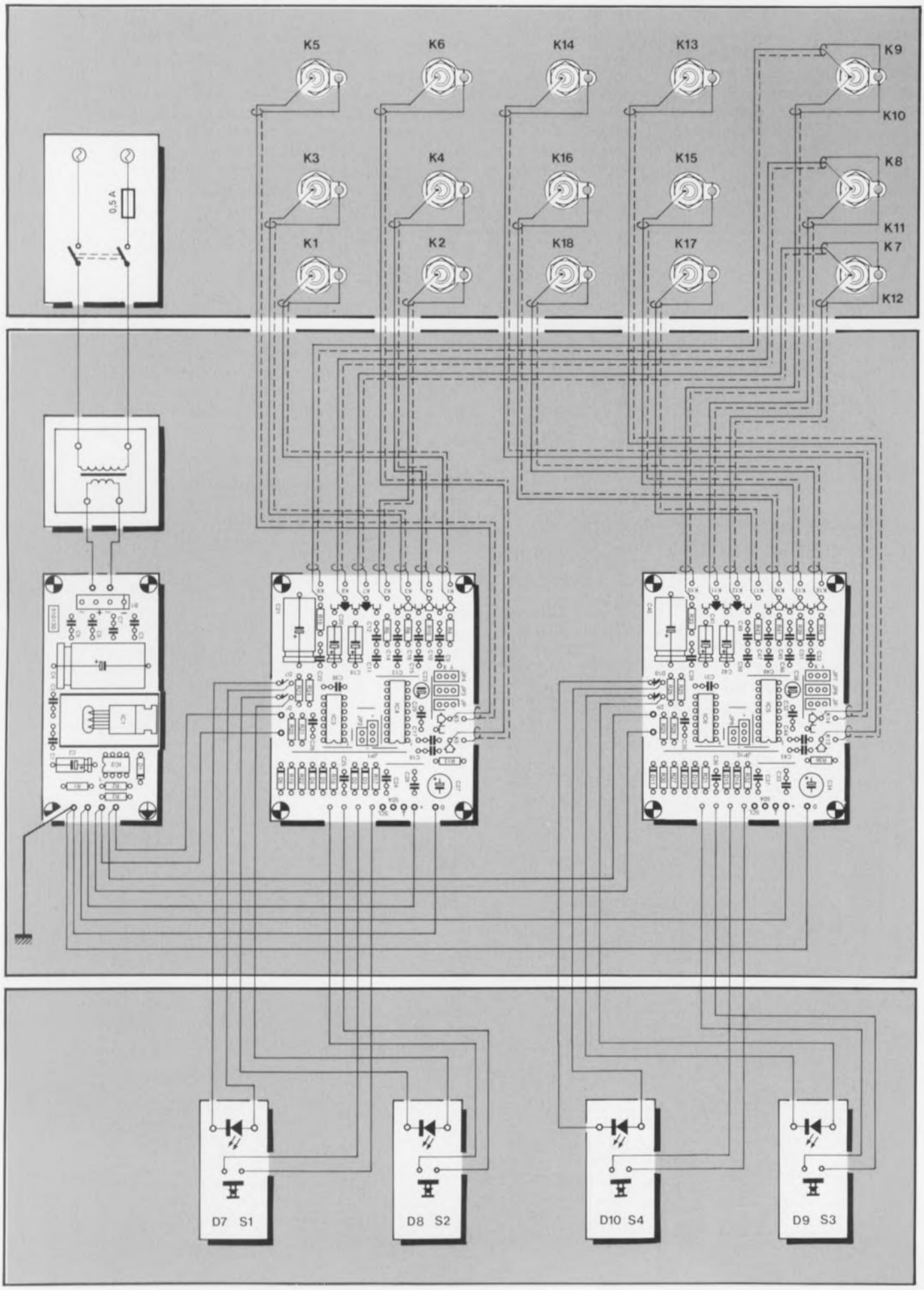


Figure 4. Plan de câblage de l'ensemble de la réalisation. On pourra, sans autre forme de procès, ajouter jusqu'à 6 modules additionnels aux 2 que comporte ce montage à l'origine.

TDA8440. Nos «sculpteurs de circuits imprimés» car c'est un peu ce que sont nos dessinateurs de platine, ont mis tout leur talent dans la conception du circuit imprimé dont la figure 3 vous propose la représentation de la sérigraphie de l'implantation des composants.

Comme nous le disions plus haut, il faudra, avant de commencer l'implantation des composants, découper la platine en 3 parties. Il suffira d'effectuer 2 traits de scie bien droits entre les points de repère présents. Une fois cette opération effectuée, on dispose donc d'une part d'une petite platine, l'alimentation commune associée au comparateur, utilisée par les 2 modules de commutation et de l'autre de 2 platines identiques, les modules de commutation proprement dits.

Commençons par la platine de l'alimentation : on y trouve les composants caractéristiques d'une alimentation, pont de redressement, condensateurs de lissage, régulateur intégré et condensateurs de filtrage, et un composant plus insolite, un amplificateur opérationnel, IC2.
Si l'on envisage de réaliser plus de 2 modules (4 entrées donc) on pourra mettre de côté la platine d'alimentation obtenue après découpage de la seconde grande platine ainsi que celle présente sur les platines suivantes. Il suffit en effet de disposer d'une seule alimentation. Mais ne jetez pas à la poubelle les platines excédentaires, car vu leur concept universel, il n'est pas exclu que vous trouviez une occasion de les utiliser.
Il est bien évidemment toujours possible d'envisager la réalisation de ce montage sur un morceau de platine d'expérimentation, encore que dans le cas présent il nous faille vivement déconseiller une telle approche. Il est en effet indispensable d'opter pour un dessin de circuit imprimé efficace si l'on veut éviter l'interaction entre les canaux et donc obtenir une diaphonie digne de ce nom. Pour peu que le câblage soit quelque peu maladroit, le résultat pourra être très décevant.
Après avoir terminé l'implantation des composants sur les 3 platines, avoir vérifié la qualité de réalisation de ces 3 circuits imprimés, on pourra penser à en effectuer l'interconnexion selon le plan de câblage représenté en **figure 4**. Comme on le voit, un vrai régal pour les plus «électriciens» d'entre nous. N'hésitez pas à faire une photocopie de la figure 4 pour passer au marqueur vert – ou jaune ou rouge, nous n'avons pas de préférence politique – les interconnexions déjà effectuées.

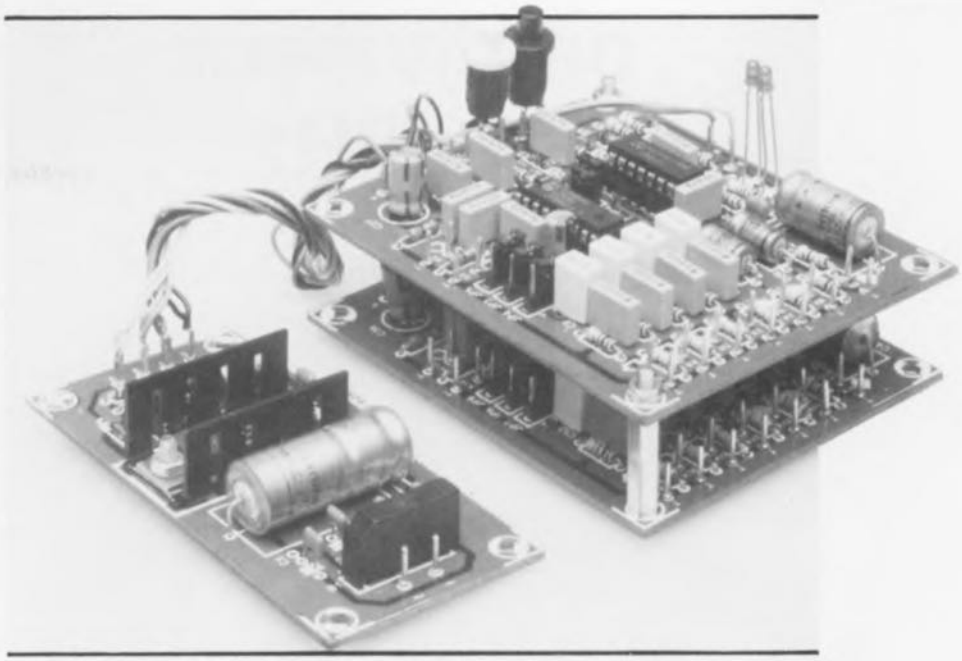


Figure 5. L'un de nos prototypes montés. L'approche «en sandwich» adoptée permet de garder au montage une compacité remarquable.

L'utilisation de touches «Digitast» à LED incorporée permettra de donner à votre commutateur audio/vidéo un aspect éminemment professionnel. Ces touches seront positionnées sur un petit morceau de platine d'expérimentation à pastilles fixé sur la face avant à l'aide de vis biseautées encastrées avant la mise en place du film plastique autocollant que l'on aura, par exemple, pu réaliser à partir de l'exemple proposé en **figure 6**. Grâce aux LED on voit instantanément quelle est la source de signal reliée à la sortie.

Peut-être s'agit-il là d'un petit montage audio/vidéo qui fera plaisir aux centaines de milliers d'amateurs de ce nouveau sport qu'est la «chasse aux images» effectuée à l'aide d'un caméscope ou autre magnétoscope portatif. ■

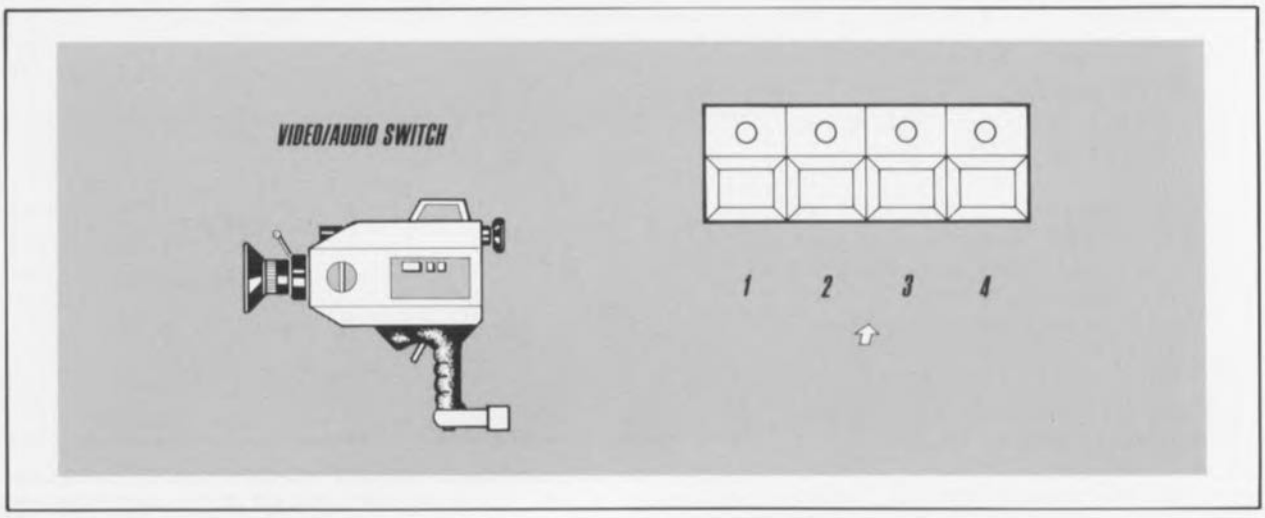
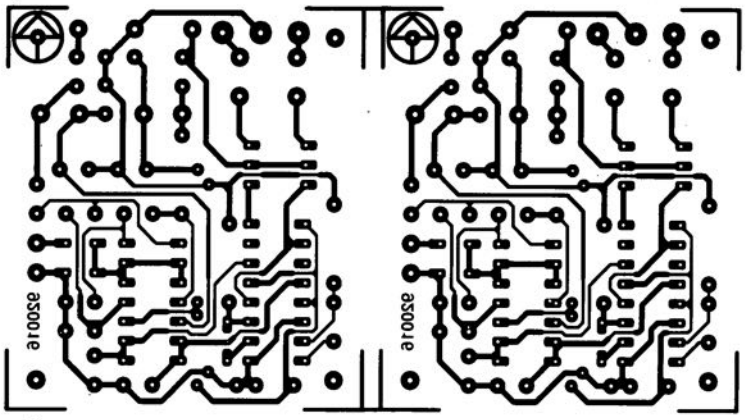


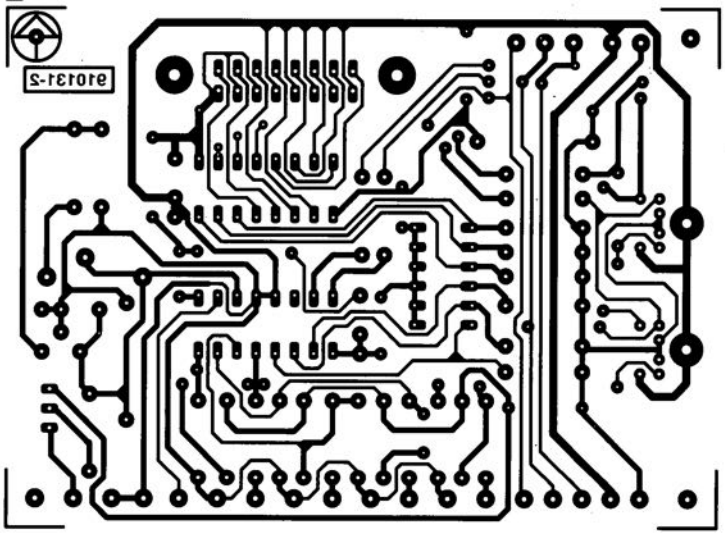
Figure 6. Exemple de dessin de face avant pour le commutateur audio/vidéo.

1 témoin d'écrêtage
2 convertisseur A/N-N/A pour I²C
3 convertisseur RS-232-A/N

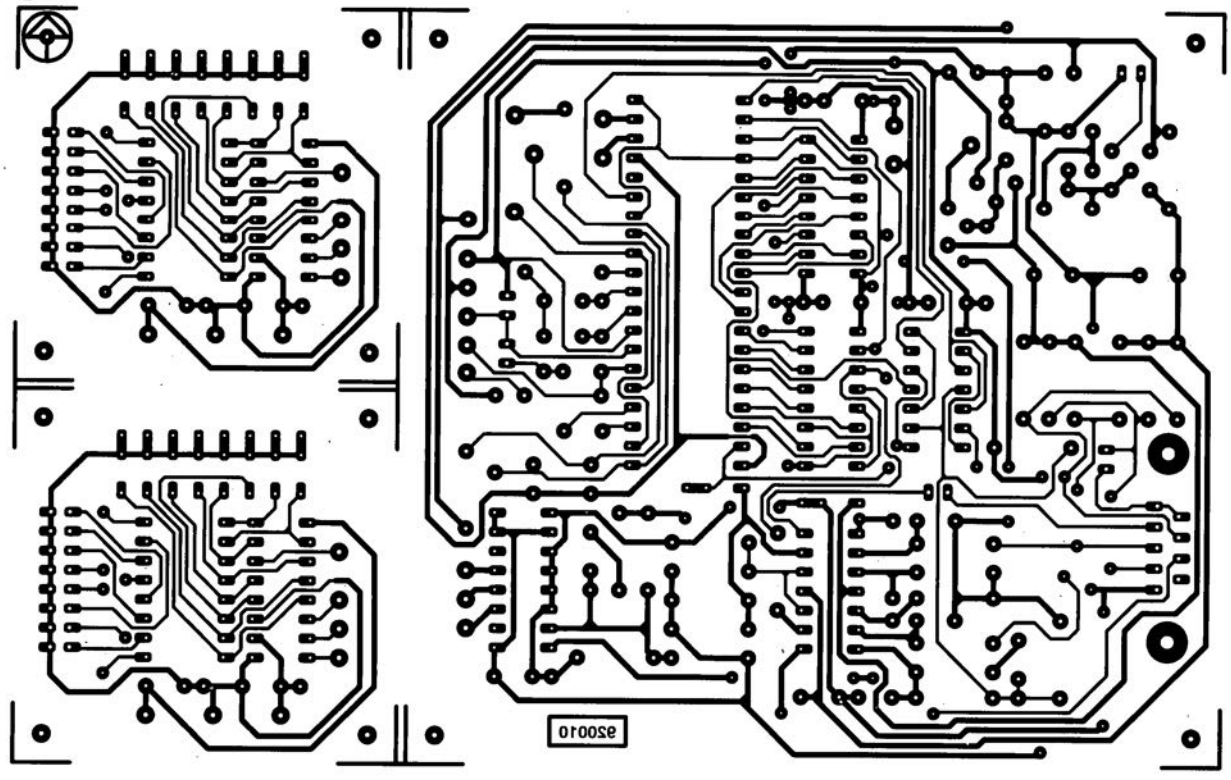
1



2

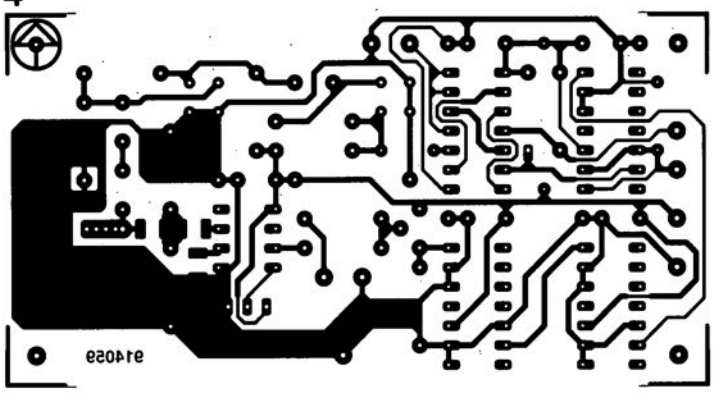


3

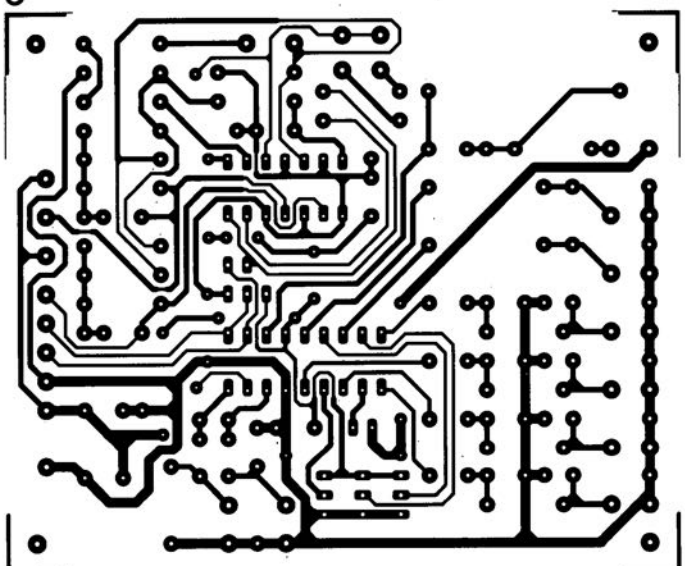


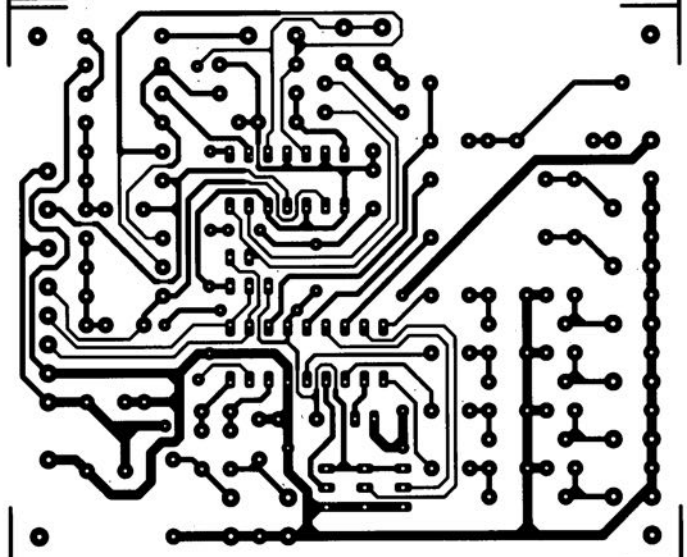
4 prédiviseur 1,3 GHz
5 commutateur audio/vidéo électronique
6 émulateur de 8751 : côté composants
7 émulateur de 8751 : côté pistes

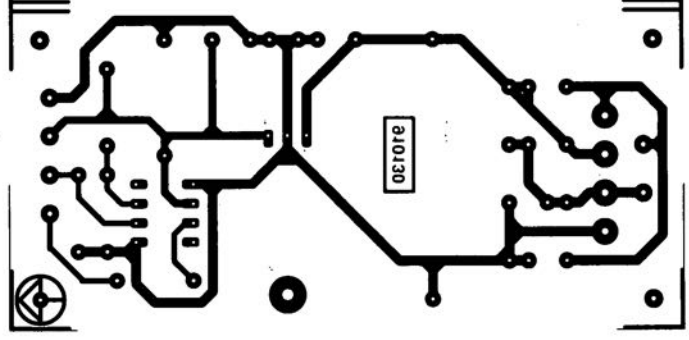
4



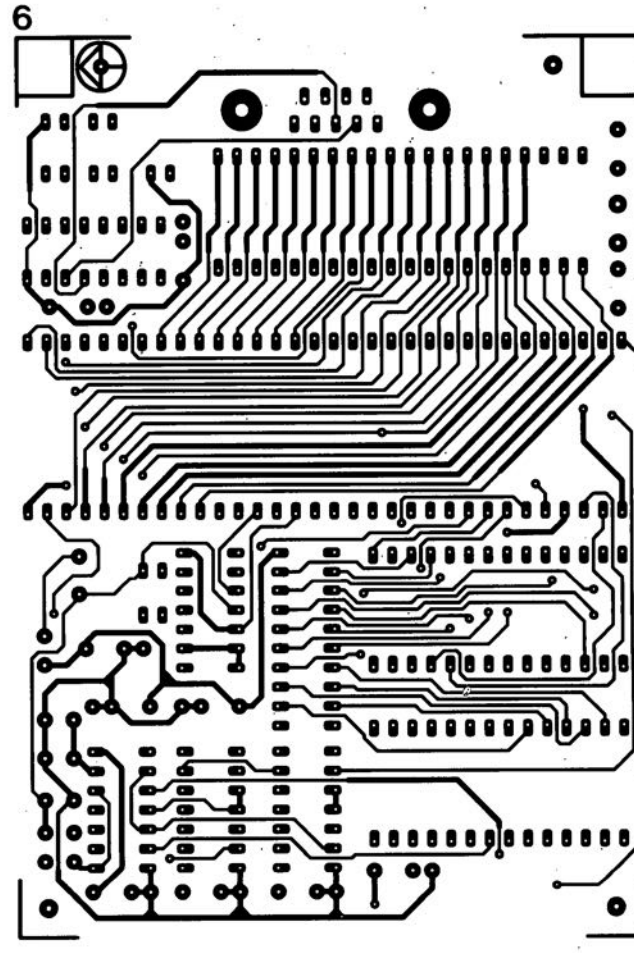
5



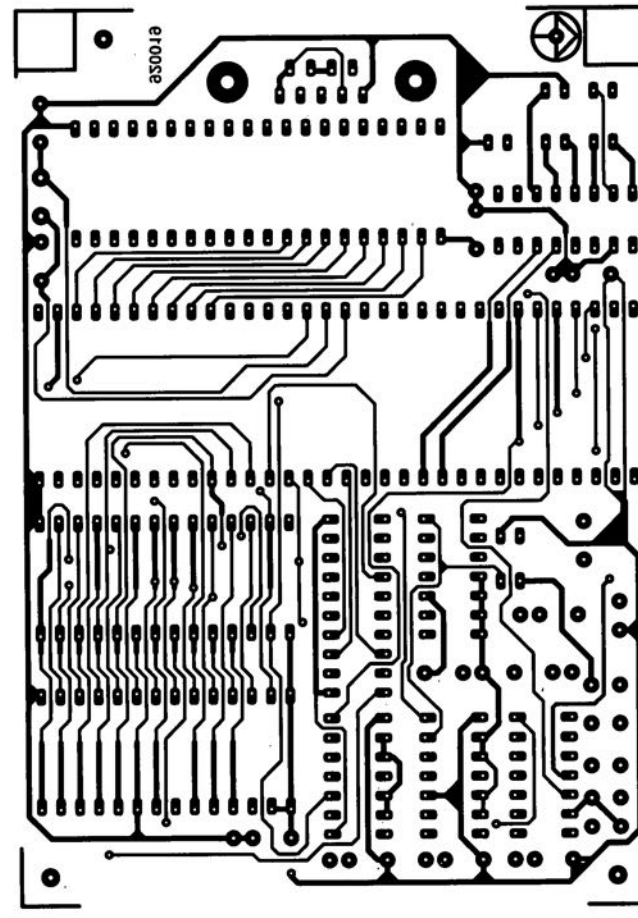




6



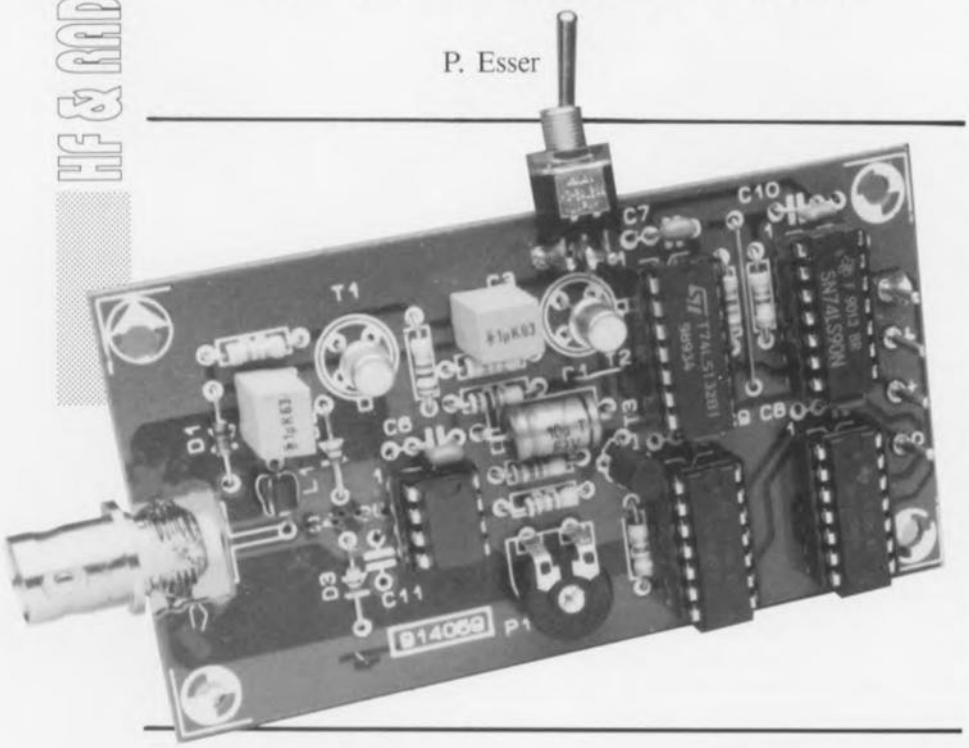
7



prédiviseur 1,3 GHz

signal UHF en entrée, niveau TTL en sortie

P. Esser



Le domaine de fréquences qu'est capable de traiter la plupart des fréquencemètres standard connaît une limite supérieure qui se situe aux environs de 10 MHz. Si on envisage, à l'occasion, de mesurer des fréquences plus élevées, on constate avec amertume qu'il est, en pratique, quasiment impossible de modifier quoi que ce soit à l'intérieur de ce type d'instruments : envolée l'idée de l'adjonction d'un convertisseur électronique. Il ne reste donc plus que la solution de faire appel à un montage auxiliaire.
Le prédiviseur, objet de cet article, vous permettra d'entrer dans le domaine des GHz car, même dans cette plage de fréquences, il fournit à sa sortie un signal rectangulaire de niveau TTL, avec lequel tout fréquencemètre digne de ce nom saura se «dépatouiller».

L'utilisation d'un prédiviseur présente un double avantage. Il permet d'une part d'élargir la plage de fréquences du fréquencemètre auquel il est connecté et, de l'autre, d'effectuer les mesures en utilisant des câbles notablement plus courts. Le prédiviseur peut en effet, en règle générale, être placé bien plus près de l'objet sur lequel se font les mesures que ne le serait le fréquencemètre lui-même (**figure 1**).

On notera cependant que l'utilisation du prédiviseur présente également un petit inconvénient : la sélection du calibre ne se fait plus sur le fréquencemètre lui-même, le prédiviseur étant doté d'un commutateur servant à choisir le calibre de mesure requis.

Les diviseurs

Un circuit intégré spécialisé pour ce type d'applications se charge de la division initiale du signal d'entrée. De tels circuits de prédivision figurent dans la gamme d'un nombre important de fabricants de circuits intégrés. L'électronique du prédiviseur de cet article fonctionne par exemple très bien, soit avec un U664B, soit avec un SDA4211. Les **figures 2a** et **2b** montrent les synoptiques de la structure interne de chacun de ces 2 composants.

Le U664B de Telefunken a été conçu pour une utilisation dans les synthétiseurs de fréquences des téléviseurs. Ce composant effectue une division par 64 en se contentant d'une circuiterie externe dérisoire. En absence de signal d'entrée, il oscille dans la plage supérieure des fréquences. La circuiterie externe est incroyablement simple : en effet, il suffit de 2 petits condensateurs additionnels.

Outre le U664B dont nous venons tout juste de parler, on pourra également faire appel au SDA4211 de Siemens. Ce second circuit intégré met à la disposition de l'utilisateur 2 facteurs de division : division par 64 ou par 256. Le choix du facteur de division s'effectue par l'intermédiaire de la broche 5 du composant. La mise à +5 V de cette broche se traduit par une division par 64. Si au contraire cette broche est reliée à la masse, on aura une division par 256.
De manière à vous permettre de mettre également à profit les possibilités offertes par le SDA4211, le circuit imprimé dessiné à l'intention du prédiviseur peut être doté d'un cavalier de codage, adjonction qui

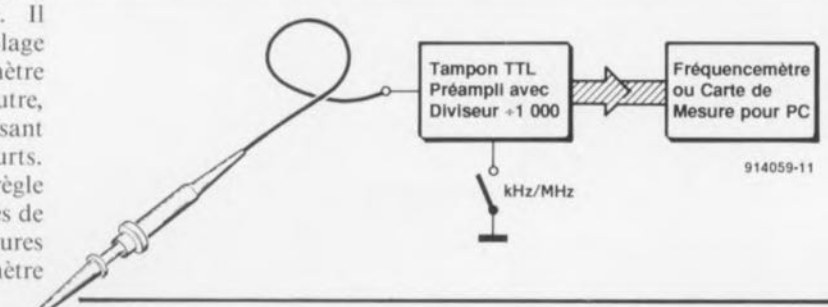


Figure 1. Ce petit synoptique illustre l'interconnexion du prédiviseur à un fréquencemètre existant. L'interrupteur dont est doté le prédiviseur permet de choisir entre un signal de mesure de rapport 1 : 1 (non divisé) ou 1 : 1 000 (divisé par 1 000).

en facilite sensiblement la configuration. On dispose ainsi de la possibilité de relier la broche 5, soit au +5 V («+» de **JP1**), soit à la masse («−» de **JP1**).

Dans le montage présenté ici, les 2 circuits intégrés mentionnés plus haut, sont compatibles à 100%, tant du point de vue fonctionnel que de leur brochage : une fois n'est pas coutume, ils sont compatibles broche à broche. En règle générale on peut toujours remplacer un U664B par un SDA4211. Inversement, on ne peut remplacer un SDA4211 par un U664B que si, et uniquement si, le SDA4211 fonctionnait en diviseur par 64 !

2 trajets

On voit, sur le schéma de la **figure 3**, que le signal qui arrive à l'embase d'entrée bifurque en aval de celle-ci. La partie inférieure du schéma représente l'électronique de l'étage de prédivision; celle du haut celles d'un étage de traitement de signal et d'un étage amplificateur.

Les 2 diodes D2 et D3 montées en tête-bêche et placées en aval du condensateur de couplage C4, limitent l'amplitude du signal d'entrée à une valeur maximale de ±0,7 V. La tension d'entrée est appliquée, à travers un second condensateur CMS (pour montage en surface), C5, à l'entrée, broche 2, du circuit intégré de prédivision, IC3. De par la connexion à la masse −à travers le condensateur CMS C11−, de la seconde entrée de ce composant, sa broche 3, l'entrée symétrique à l'origine de ce composant se voit associée à une circuiterie asymétrique.
La petite barrette à 3 contacts mâles, **JP1**, avec son cavalier de court-circuit, connectant la broche 5 de IC3 soit à la masse soit à la tension positive de 5 V, permet de choisir, nous l'avons mentionné plus haut, le facteur de division requis (à condition bien entendu cependant que l'on utilise le SDA4211 de Siemens).

Le signal de mesure est disponible, après une division par 64, à la broche 6 de IC3 et arrive au diviseur de tension réalisé à l'aide des résistances R7 et R8 et l'ajustable P1. On trouve, en aval de ce diviseur de tension, un étage amplificateur centré sur le transistor T3, un BF324. Le signal arrive ensuite, après avoir été dérivé de l'une des bornes de la résistance R9 prise dans

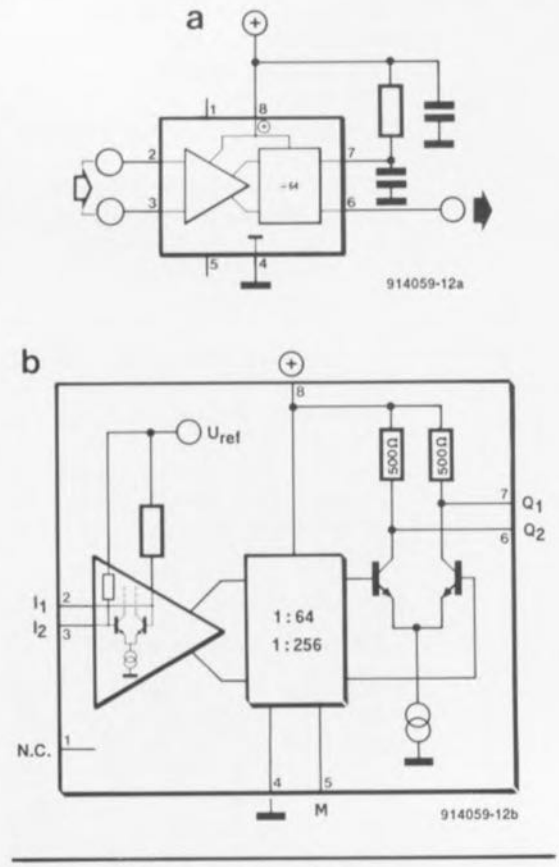


Figure 2. La figure 2a donne le synoptique de la structure interne du U664B de Telefunken; la figure 2b celui de celle du SDA4211 de Siemens.

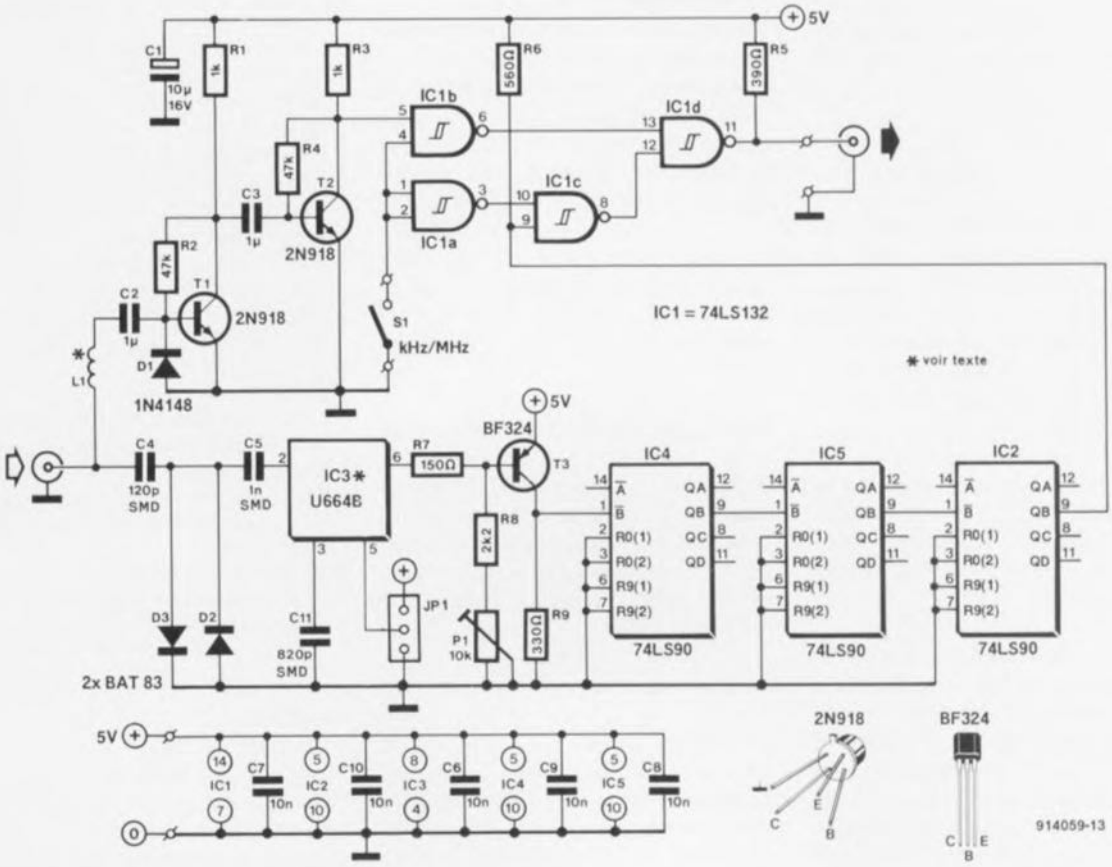


Figure 3. L'électronique du prédiviseur 1,3 GHz.

la ligne du collecteur de T3, à une chaîne de diviseurs, réalisée par la mise en série de 3 compteurs décadiques du type 74LS90 (IC2, IC4 et IC5).
Chacun de ces circuits intégrés divise le signal par 2,5. Un coup d'oeil sur la table de vérité donnée dans le **tableau 1**, aura vite fait de vous faire comprendre pourquoi il est nécessaire d'utiliser ce facteur de division peu courant :

La moitié supérieure du 74LS90 divise par 5. Au cours de 5 impulsions d'entrée la sortie Q_B du compteur passe 2 fois au niveau logique «1», ou formulé autrement, la sortie Q_B fournit, toutes les 2,5 impulsions d'entrée, 1 impulsion de sortie.

Tableau 1 Table de vérité

Impulsion	Sortie			
	Q_B	Q_D	Q_C	Q_B
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

À la fin de cette chaîne de diviseurs, nous disposons donc d'un signal de mesure, correspondant au signal d'entrée divisé par

$64 \cdot 2{,}5 \cdot 2{,}5 \cdot 2{,}5 = 1\,000.$

Le second itinéraire du signal dans ce montage passe d'abord par la self L1 et le condensateur de couplage C2, avant d'arriver à la base d'un transistor VHF, T1. Ce transistor, un 2N918, monté en émetteur commun, se comporte en fait comme un amplificateur opérationnel inverseur. Le gain en tension introduit par cet étage correspond, à peu de choses près, au gain à vide du transistor, sachant cependant qu'il dépend également de l'impédance de la source.

La diode D1 limite d'éventuelles amplitudes négatives du signal à une valeur maximale de −0,7 V. Le signal de sortie de T1 est dérivé de son collecteur avant de subir, par l'intermédiaire du transistor T2 (lui aussi monté en émetteur commun), une nouvelle amplification qui correspond, à nouveau, au gain à vide de ce transistor.
Le signal arrive ensuite à la porte NAND (NON-ET) à trigger de Schmitt, IC1b. Les 4 portes NON-ET garantissent des flancs nets et effectuent une intégration des signaux de mesure arrivant par les 2 trajets différents.

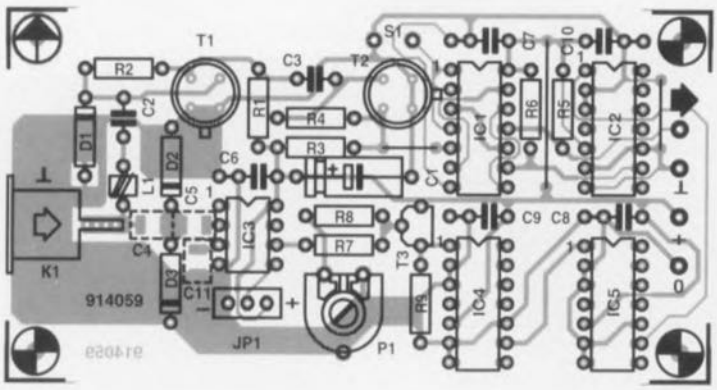


Figure 4. Représentation de la sérigraphie de l'implantation des composants du prédiviseur 1,3 GHz.

Si l'interrupteur S1 est ouvert, le signal de mesure non divisé de la partie supérieure du schéma, arrive à l'embase de sortie. Dans le cas inverse –l'inverseur S1 est fermé– c'est le signal de mesure divisé que l'on retrouve à l'embase de sortie.

Réalisation

La disposition des différents composants sur le circuit imprimé, dont on retrouve la sérigraphie en **figure 4**, n'est pas critique. Seul le montage des composants de type CMS (pour montage en surface) du circuit d'entrée exige un certain soin.

Il vous faudra bobiner vous-même la self L1. Elle comporte 2 à 3 spires de fil de cuivre émaillé de 0,4 mm de section effectuées sur une petite perle de ferrite.

On notera que la platine est conçue pour recevoir à l'entrée d'une embase BNC encartable qui vient s'encastrer dans la découpe effectuée à cette intention. Cette solution évite de devoir faire appel à du câble blindé pour relier l'embase à l'entrée du circuit.

La présence et le positionnement du cavalier de court-circuit **JP1** –caractéristiques qui, comme le prouve la photo en début d'article, prenaient sur notre prototype la forme d'un inverseur– dépendent du type de circuit intégré utilisé pour IC3.
Si vous faites appel au U664B de Telefunken, la présence du cavalier de codage est parfaitement superflue.
L'utilisation du SDA4211 de Siemens nécessite la connexion de la broche 5 de ce composant à la tension positive de 5 V. Il faudra donc connecter le contact du milieu de la petite barrette au contact «+». En cas d'erreur, pas de panique, cela n'a pas de conséquence grave. Le

Liste des composants

Résistances :
R1,R3 = 1 kΩ
R2,R4 = 47 kΩ
R5 = 390 Ω
R6 = 560 Ω
R7 = 150 Ω
R8 = 2kΩ2
R9 = 330 Ω
P1 = 10 kΩ ajust. horizontal

Condensateurs :
C1 = 10 µF/16 V
C2,C3 = 1 µF
C4 = 120 pF CMS
C5 = 1 nF CMS
C6 à C10 = 10 nF
C11 = 820 pF CMS

Semi-conducteurs :
D1 = 1N4148
D2,D3 = BAT 83 (Philips)
T1,T2 = 2N918
T3 = BF324
IC1 = 74LS132
IC2,IC4,IC5 = 74LS90
IC3 = U664B (Telefunken) **ou** SDA4211 (Siemens)*
* = voir texte

Divers :
L1 = 2 à 3 spires de fil de cuivre émaillé de 0,4 mm de section sur perle de ferrite
S1 = interrupteur unipolaire
K1 = embase BNC femelle coudée encartable
JP1 = barrette sécable à 3 contacts mâles avec cavalier de codage (voir texte)

```
200  GOSUB 3000
300  REM -----------------------------------------------------
301  REM BOUCLE DE COMPTAGE
305  FOR H = 1 TO 2
306  OUT 771,147
310  OUT 770,255
320  OUT 770,0
330  OUT 771,155
340  FOR I = 0 TO 4401: NEXT I
350  A = INP (768)
360  B = INP (769)
370  C = INP (770)
375  IF C >= 128 THEN C = C - 128
380  A = A * 256 * 256 * 128
390  B = B * 256
400  C = C * 256 * 256
410  A = A / 1000000!
420  B = B / 1000000!
430  C = C / 1000000!
440  FF = ((A + B + C) / 256) * 1000 * 1000 *1000
445  F = F + FF
450  NEXT H
455  F = F / 2 : FF = 0
460  F - (INT(F /696!)) /1000
470  LOCATE 10,35: PRINT "            ":LOCATE 10,35: PRINT F
480  GOSUB 1000
500  GOTO 305
1000 REM -----------------------------------------------------
1005 REM AFFICHAGE ANALOGIQUE
1010 LOCATE 19,15:PRINT"
                    "
1020 S = (F / 1000) * 5
1030 LOCATE 19,15+S  :PRINT CHR$(24)
1050 IF INKEY$ = "f" THEN GOTO 5000
1060 RETURN
3000 REM -----------------------------------------------------
3001 REM CREATION DE L'IMAGE
3010 CLS
3012 LOCATE 3,26: PRINT "FREQUENCEMETRE -> 1,3 GHZ"
3015 LOCATE 4,6: PRINT CHR$(201): LOCATE 4,75: PRINT CHR$(187)
3020 LOCATE 4,7
3030 FOR I = 1 TO 68: PRINT CHR$(205);: NEXT I
3040 FOR I = 5 TO 20:LOCATE I,6:PRINT CHR$(186) :LOCATE I,75:
     PRINT CHR$(186) : NEXT I
3050 LOCATE 21,7
3060 FOR I = 1 TO 68: PRINT CHR$(205);: NEXT I
3070 LOCATE 21,6: PRINT CHR$(200): LOCATE 21,75: PRINT CHR$(188)
3072 LOCATE 15,6:PRINT CHR$(199):LOCATE 15,75:PRINT CHR$(182)
3074 LOCATE 15,7:FOR I = 1 TO 68: PRINT CHR$(196);: NEXT I
3080 LOCATE 18,15
3090 FOR I = 1 TO 10: PRINT CHR$(197);CHR$(196);CHR$(196);
     CHR$(196);CHR$(196);: NEXT I: LOCATE 18,E65: PRINT CHR$(197)
3100 LOCATE 17,15:PRINT "0    1    2    3    4    5    6    7
     8    9    10"
3110 LOCATE 19,15: PRINT CHR$(24)
3200 LOCATE 10,35: PRINT "00000000"
3210 LOCATE 22,8: PRINT " f = fin"
3300 RETURN
5000 CLS:END

914059f.wp
```

Figure 5. Listing d'un petit programme permettant la visualisation d'une fréquence sur un PC doté du fréquencemètre 1 GHz encartable pour PC décrit dans le n°135 (septembre 1989).

seul problème rencontré alors est que le facteur de division défini ne sera pas le bon. Si vous reliez la broche 5 du SDA4211 à la masse, le facteur de division sera en effet de :

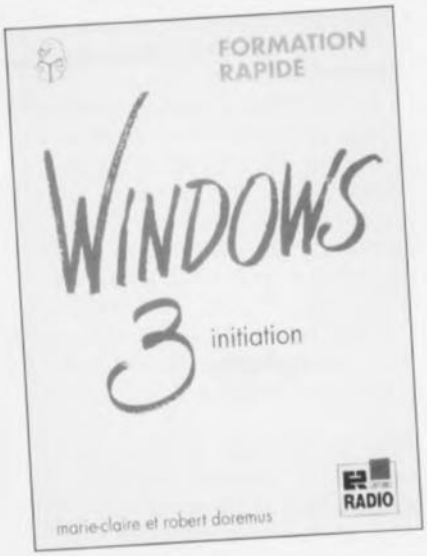
$256 \cdot 2{,}5 \cdot 2{,}5 \cdot 2{,}5 = 4\,000$,

c'est-à-dire 4 fois trop important.

Ceci termine donc la description d'un petit montage, très abordable tant du point de vue de la réalisation que de celui de la technique, permettant de mesurer des fréquences très élevées à l'aide d'un fréquencemètre «standard». En bref, un outil indispensable à tout électronicien qui s'égare de temps à autre dans le domaine des signaux HF, télévision RDS oblige. ◀

Windows 3, initiation

M.C. & R. Doremus



Bien que Windows 3 reste réservé à ''l'élite'' des possesseurs de PC, 386 et au-delà, soyons réaliste, est-il supportable de devoir attendre quelques secondes, voire plus, entre 2 images d'écran lorsque l'on passe d'un programme à l'autre, c'est devenu le programme d'exploitation d'environnement numéro un, et si cela tenait uniquement à Mr Bill Gates de Microsoft, il le resterait. Cet ouvrage fait partie de la catégorie ''Formation rapide'' de cet éditeur, ce qui sous-entend que l'on ne s'intéresse qu'à l'essentiel, car comment veut-on parler de tout en 160 pages? Rassurez-vous cependant, les auteurs, formateurs professionnels dans de grandes entreprises, ont mis au point la pédagogie et le contenu de cet ouvrage en le testant d'abord sous forme de support de cours. Ce livre a donc pour ainsi dire, subi d'abord l'épreuve du feu sur le terrain. Quelle meilleure garantie peut-on espérer d'une bonne adaptation aux attentes et aux difficultés du débutant?
Il s'agit pas non plus de se perdre dans les détails techniques secondaires.

Ceci explique que l'utilisateur chevronné de Windows 3 ne trouvera pas dans ce livre une réponse à toutes ses questions, telles par exemple à celle de la raison de cette impossibilité de choisir librement d'attribuer le port COM4 au modem sans entrer en conflit avec Windows 3, discussion au cours de laquelle ce logiciel a toujours le dernier mot. Si si, il existe une solution à ce problème.
Le débutant au contraire trouvera tout ce qu'il lui faut pour se lancer dans le monde fascinant de Windows 3.

prix 95 FF
Editions Radio
11, rue Gossin
92543 Montrouge

ELEKTURE

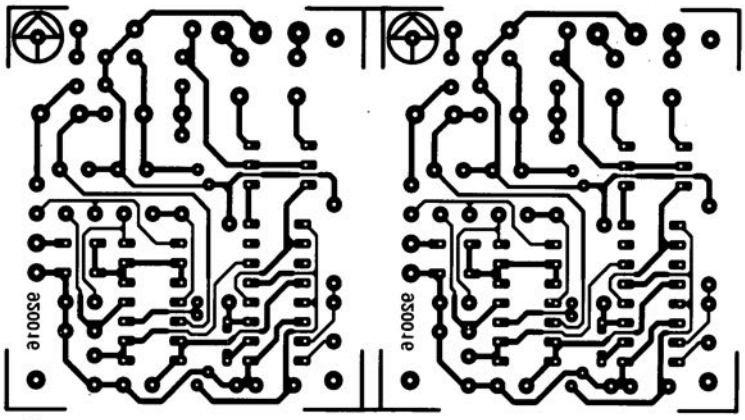
Si vous avez un problème dont vous ne pouvez pas trouver la solution, n'hésitez pas à contacter le serveur d'Elektor en faisant

3615
+
Elektor

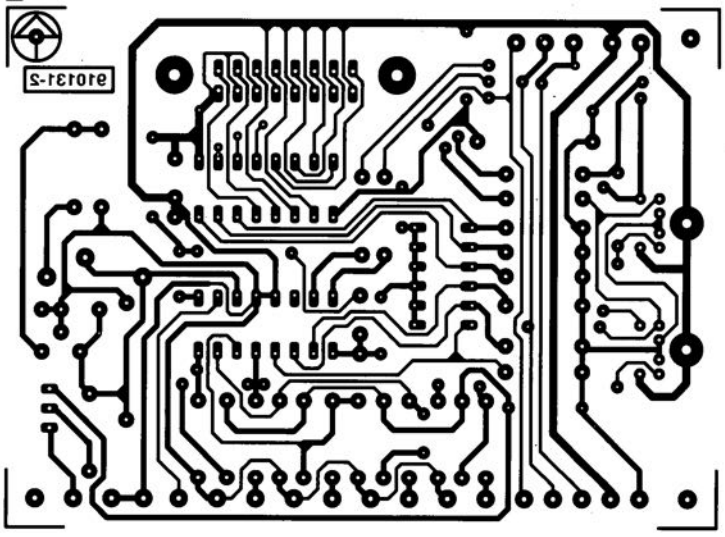
il n'est pas exclu que vous trouviez une main secourable.

1 témoin d'écrêtage
2 convertisseur A/N-N/A pour I²C
3 convertisseur RS-232-A/N

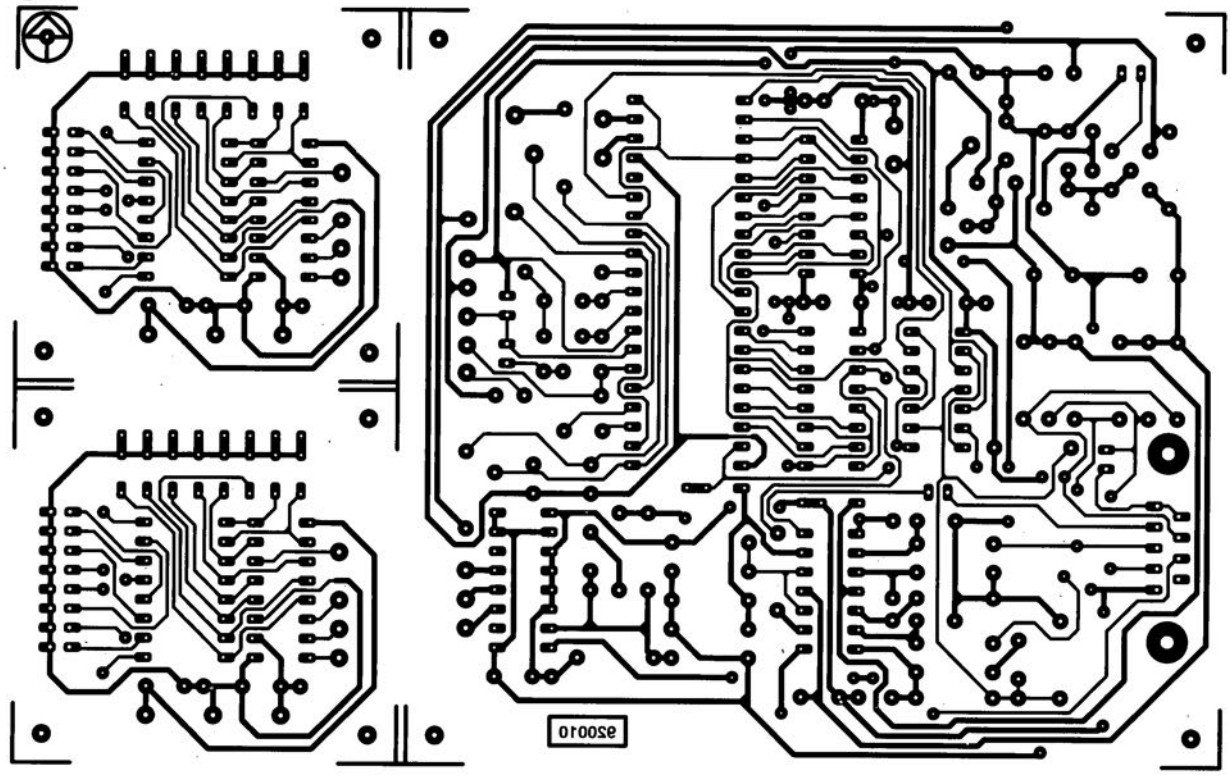
1



2

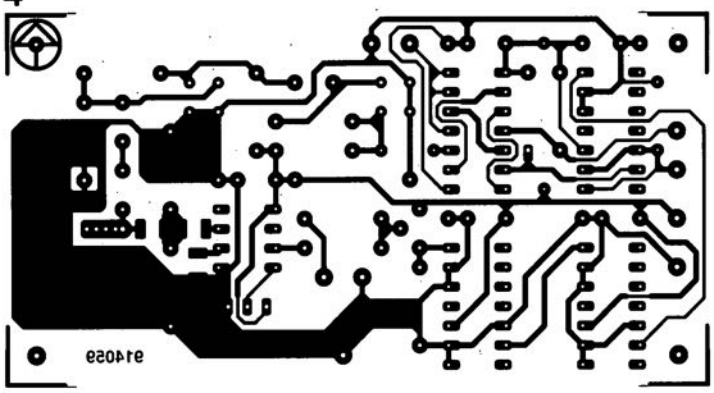


3

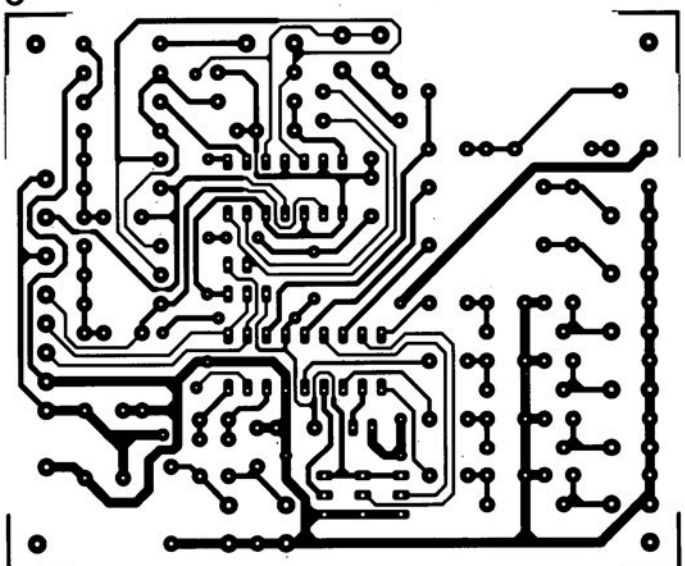


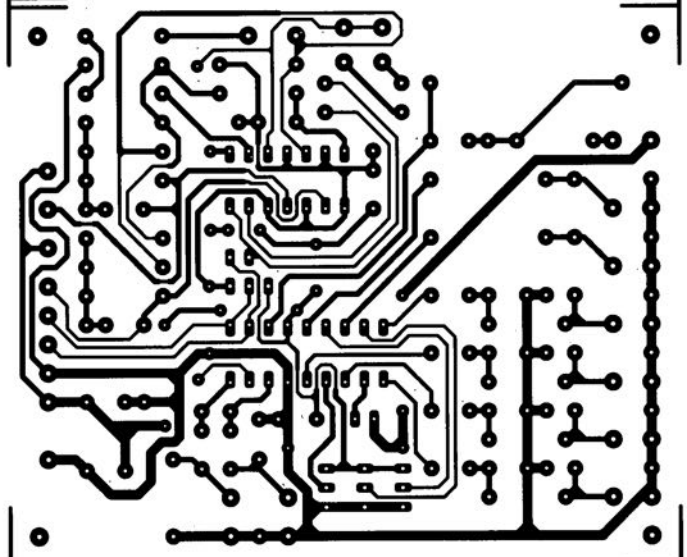
4 prédiviseur 1,3 GHz
5 commutateur audio/vidéo électronique
6 émulateur de 8751 : côté composants
7 émulateur de 8751 : côté pistes

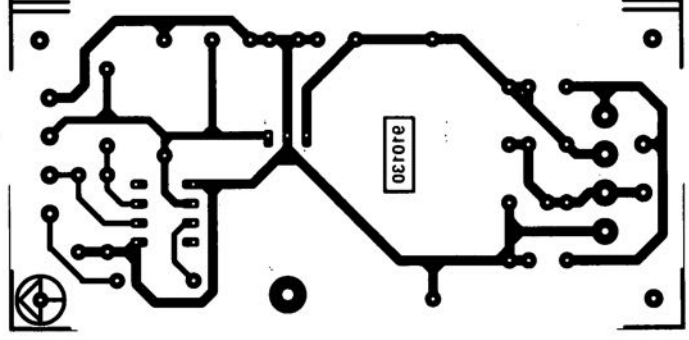
4



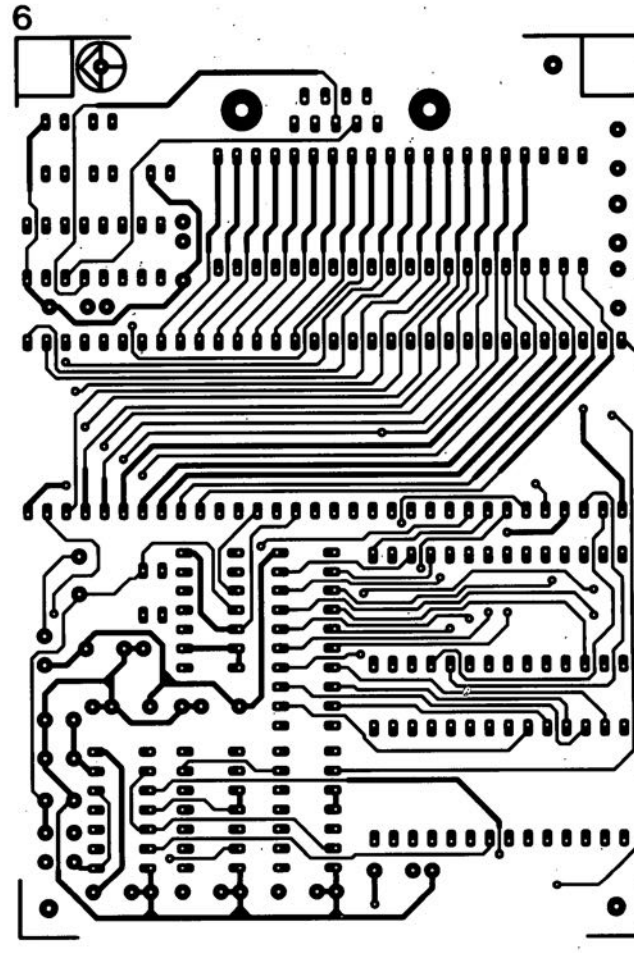
5



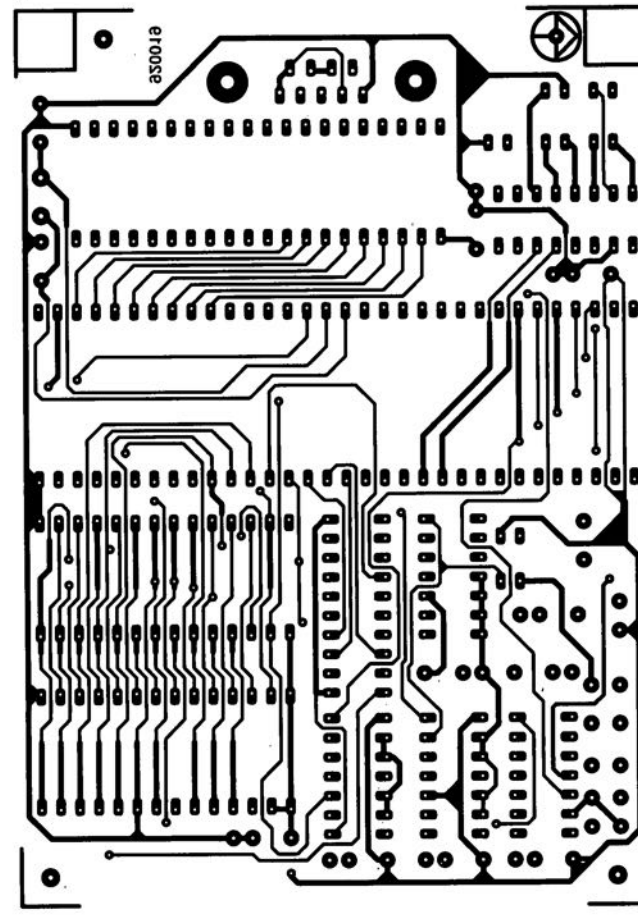




6



7



cours μC–8051 & assembleur

4e partie : instructions arithmétiques

M. Ohsmann

Ce nouveau chapitre de notre cours de programmation en assembleur sera consacré à la description des dernières instructions du μC–8051. Il ne nous reste, en fait, qu'à vous faire faire la connaissance des instructions arithmétiques, périple tout au long duquel nous vous proposerons quelques techniques de programmation qui vous permettront d'effectuer, au niveau de l'assembleur, des calculs rudimentaires.

L'un des exemples de programmation, destiné à la mesure de capacité (voire de durée), servira à mettre en pratique ces nouvelles connaissances de programmation. À la fin de cet article vous devriez être capable de programmer tout seul un générateur de bruit simple.

Addition

Pour effectuer l'addition de 2 valeurs à 8 bits –des octets donc–, le 8051 connaît l'instruction suivante :

ADD	A,octet opérande	;addition de l'octet opérande au ;contenu de l'accumulateur A

Bien que cette instruction ne semble pas poser de problème particulier, c'est, comme disent les Latins, dans la queue que se trouve le venin («*in cauda venenum*») ! Le venin caché dans cette instruction se manifeste par un effet sur les indicateurs.

Les conditions concernant les différents indicateurs sont les suivantes :

■ Le bit de retenue **CY** (ou plus court **C**) est positionné s'il se produit une retenue sur le bit d'ordre 7. Dans le cas contraire le bit de retenue **CY** est remis à zéro. Ce bit fournit donc une information bien claire concernant l'existence, ou non, d'une retenue entraînée par l'addition de nombres non signés compris entre 0 et 255.

■ L'indicateur auxiliaire **AC** (***A**uxiliary **C**arry*) destiné à l'addition de nombres BCD (***B**inary **C**oded **D**ecimal* = décimal codé binaire) est positionné s'il se produit une retenue sur le bit d'ordre 3. En absence de retenue ce bit est remis à zéro.
L'instruction **DA A** (voir plus loin) fait appel à l'indicateur auxiliaire.

■ En cas de dépassement sur le bit d'ordre 6 **et**, simultanément, l'absence de dépassement sur le bit d'ordre 7, l'indicateur de dépassement **OV** (***OV**erflow*) sera à 1 (positionné donc). Cet indicateur sera également positionné dans le cas contraire : dépassement sur le bit d'ordre 7 **et** absence de dépassement sur le bit d'ordre 6.
Dans tous les autres cas cet indicateur sera remis à zéro.
L'indicateur de dépassement **OV** sert donc à signaler qu'un dépassement a eu lieu lors de l'addition de nombres signés allant de –128 à 127.

La signification des indicateurs dépend donc de l'interprétation du contenu d'un octet, soit comme nombre non signé, soit comme nombre signé, soit encore comme nombre BCD. Le paragraphe suivant est de ce fait consacré aux différentes représentations des chiffres et des nombres.

Représentation de nombres

1. Nombres non signés

L'interprétation d'un octet comme nombre non signé correspond à la représentation connue d'un nombre en notation binaire. On attribue, en commençant avec le bit 0, successivement les valeurs 1, 2, 4, 8, 16 etc, aux différents bits de l'octet. Le nombre représenté résulte donc en :

$$(128 \cdot \text{bit } 7) + (64 \cdot \text{bit } 6) + \ldots + (4 \cdot \text{bit } 2) + (2 \cdot \text{bit } 1) + (1 \cdot \text{bit } 0)$$

Cette notation permet donc de représenter, à l'aide de 1 octet seulement, les chiffres et les nombres allant de 0 à 255.

2. Nombres BCD

Dans le cas d'un nombre en notation BCD, les 4 bits de droite et les 4 bits de gauche de l'octet constituent 2 groupes distincts –que l'on appelle aussi des quartets (*nibbles* en anglais)– qui servent chacun à la représentation d'un nombre décimal codé binaire (BCD).

Cette technique de représentation a l'avantage d'une grande facilité de transfert. Le seul inconvénient, non négligeable pourtant, est que l'on ne peut pas, pour une addition ou pour une soustraction, utiliser ni l'addition binaire ni la soustraction binaire. Il faudra, pour ce faire, effectuer quelques corrections. Pour une addition à l'aide de l'indicateur **AC** on peut, par exemple, se servir de l'instruction **DA A** (voir plus loin).

3. Nombres signés

Il existe beaucoup d'applications dans lesquelles il est nécessaire d'utiliser des nombres négatifs. Afin de pouvoir représenter un nombre compris entre –128 à 127, on pourra procéder de la façon suivante :
Si le nombre **x** à représenter est positif, on lui donne la forme d'un octet non signé. Si pourtant le nombre **x** est négatif (**et** inférieur ou égal à 128) on lui donnera la forme du nombre **256 + x** NON SIGNÉ (compris alors entre 128 et 255) dans l'octet. Ce type de représentation se traduit donc par un positionnement du bit 7 si le nombre est négatif.

Pour un traitement d'un nombre négatif, il faudra d'abord en reconstituer la valeur pour la doter ensuite du signe négatif.
À l'image de la soustraction et de l'addition de nombres négatifs, il faudra, dans le cas d'une division et d'une multiplication, traiter les nombres négatifs d'une manière spéciale.

Voici, ci-dessous, quelques exemples montrant que l'interprétation de la même série de bits dans un octet peut avoir des résultats totalement différents :

Binaire	Hexadécimal	non signé	signé	BCD
00000000	00	0	0	0
00000001	01	1	1	1
00000010	02	2	2	2
00010001	11	17	17	11
00100101	25	37	37	25
11111111	FF	255	–1	indéfini
10000000	80	128	–128	80
01111111	7F	127	127	indéfini

4. Grands nombres

Il arrive souvent qu'un seul octet ne suffise pas pour la représentation de toutes les valeurs d'un nombre, si l'on veut obtenir la précision requise. Il faudra, dans ces conditions, faire appel à plusieurs octets, en réunissant soit 2 octets pour en faire une valeur à 16 bits par exemple, soit encore 4 octets pour obtenir une valeur à 32 bits.

S'il s'agit de mettre en mémoire de tels nombres, on peut «ranger» les octets concernés dans 2 colonnes. Lors de notre cours de programmation nous mettrons l'octet le moins significatif (avec l'adresse la plus faible des deux) en premier lieu.

À l'image de ce qui a été décrit dans ce paragraphe, on peut interpréter ces octets consécutifs comme un nombre, soit signé, soit non signé, soit encore comme un nombre BCD.

Si l'on utilise 2 octets, ce qui suffit amplement dans la plupart des applications faisant appel à un microcontrôleur, on couvre les plages suivantes :

non signé :	0 à 65 535
signé :	–32 768 à 32 767
BCD :	0 à 9 999

Le **tableau** suivant donne quelques exemples des différentes interprétations d'une paire d'octets :

Octet à l'adresse (hexadécimale) m + 1	m	Hexadécimal	Non signé	Signé	BCD
00	00	0000	0	0	0
12	34	1234	4 660	4 660	1234
0A	BC	0ABC	2 748	2 748	indéfini
80	00	8000	32 768	–2 768	8000
FF	FF	FFFF	65 535	–1	indéfini

Ce paragraphe, consacré à la notation et à l'interprétation de nombres en tout genre, nous amène automatiquement à la description de l'instruction arithmétique suivante :

Addition avec retenue

Lors de l'addition de nombres à 16 ou à 32 bits on se trouve confronté au problème suivant: la retenue, résultant de la position donnée d'un bit, doit être prise en compte lors de l'addition des octets suivants. Pour ce faire on utilise l'instruction :

```
ADDC   A,octet opérande   ;addition de l'opérande + CY (retenue)
                          ;au contenu de l'accumulateur A
```

Cette instruction a pour résultat que, non seulement l'opérande, mais également l'indicateur **CY**, est additionné au contenu de A. Si l'on veut par exemple additionner au nombre non signé à 16 bits, NOM1, présent à une certaine adresse en RAM, un nombre, NOM2 (lui aussi non signé), il faudra programmer :

```
MOV    A,NOM1       ;cherche l'octet de poids faible de NOM1
ADD    A,NOM2       ;addition de l'octet de poids faible de NOM2
MOV    NOM1,A       ;stocke le résultat
MOV    A,NOM1 + 1   ;octet suivant de NOM1
ADDC   A,NOM2 + 1   ;addition de l'octet suivant et de la retenue
                    ;précédente
MOV    NOM1 + 1,A   ;stocke le résultat
```

Correction BCD

Pour, après avoir effectué l'addition de 2 nombres BCD, retrouver un nombre BCD dans l'accumulateur, il faudra que l'addition soit suivie immédiatement de l'instruction suivante :

```
DA   A   ;correction BCD
```

Que fait-elle donc, cette instruction ?

Si la valeur des 4 bits de poids faible dans l'accumulateur dépasse 9 **ou** si l'indicateur auxiliaire **AC** est positionné, on a addition de la valeur 6 au contenu de l'accumulateur. De par cette opération on obtient déjà, dans la position BCD de poids le plus faible, le nombre correct. Au cas où cette addition de correction a pour résultat une retenue, qui se décale à travers tous les bits d'ordre plus élevé, l'indicateur **CY** est positionné.

On effectue ensuite un test pour voir si l'indicateur **CY** est positionné ou bien si la valeur des 4 bits de poids fort dépasse la valeur 9. Si tel est le cas, on additionne aussi la valeur 6 aux bits de poids fort. Si cela est le vrai, la valeur 6 est additionnée aux bits de poids supérieur. Ceci fournit donc le nombre BCD suivant. S'il se produit maintenant une retenue, l'indicateur de retenue **CY** est positionné une fois de plus. Cet indicateur signale donc, après l'instruction **DA**, que l'exécution de l'instruction **ADD**, voire **ADDC**, a pour résultat un nombre dépassant 99.

L'exemple suivant illustre nettement les effets d'une telle opération:

```
; Addition des nombres 99HEX et 11HEX
MOV A,#11H  ;transfert du nombre 11HEX
            ;vers l'accumulateur
            ;00010001B                    11
ADD A,#99H  ;accumulateur + nombre BCD 99
            ;+10011001B                   99
            ;pas de nombre BCD =
            ;10101010B                    AA
DA  A       ;résulte en
            ;00010000B + retenue
            ;correspondant au nombre BCD
            ;1 00010000 = 110
```

Après l'instruction **DA A**, l'indicateur **CY** indique à nouveau la

retenue qui correspond, en nombre BCD, à la valeur 100.

Soustraction

Pour effectuer une soustraction on ne dispose que d'une seule instruction, à savoir:

```
SUBB  A,opérande octet   ;soustraction de l'opérande et de
                         ;la retenue de l'accumulateur
```

Cette instruction se traduit par la soustraction de l'opérande indiqué du contenu de l'accumulateur. On a de plus, également soustraction du bit de retenue (Carry **CY**, **C** respectivement). Si l'on veut éviter cela, il faudra, au préalable, remettre à zéro l'indicateur **C** à l'aide de l'instruction **CLR C**.

Comparaison

Dans le cas du 8051, l'instruction de soustraction connaît une autre utilisation importante. On peut s'en servir pour effectuer une comparaison de 2 valeurs. Pour cela on se sert du fait suivant: en cas de soustraction de la valeur **y** d'un nombre **x**, on aura une retenue si **y** est supérieure à **x**. Pour vérifier si une valeur **y** est supérieure à un nombre **x**, on procédera à une soustraction de **y** de **x** (x – y) pour examiner ensuite l'indicateur **CY**. En langage de programmation cette opération prend la forme suivante:

```
MOV   A,x          ;cherche la valeur x dans la RAM interne
CLR   C            ;remise à zéro de la retenue
SUBB  A,y          ;soustraction: x – y
JC    SUPÉRIEUR    ;si y > x, saute à la routine SUPÉRIEUR
```

Les 2 nombres (**x** et **y**) en question sont interprétés comme nombres non signés.

La comparaison de nombres signés est beaucoup plus délicate à effectuer.

Multiplication et division

Le 8051 est doté d'un accumulateur auxiliaire, B. Il possède l'adresse SFR $0F0_{HEX}$ et il est utilisé en règle générale pour la multiplication et la division de nombres non signés à 8 bits. Pour effectuer ces opérations on dispose des instructions suivantes:

```
MUL   AB   ;multiplication A · B, A devient partie de poids faible
DIV   AB   ;division A := A / B; B := reste
```

La première de ces instructions multiplie les contenus des accumulateurs **A** et **B** et stocke le résultat (un nombre à 16 bits) dans **A** et **B**. L'octet de poids faible est stocké dans l'accumulateur **A**, l'octet de poids fort dans **B**. Si le résultat de la multiplication est supérieur à 255, l'indicateur **OV** est positionné.

L'instruction **DIV AB** se traduit par la division de **A** par **B**. Le nombre entier du résultat (le quotient, n'est-ce pas) de cette opération est stocké ensuite dans l'accumulateur **A**, le reste dans **B**. Sachant que l'on ne peut ni multiplier 2 nombres à 16 bits, ni diviser un nombre à 16 bits par un octet, les instructions de multiplication et de division ne sont pas aussi puissantes qu'on pourrait bien le souhaiter. Il existe pourtant de nouvelles versions du 8051 –tel que le 80537 de Siemens par exemple– qui connaissent des instructions beaucoup plus puissantes.

Pour effacer quelque peu les faiblesses arithmétiques du 8051, nous avons intégré quelques sous-programmes dans le moniteur **EMON51**, routines qui permettent de constituer un produit à 16 x 16 bits avec un résultat à 32 bits. L'examen du listing **EMON51.LST** présent sur la disquette du cours, vous permettra de mieux saisir la technique utilisée.

Le moniteur est également capable d'effectuer une division d'un nombre à 32 bits par un nombre à 16 bits. Dans ce sous-programme, le calcul se fait quasiment «par écrit» avec des valeurs à 16 et à 32 bits. Pour dévoiler tous les «secrets» de cette routine il faudra se référer au listing du moniteur sur la disquette du cours.

Mesure de capacité à l'aide de la sortie V24

Après toutes les explications théoriques concernant les possibilités arithmétiques du 8051, nous allons maintenant passer à la pratique et réaliser un capacimètre «rustique», fournissant la capacité mesurée à travers une sortie sérielle. Il va sans dire que la méthode de mesure mise en oeuvre ici ne convient pas uniquement à la mesure de capacités, mais également à celle de durées et de résistances.

La **figure 1** montre l'électronique requise pour cette opération, électronique présente sur la **carte d'extension pour SIMCAD**, réalisation décrite dans le n°162 d'Elektor (décembre 1991). La **figure 2** donne l'organigramme du programme **EXEMP8**, le logiciel nécessaire pour effectuer les mesures.

L'ensemble du processus est relativement simple. On commence par déclencher, à l'aide du logiciel, la bascule monostable IC8a. Ensuite, le programme «mesure» la durée de l'impulsion présente à la sortie de la bascule monostable. Cette durée est convertie en une valeur de capacité et transférée à travers le port sériel. La complexité de l'opération se cache, comme toujours, dans les détails:

1. Mesure de durée

Pour la mesure d'une durée, on utilise un nombre à 16 bits, **MTEMPS**, avec lequel on «compte» combien de temps dure l'impulsion de la bascule monostable. Au début de la mesure (étiquette ou *label* lp1) cette valeur est remise à 0 (lignes 20 et 21 de la **figure 3**). À l'aide du bit de port P1.2, une impulsion de démarrage (flanc positif) est alors appliquée à la bascule monostable. Afin que la bascule dispose d'un certain temps pour se remettre avant que ne lui arrive une nouvelle impulsion, la boucle d'attente (avec l'étiquette lpp) augmente la durée de cette impulsion de démarrage.

Ce n'est que maintenant que débute la boucle de mesure de durée (étiquette MLP). Le bit 3 de l'adresse de RAM $0C000_{HEX}$ sert à déterminer si la bascule est toujours déclenchée. Si cela est le cas («vrai»), la valeur dans **MTEMPS** est incrémentée (augmentée de 1). Si la bascule n'est plus déclenchée, la fin de la mesure est atteinte et le programme saute à l'étiquette **FINMES**.

L'augmentation de la valeur à 16 bits **MTEMPS**, trouve place dans les lignes 29 à 34 et illustre le

processus général des calculs de variables à bits, à 16, voire 32 bits. On calcule «par écrit», en commençant par l'octet de poids faible, tout en prenant en compte la retenue. Après l'augmentation de la valeur **MTEMPS**, la boucle redémarre.

2. Détection de dépassement
Lors du comptage dans la boucle **MLP**, il se peut que la valeur sorte hors de la plage admissible pour les nombres à 16 bits. Ceci se produit par exemple si l'on procède à la mesure d'un condensateur de capacité trop importante.

Pour vérifier cela, l'instruction dans la ligne 35 du logiciel fait appel à la retenue s'étant éventuellement produite. En cas de dépassement on a affichage d'un message d'erreur (lignes 36 et 37).

3. Conversion
Il va sans dire que la valeur affichée doit être rendue sous la forme d'une grandeur «standard», habituellement utilisée pour indiquer la valeur de la capacité d'un condensateur. Pour ce faire il faudra déterminer la durée d'impulsion de la bascule monostable en s'aidant du nombre de fois que le programme aura parcouru la boucle **MLP**. Sachant que le programme a besoin de 12 μs pour parcourir la boucle complète, la durée d'impulsion à la fin de la mesure est donc de:

$T = 12 \cdot \text{MTEMPS} \quad [\mu s]$.

On notera que ce résultat n'est pas parfaitement exact sachant qu'il faut déjà 6 μs au programme pour entrer dans la boucle de mesure.

Pour l'application prévue, cette petite erreur est pourtant négligeable.

D'après les indications fournies par les fabricants du 74HC123, la durée d'impulsion de cette bascule monostable est de:

$T = 0{,}45 \cdot R_{ext} \cdot C_{ext}$

(pour $C_{ext} > 10$ nF: valeurs en Ω, s et F).

Dans le circuit électronique, constituant la **carte d'extension pour SIMCAD**, R a la valeur 47 kΩ, ce qui nous donne une valeur de capacité de:

(en μF) = 0,5673 MTEMPS.

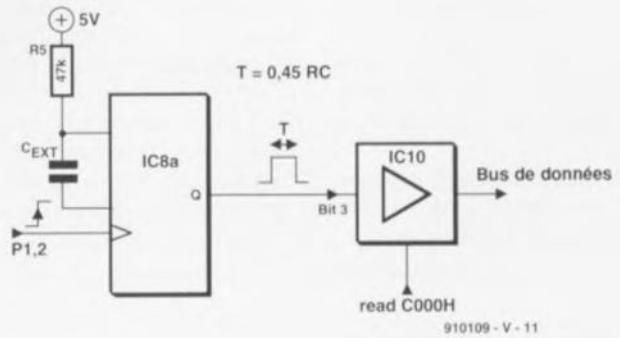


Figure 1. Schéma du capacimètre.

Comment faut-il faire alors pour calculer cette valeur ? Le 8051 ne connaît pas les possibilités d'une arithmétique à virgule flottante ! Pas de panique, il y longtemps que les programmeurs en Forth connaissent la solution à ce problème: les calculs font appel aux fractions.

Tout le monde sait que 0,5673 = 5673/10 000. Le numérateur et le dénominateur de cette fraction (p1 et q1 dans le programme) peuvent être représentés comme des nombres à 16 bits. Pour multiplier par 0,5673, nous allons d'abord multiplier **MTEMPS** par le numérateur (lignes 39 à 52) pour ensuite diviser le résultat, un nombre à 32 bits stocké dans les 4 octets commençant à **PROD32**, par le dénominateur (lignes 53 à 63) et voilà . . . le résultat est disponible sous forme d'une valeur à 16 bits (stockée en **Cmes**).

Pour visualiser cette valeur sous forme décimale, on fait appel, dans la ligne 66, à une routine d'**EMON51**.

Théorie et pratique

Hélas, dans la pratique, tout ce que nous avons dit concernant les formules mentionnées plus haut n'est vrai que partiellement.

Il faudra, pour commencer, noter que la résistance à 47 kΩ, utilisée sur la carte d'extension, se caractérise par une certaine imprécision. Beaucoup plus importantes (dans le sens négatif malheureusement !) sont pourtant les tolérances du circuit intégré 74HC123. Il est donc indispensable d'étalonner notre capacimètre.

Au lieu cependant, d'étalonner la résistance, nous allons procéder à un étalonnage par le, et du, logiciel. Pour ce faire nous accordons aux variables p1 et q1 la valeur 1 000 et démarrons le programme. La valeur affichée dans ces conditions ne correspondra donc pas à la valeur du condensateur, mais à la valeur exacte de la variable **MTEMPS**. Après avoir connecté au circuit un condensateur de 1 μF (ayant la tolérance la plus faible possible), il faudra mémoriser la valeur affichée (telle que 1 540 par exemple) pour l'utiliser comme variable q1 dans le programme définitif. La valeur de p1 sera de p1 = 1 000. Voilà, l'étalonnage est terminé !

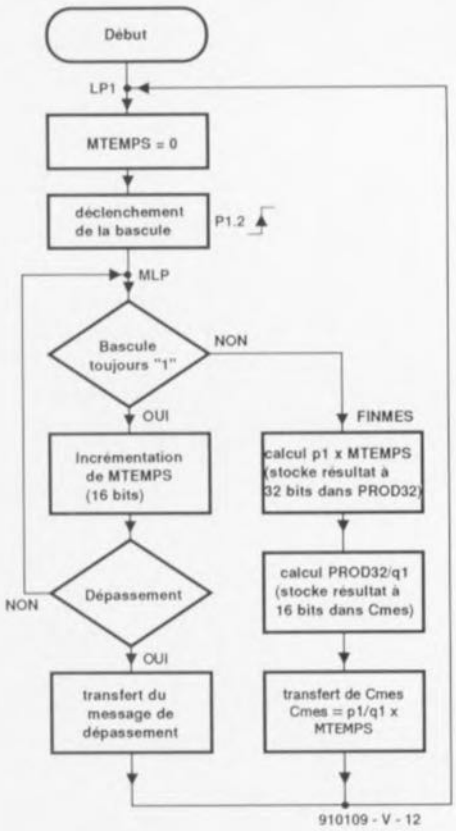


Figure 2. Organigramme du logiciel pour mesurer des capacités.

Si donc on reconnecte un condensateur de 1 µF, la valeur de **MTEMPS** sera à nouveau 1 540. Multipliée par p1 = 1 000 et divisée ensuite par q1 = 1 540, la valeur affichée sera exactement 1 000 nF.

Comme la durée pendant laquelle la bascule monostable est déclenchée, est –approximativement– proportionnelle à la capacité du condensateur connecté, notre capacimètre est étalonné correctement.

Ce simple processus de mesure illustre déjà nettement, qu'il faudra disposer de toute une série de connaissances détaillées dès qu'il s'agit de convertir des résultats de mesure. Tous ceux d'entre nos lecteurs qui désirent se «faire les dents» sur cette matière complexe peuvent étudier et examiner les routines arithmétiques du moniteur pour les tester ensuite dans l'un ou l'autre petit programme expérimental de leur crû.

```
****** LISTING of EASM51 (EXEMP8) ******
LINE LOC  OBJ          T        SOURCE
   1 0000
   2 0000                ; ****** FICHIER EXEMP8.A51 ************************************
   3 0000                ;
   4 0000                P1      EQU 090H          ; différentes adresses SFR
   5 0000                ACC     EQU 0E0H
   6 0000                DPL     EQU 082H
   7 0000                DPH     EQU 083H
   8 0000                ;                         ; constantes d'étalonnage
   9 0000                p1      EQU    1000       ; numérateur
  10 0000                q1      EQU    1540       ; dénominateur du facteur d'étalonnage
  11 0000                ;
  12 0000                MONTOP  EQU    050H       ; utilise assembleur pour gestion de la RAM
  13 0000                        ORG    MONTOP     ; au-delà de la RAM du MONITEUR
  14 0050                MTEMPS  DS     2          ; durée mesurée 16 bits
  15 0052                p       DS     2          ; p=p1 : multiplicateur ( valeur à 16 bits )
  16 0054                q       DS     2          ; q=q1 : diviseur ( valeur à 16 bits )
  17 0056                PROD32  DS     4          ; produit MTEMPS*p ( valeur à 32 bits )
  18 005A                Cmes    DS     2          ; résultat en nF ( valeur à 16 bits )
  19 005C                ;
  20 005C                        ORG    4100H          ; adresse du début du programme
  21 4100 75 50 00 [2] lp1       MOV    MTEMPS+0,#0    ; remise à zéro de la valeur à 16 bits MTEMPS
  22 4103 75 51 00 [2]           MOV    MTEMPS+1,#0
  23 4106 C2 92    [1]           CLR    P1.2           ; déclenchement:=0
  24 4108 E4       [1]           CLR    A
  25 4109 D5 E0 FD [2] lpp       DJNZ   ACC,lpp        ; boucle d'attente de l'impulsion de déclenchement
  26 410C D2 92    [1]           SETB   P1.2           ; déclenchement de la bascule monostable
  27 410E 90 C0 00 [2]           MOV    DPTR,#0C000H   ; adresse de la sortie de la bascule monostable
  28 4111 E0       [2] MLP       MOVX   A,@DPTR        ; lecture de la sortie de la bascule monostable
  29 4112 30 E3 15 [2]           JNB    ACC.3,FINMES   ; si bit 3 = 0, saute à FIN
  30 4115 E5 50    [1]           MOV    A,MTEMPS       ; sinon: augmentation de la valeur à 16 bits
  31 4117 24 01    [1]           ADD    A,#1           ; cela veut dire: incrémentation
  32 4119 F5 50    [1]           MOV    MTEMPS,A       ; stockage du bit de poids faible (LSB)
  33 411B E5 51    [1]           MOV    A,MTEMPS+1     ; cherche bit de poids fort (MSB)
  34 411D 34 00    [1]           ADDC   A,#0           ; addition de 0 + retenue
  35 411F F5 51    [1]           MOV    MTEMPS+1,A     ; stockage du MSB
  36 4121 50 EE    [2]           JNC    MLP            ; retenue - dépassement, sinon: continuer
  37 4123 90 41 74 [2]           MOV    DPTR,#OVTXT    ; transfert du message de dépassement
  38 4126 31 8B    [2]           ACALL  STXT
  39 4128 80 D6    [2]           SJMP   lp1
  40 412A 90 03 E8 [2] FINMES    MOV    DPTR,#p1       ; fin de la mesure de durée
  41 412D 85 82 52 [2]           MOV    p+0,DPL        ; p:=p1 ( valeur à 16 bits )
  42 4130 85 83 53 [2]           MOV    p+1,DPH
  43 4133 78 52    [1]           MOV    R0,#p
  44 4135 79 50    [1]           MOV    R1,#MTEMPS
  45 4137 75 30 52 [2]           MOV    COMMAND,#ccMUL ; calcul de MTEMPS*p
  46 413A 12 02 00 [2]           LCALL  MON
  47 413D 86 56    [2]           MOV    PROD32+0,@R0   ; stockage dans PROD32 (valeur à 32 bits )
  48 413F 08       [1]           INC    R0
  49 4140 86 57    [2]           MOV    PROD32+1,@R0
  50 4142 08       [1]           INC    R0
  51 4143 86 58    [2]           MOV    PROD32+2,@R0
  52 4145 08       [1]           INC    R0
  53 4146 86 59    [2]           MOV    PROD32+3,@R0
  54 4148 90 06 04 [2]           MOV    DPTR,#q1       ; q:=q1
  55 414B 85 82 54 [2]           MOV    q+0,DPL
  56 414E 85 83 55 [2]           MOV    q+1,DPH
  57 4151 78 56    [1]           MOV    R0,#PROD32
  58 4153 79 54    [1]           MOV    R1,#q
  59 4155 75 30 53 [2]           MOV    COMMAND,#ccDIV ; calcul de (MTEMPS*p)/q  ( valeur à 16 bits )
  60 4158 12 02 00 [2]           LCALL  MON
  61 415B 86 5A    [2]           MOV    Cmes+0,@R0     ; stockage dans Cmes
  62 415D 08       [1]           INC    R0
  63 415E 86 5B    [2]           MOV    Cmes+1,@R0
  64 4160 90 41 82 [2]           MOV    DPTR,#TXT1     ; transfert du premier texte
  65 4163 31 8B    [2]           ACALL  STXT
  66 4165 78 5A    [1]           MOV    R0,#Cmes
  67 4167 75 30 05 [2]           MOV    COMMAND,#ccdRO16 ; transfert de Cmes sous forme décimale
  68 416A 12 02 00 [2]           LCALL  MON
  69 416D 90 41 85 [2]           MOV    DPTR,#TXT2     ;transfert du second texte
  70 4170 31 8B    [2]           ACALL  STXT
  71 4172 21 00    [2]           AJMP   lp1
  72 4174                ;
  73 4174 0D 0A 44       OVTXT   DB     13,10,'Dépassement',0
          82 70 61
          73 73 65
          6D 65 6E
          74 00
  74 4182 43 3D 00       TXT1    DB     'C=',0
  75 4185 20 6E 46       TXT2    DB     ' nF',13,10,0
          0D 0A 00
  76 418B                ;
  77 418B                ccSTXT  EQU  002H  ; instructions et adresses du MONITEUR
  78 418B                ccdRO16 EQU  005H
  79 418B                ccMUL   EQU  052H
  80 418B                ccDIV   EQU  053H
  81 418B                COMMAND EQU  030H
  82 418B                MON     EQU 0200H
  83 418B                ;
  84 418B 75 30 02 [2] STXT      MOV COMMAND,#ccSTXT
  85 418E 02 02 00 [2]           LJMP MON
  86 4191                        END
********** SYMBOLTABLE (26 symbols) **********
       P1 :0090        ACC :00E0        DPL :0082        DPH :0083
       p1 :03E8         q1 :0604     MONTOP :0050     MTEMPS :0050
        p :0052          q :0054     PROD32 :0056       Cmes :005A
      lp1 :4100        lpp :4109        MLP :4111     FINMES :412A
    OVTXT :4174       TXT1 :4182       TXT2 :4185     ccSTXT :0002
  ccdRO16 :0005      ccMUL :0052      ccDIV :0053    COMMAND :0030
      MON :0200       STXT :418B
```

Figure 3. Listing du programme du capacimètre.

Décalage et rotation

On dispose, pour effectuer une rotation vers la droite ou vers la gauche des bits compris dans un octet, des instructions suivantes:

RL	A	;rotation de l'accumulateur vers ;la gauche
RLC	A	;rotation de l'accumulateur vers ;la gauche en tenant compte du bit C
RR	A	;rotation de l'accumulateur vers ;la droite
RRC	A	;rotation de l'accumulateur vers ;la droite en tenant compte du bit C

Dans la **figure 4** sont illustrés les effets des instructions pour une rotation vers la gauche.

Si l'on remet à zéro l'indicateur **C** avant une instruction **RLC** ou **RRC** il devient également possible de décaler l'accumulateur (en le remplissant avec des 0).

Les instructions mentionnées ci-dessus permettent de simuler des registres à décalage ou encore de décaler des nombres destinés à des opérations arithmétiques (voir la routine **DIV** dans **EMON51**).

Générateur de bruit

Dans l'ouvrage* cité en fin de cet article, il est décrit un générateur de bruit dont la «pièce de résistance» est représentée en **figure 5**: un registre à décalage à réinjection.

Il s'agit maintenant d'émuler ce circuit sur la **carte d'extension pour SIMCAD** en programmant un générateur de bruit numérique. Le haut-parleur, qui fait partie de la carte d'extension est adressable par l'intermédiaire du bit P1.1, fera ici office de dispositif de «sonorisation».

La figure 5 illustre également comment il faudra transformer l'électronique en logiciel. Le registre à décalage est stocké dans 4 octets consécutifs de la RAM interne. Le bit ayant la position la plus à «droite» se trouve dans l'octet avec l'adresse la plus

faible. Cette méthode de mise en mémoire correspond exactement à la représentation d'un nombre à 32 bits. Il s'agit donc de faire effectuer à ce nombre un décalage vers la gauche. Lors de ce processus, la porte EXNOR (NON-OU-EXCLUSIF) détermine quel bit prendra la place libre à droite. La position de poids le plus fort, celle du bit 31, est reliée à l'haut-parleur.

La **figure 6** montre le listing du logiciel qui consiste en fait en une permutation directe des synoptiques de la figure 5. La programmation de la porte EXNOR dans les lignes 15 à 20 est la seule astuce à laquelle on fasse appel dans ce logiciel. Pour créer, à l'aide des bits SREG3.6 et SREG2.1 le nouveau bit requis, on utilise ici le bit 1 de l'accumulateur. Le registre à décalage à réinjection semble générer un modèle de bits «aléatoire» (ordre de chiffres pseudo-aléatoire), qui ne se répète qu'après une période assez longue. Calculons-la avec une meilleure précision:

Il faut, pour parcourir une fois la boucle, 23 µs et le registre à décalage prend quelque 2 000 000 000 états différents avant de «se répéter». On génère donc un «signal aléatoire» ayant une fréquence de quelque 43 kHz qui ne se répète qu'après 13 heures environ.

Si l'on envisage une utilisation comme générateur de bruit pour effectuer des mesures, il est recommandé de lui adjoindre un filtre passe-bas pour définir le bon spectre. Même sans filtre passe-bas, le petit haut-parleur de notre prototype faisait entendre un bruit relativement puissant jusqu'à quelque 20 kHz, avec un spectre de bruit pratiquement blanc.

«Devoirs»

Tous ceux d'entre nos lecteurs qui désirent s'entraîner à utiliser toutes les instructions arithmétiques du 8051, peuvent attaquer l'un des problèmes suivants:

- accroissement de la plage du capacimètre, de manière à pouvoir mesurer **et** des capacités faibles **et** des capacités plus importantes,
- programmation d'un changement automatique de plage de mesure (calibre),
- modification du logiciel du capacimètre pour pouvoir vérifier si le condensateur connecté a bien la valeur indiquée (avec une certaine tolérance bien entendu) et
- modification du programme du générateur de bruit pour le transformer en générateur de rythmes.

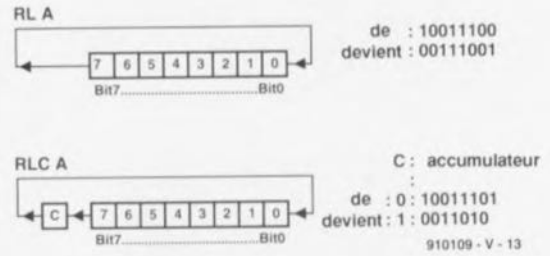


Figure 4. Fonctionnement des instructions de rotation.

Le mois prochain

Nous voici arrivés, trop tôt ?, à la fin de ce 4e article. Nous vous avons proposé pratiquement toutes les instructions du 8051 et

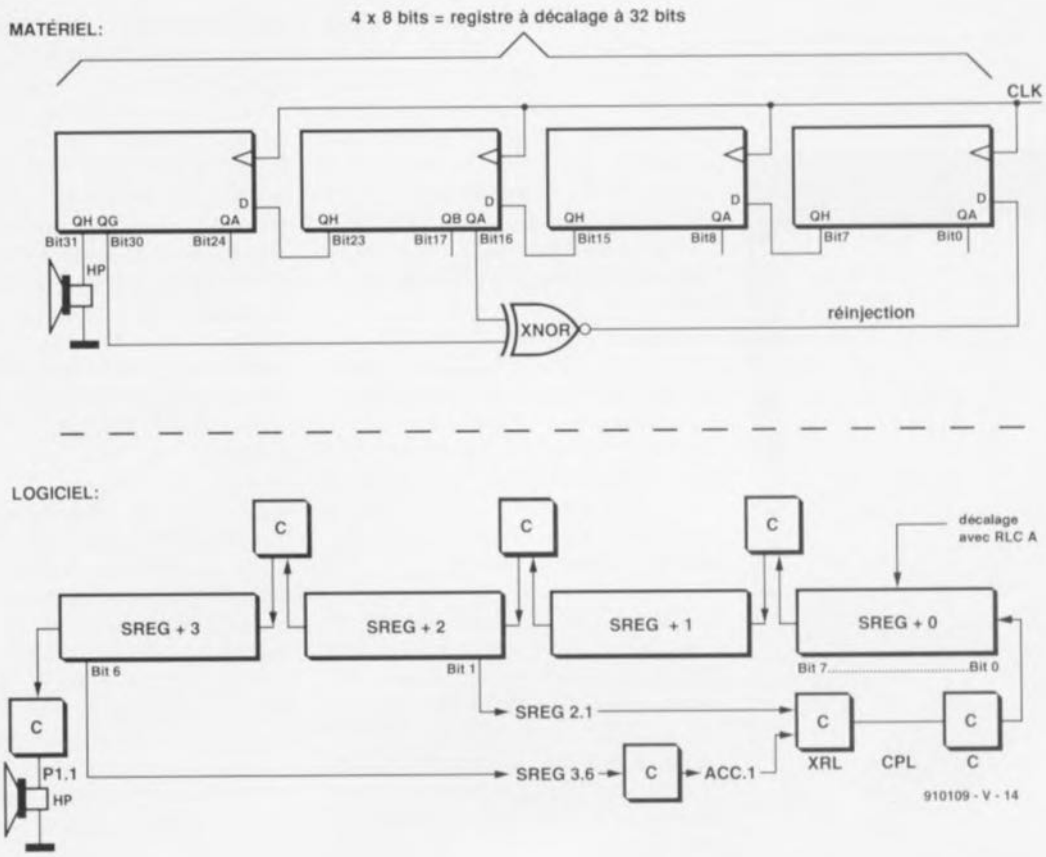


Figure 5. Registres à décalage, utilisés ici pour la génération de bruit. La partie supérieure de la figure montre le synoptique de la version électronique, celle du bas la version en logiciel.

```
; ****** FICHIER EXEMP9.A51 ************************************
;
P1      EQU 090H            ; adresses SFR habituelles
ACC     EQU 0E0H
;
MONTOP  EQU   050H          ; utilise assembleur pour gérer RAM
        ORG   MONTOP        ; au-delà de la RAM du MONITEUR
SREG    DS    4             ; registre à décalage à 32 bits
;
        ORG   4100H         ; adresse du début du programme
        MOV   SREG+0,#0     ; effacement du registre à décalage (à 32 bits)
        MOV   SREG+1,#0
        MOV   SREG+2,#0
        MOV   SREG+3,#0
NEUF    MOV   A,SREG+3      ; effectuer fonction NON-OU-EXCLUSIF (EXNOR)
        MOV   C,ACC.6       ; déplacement du bit 6 de l'octet 3 de SREG
        MOV   ACC.1,C       ; vers bit 1 de l'accumulateur A
        XRL   A,SREG+2      ; avec position 1 de SREG+2 XOR
        MOV   C,ACC.1       ; si XOR du bit 17 et 30
        CPL   C             ; C est fonction NOT-XOR des bits 17 et 30
;
DECALE  MOV   A,SREG+0      ; Registre à décalage à 32 bits
        RLC   A             ; décaler octet droit
        MOV   SREG+0,A      ; et le mémoriser
        MOV   A,SREG+1      ; chercher l'octet suivant
        RLC   A             ; le bit est stocké momentanément en C
        MOV   SREG+1,A
        MOV   A,SREG+2      ; etc.
        RLC   A
        MOV   SREG+2,A
        MOV   A,SREG+3
        RLC   A
        MOV   SREG+3,A      ; C contient alors nouvelle valeur
        MOV   P1.1,C        ; transférer cette valeur vers haut-parleur
        SJMP  NEUF          ; reprendre au début
        END
```

Figure 6. Listing du générateur de bruit numérique.

vu leur fonctionnement. On pourrait donc presque dire, soyons un rien fanfarons, que, du côté du logiciel, le 8051 n'a plus de secrets pour vous et nous. C'est pour cette raison que les prochains épisodes du cours seront consacrés au matériel: le temporisateur du 8051, la connexion d'un affichage à cristaux liquides, le port sériel du 8051 et la conversion N/A, voire A/N.

** Horowitz,P.,Hill,W.: The Art of Electronics, Cambridge University Press, 1989*

Minitel
3615
+
Elektor

The Realizer

Actum Solutions

Logiciel CASE pour «dessiner des logiciels»

NdlR: seuls pourront faire l'objet d'un compte-rendu rédactionnel les logiciels envoyés à la rédaction tels qu'ils le seraient à un éventuel acheteur.

Il n'existe pratiquement plus aujourd'hui le moindre appareil pouvant se targuer d'une certaine sophistication qui ne soit pas doté de l'un ou l'autre type de microcontrôleur. Il semble donc logique que certaines sociétés créant des logiciels se soient mises à la recherche de procédés permettant la production rapide et fiable de logiciel-système pour les microcontrôleurs. Dans ce domaine, les Néerlandais n'ont pas à rougir face à leurs concurrents européens, voire américains.
The Realizer, un produit des Pays-Bas, est un logiciel CASE (***C**omputer **A**ided **S**oftware **E**ngineering* = Développement et mise au point de logiciel assistés par ordinateur) qui a pour but de simplifier très sensiblement la vie du concepteur de logiciel-système: avec The Realizer, la conception d'un programme devient alors une affaire de dessin.

L'époque où la seule façon de programmer un microcontrôleur était d'intervenir au niveau des bits et des octets en code-assembleur est loin derrière nous. Cette évolution est sans doute due à un certain nombre de facteurs dont le plus important est sans doute qu'il est plus difficile d'écrire en assembleur qu'en un langage plus évolué. Autre inconvénient accessoire de l'assembleur est que le logiciel n'est développé que pour un type de microcontrôleur donné. Ceci sous-entend que le passage à une famille de processeurs plus puissante rend dépassées toutes les applications écrites précédemment. Il n'est pas surprenant dans ces conditions que les fabricants de ce type de composant se soient mis en quête d'un système de développement de remplacement.

Il n'a pas fallu longtemps avant de voir apparaître des compilateurs pour les langages de haut niveau. Tout au début on avait fait appel à des langages développés spécifiquement pour ces processeurs, tels que, par exemple le PL/M d'Intel, et à un stade ultérieur, un langage universel tel que C.
Dès qu'apparaît sur le marché un compilateur C pour une nouvelle génération de microcontrôleurs, il est possible, en principe, d'utiliser les logiciels existants avec cette nouvelle famille. Lors de ce développement la tendance générale observée a été que l'utilisateur commence par opter pour une famille de processeurs pour ensuite acquérir les outils de développement de logiciel adéquats.

Dans le cas de The Realizer c'est le processus inverse que l'on suit. Avec ce logiciel, l'utilisateur dispose d'un environnement de développement puissant lui permettant d'écrire du logiciel-système sans qu'il ne soit nécessaire de choisir pour un type de processeur donné. Ce n'est qu'en phase terminale, lorsqu'il faut créer le code spécifique au processeur, qu'il faudra que l'utilisateur manifeste sa préférence et choisisse une famille de microcontrôleur bien définie.

Du synoptique au langage machine

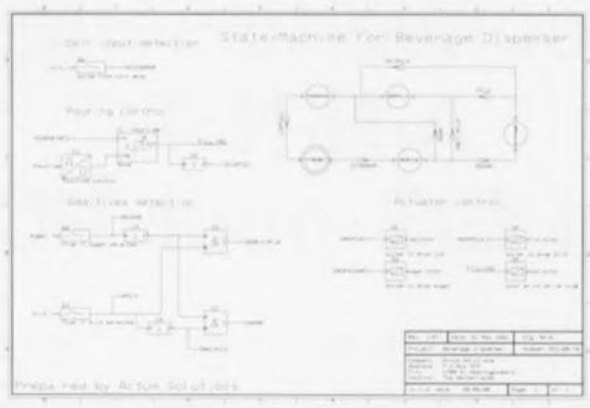
Un ingénieur chargé de développer un logiciel commence en règle générale

(enfin c'est ce qu'il devrait faire) par réfléchir à ce que doit faire son programme. L'une des techniques les plus efficaces consiste à dessiner un synoptique comportant tous les points d'entrée et de sortie (import/export). Et c'est à partir de là que The Realizer se trouve dans son élément et qu'il peut remplir la fonction pour laquelle il a été conçu.
On dessine, à l'aide du programme de dessin de schéma (UltiCap), le synoptique qui rend le concept fonctionnel de l'application requise. On peut utiliser pour ce dessin les éléments de la bibliothèque fournie qui accompagne ce logiciel :
* entrées/sorties (numériques et analogiques)
* éléments divers de temporisation et compteurs de tout genre
* convertisseurs et comparateurs
* mémoires
* opérations logiques et arithmétiques.

Le logiciel vérifie que l'utilisateur a bien interconnecté les différents composants en respect des protocoles prévus. Les dessins ainsi réalisés sont parfaitement indépendants du matériel, peuvent recevoir des dénominations symboliques et être accompagnés de commentaires.
Une fois le dessin terminé, on lance l'analyseur de The Realizer à son assaut. Ce module du programme vérifie l'absence de «court-circuit» dans le schéma et la connexion de toutes les entrées.
Il vérifie en outre que les blocs interconnectés doivent bien l'être en réalité. Il ne saurait, par exemple, être question de relier une entrée binaire à une entrée intégrale (faisant appel à des chiffres ou nombres entiers).
En l'absence d'erreur, le logiciel génère un code provisoire. Parallèlement, on a génération d'une base de données dans laquelle sont documentées toutes les étapes du processus. On voit ainsi très précisément quels documents (y compris leur numéro de version) il faut pour développer l'application.

Simulation intégrée

Plus tôt on découvre une erreur dans un projet, plus il est aisé et donc bon marché de l'éliminer. Ceci explique que l'on consacre énormément de temps aux essais et simulations d'un système quel qu'il soit. The Realizer présente, en ce qui concerne cette simulation, un avantage majeur par rapport aux systèmes de développement classiques. Le système est en effet capable de simuler l'application d'une seule pièce et non pas uniquement de faire tourner le programme du microcontrôleur. La seule technique d'élimination des derniers «bugs» que connaisse bon nombre de concepteurs de logiciel est de



travailler au niveau de l'octet, voire du bit.

Le simulateur «View Picture» utilise la banque de données générée auparavant et propose à l'utilisateur un certain nombre d'instruments de mesure. On trouve parmi les auxiliaires spécifiques de cette partie du logiciel les outils suivants:
* affichage d'entiers (numérique)
* barregraphe d'entiers (histogramme)
* scope d'entiers (enregistrement dans le temps des variables entières)
* entier ajustable (potentiomètre)
* bit ajustable (commutateur)
* affichage de bit (petite ampoule)
* scope de bit (enregistrement dans le temps de l'information au niveau du bit).

Il est possible à l'aide de ces outils de tester, depuis View Picture, qui comme nous le disions est une partie de The Realizer, de tester dans le plus petit détail l'ensemble du processus sans qu'il ne doit nécessaire de disposer de la moindre extension matérielle.
Il est éventuellement possible de sauvegarder l'ensemble de la disposition de test pour, lors d'une phase ultérieure, utiliser la disposition de mesure électronique.

Le couplage au matériel

Bien que le projet soit déjà fort avancé, nous n'avons pas encore mentionné son aspect «matériel». Nous allons maintenant nous y intéresser. Tout ce que nous avons fait jusqu'à présent a été simulé dans un environnement symbolique. Voici arrivé le moment crucial où les différentes entrées/sorties, compteurs et autres registres sont accouplés à matériel réel. Il faut maintenant, dans la base de données indiquant tous les matériels connus par le logiciel, choisir l'un des nombreux processeurs de Motorola, Intel, Texas Instruments ou Hitachi. On peut également faire appel à des PLC ou à des langages de programmation de haut niveau tels que C ou Pascal comme «*device*». L'utilisateur peut au cours de cette phase quels broches et/ou registres du matériel seront utilisés pour remplir une fonction donnée. Le fichier de configuration du matériel créé à partir de là dépend du type de processeur utilisé et devra donc être modifié et adapté si l'utilisateur choisit à un moment ou à un autre de changer de matériel (processeur principalement). Sachant que nous en étions pratiquement arrivés à la dernière étape du développement du système, la majeure partie du travail est déjà effectuée de sorte que le changement de matériel ne se traduit que par une faible perte de l'énergie investie dans le projet.
Il suffit d'une action sur une touche pour faire générer à The Realizer le code-source agrémenté d'une quantité suffisante de commentaires. Ce code-source, en C, Pascal ou assembleur, peut ensuite être converti à l'aide du compilateur ou de l'assembleur convenable en code machine. L'assembleur ou le compilateur ne font pas partie de The Realizer: il faudra donc se les fournir séparément.

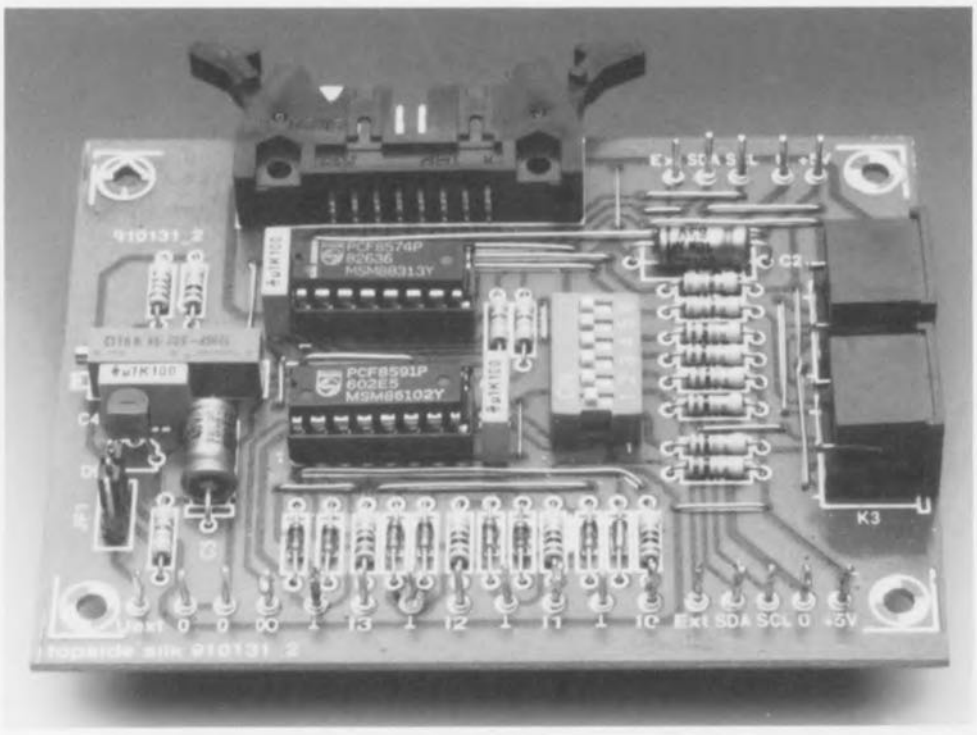
Un code fourni en Pascal ou en C pourra être utilisé de 2 manières différentes: il peut être, à l'aide d'un compilateur, converti en code machine pour un microcontrôleur, mais peut également être utilisé comme routine pour un système plus important. Si l'on n'a pas fait d'erreur dans les définitions en début de programme, le code produit est bon à 100%, caractéristique d'une importance capitale vue la durée de vie de plus en plus courte de la majorité des appareils commercialisés.

Note: Actum Solutions n'a pas encore de distributeur officiel en France, mais est en pourparlers très avancés avec un grand distributeur de composants français. Lorsque le contrat sera signé, nous vous en ferons part.

Ce logiciel a été mis gracieusement à notre disposition par la société:
Actum Solutions
P.O. Box 373
NL 1700 AJ Heerhugowaard
(Pays-Bas)
tél.: (+31).2207.45553
fax.: (+31).2207.12457

convertisseur A/N-N/A et E/S pour I²C

mesurer, commuter, réguler avec un PC sous I²C



Dans le numéro du mois de janvier nous vous avons proposé, sous le nom d'interface I²C encartable pour PC, une interface compacte permettant à tout possesseur de PC de communiquer avec des circuits intégrés dotés eux-mêmes d'une interface I²C.
Le présent article décrit une carte d'extension pour ce système, platine sur laquelle se trouvent un port d'Entrées/Sorties et un convertisseur double A/N et N/A.
C'est en outre une occasion rêvée pour vous présenter le logiciel-système écrit pour la carte I²C.

Dans l'article du mois de janvier nous avions affirmé, sans sourciller le moins du monde, que l'utilisation du bus I²C serait, de par la description de l'interface I²C effectuée dans le dit numéro, mise à la portée de tout utilisateur de PC.

L'interface va bien évidemment de pair avec un minimum de logiciel qui fera l'objet d'une description sommaire dans ce même article.
Ce logiciel de commande de périphérique (*device driver*, comme on l'appelle en anglais) est en fait une extension du système d'exploitation du PC et comporte toutes les routines permettant, en conformité avec le protocole de Philips, d'envoyer des données vers les circuits intégrés connectés au bus I²C. La prise en compte (lecture) de données se fait automatiquement en respect de ce protocole. Le PC connaît différents «*device drivers*»; on y fait par exemple appel pour la gestion d'un pseudo-lecteur en RAM (*RAM disk*), la commande d'une imprimante; la gestion de l'écran et du clavier se font elles aussi à l'aide d'un tel «*device driver*».

Il existe, en principe, 2 types de «*device driver*»: les drivers de bloc et les drivers de caractères. Les premiers sont utilisés pour les supports de masse tels que les disques durs, les drivers de caractères l'étant pour l'écran, le clavier et, dans le cas présent, aussi pour le bus I²C.
Comme nous le disions plus haut, un «*device driver*» est à considérer comme une extension du système d'exploitation (SED, plus communément appelé DOS). Il est de ce fait toujours «invoqué» par l'intermédiaire du DOS. Selon les spécifications du DOS, un «*device driver*» comporte 17 routines, l'utilisateur ayant cependant la liberté de ne pas s'encombrer de routines inutiles qu'il a donc le loisir de supprimer. C'est d'ailleurs ce que nous avons fait ici.

Voici les sous-programmes d'un «*device driver*» selon DOS:

0* Initialisation du driver
1 Test du médium
2 Création du bloc paramètres du BIOS
3 Lecture de la commande d'E/S
4* Lecture
5 Lecture non-destructrice
6 État d'entrée
7 Effacement des tampons d'entrée
8* Écriture
9* Écriture & vérification
10 État de sortie
11 Effacement des tampons de sortie
12* Écriture de commande d'E/S
13* Ouverture du «*device*»
14* Fermeture du «*device*»
15 Medium amovible
16* Sortie jusqu'à état «*Busy*»

Les routines identifiées par un «*» sont utilisées dans le driver I²C. La disquette proposée en parallèle à cet article sous l'appellation **ESS1674** –disponible auprès des adresses habituelles– comporte le fichier-source de ce «*device driver*» (écrit bien entendu en langage machine pur). Sachant qu'une description détaillée de la structure de ce driver nous amènerait bien trop loin, nous recommandons à tous les amateurs de ce genre de matériau, d'analyser ce fichier-source. Une visite dans une librairie spécialisée ou l'examen du catalogue d'une maison d'édition bien achalandée en ouvrages consacrés à la micro-informatique, devrait vous permettre de trouver les informations nécessaires et suffisantes à la compréhension du mode de fonctionnement d'un «*device driver*» ainsi qu'à la programmation en

langage machine en environnement DOS.

L'installation

La disquette **ESS1674** comporte le fichier assemblé **I2CDRIV.SYS** correspondant au fichier-source évoqué plus haut. On pourra copier ce fichier dans le répertoire racine (ROOT) du PC. On pourra ensuite modifier en conséquence le fichier de configuration **CONFIG.SYS**. Il suffit pour ce faire de lui ajouter la ligne suivante :

DEVICE = I2CDRIV.SYS

On pourra faire suivre le nom de fichier **I2CDRIV.SYS** de 2 paramètres, à savoir **B :xxxx** et **C :y**, le xxxx représentant l'adresse de base, c'est-à-dire l'adresse d'Entrées/Sorties la plus basse de la carte.

Comme l'expliquait l'article consacré à la description de l'**interface I²C encartable pour PC**, on peut définir l'adresse sur la carte à l'aide d'un septuple interrupteur DIL dont les contacts permettent, par leur ouverture ou leur fermeture, de définir une adresse comprise entre 300_{HEX} et $3FE_{HEX}$; le **y** sert à entrer un code pour la fréquence d'horloge SCL du bus I²C. Les valeurs de la fréquence d'horloge diffèrent quelque peu des valeurs données dans les recueils de caractéristiques (*data books*), pour la simple et bonne raison que le contrôleur I²C travaille avec un signal d'horloge de 7,16 MHz et non pas de 8 MHz. On pourra attribuer à **y** l'une des valeurs suivantes :

0 : f_{SCL} = 81 kHz
1 : f_{SCL} = 40 kHz
2 : f_{SCL} = 9,8 kHz
3 : f_{SCL} = 1,3 kHz.

En l'absence de définition des valeurs B et C, le logiciel «suppose» que l'adresse de base est 300_{HEX} et que la fréquence d'horloge est de 9,8 kHz.

Après toute réinitialisation du système, le logiciel du système d'exploitation lit le (nouveau) fichier de configuration **CONFIG.SYS** et le PC connaît toutes les routines capables de s'entendre avec la carte I²C. Le vrai travail peut commencer.

Un certain nombre d'exemples de programmes, proposés eux aussi sur la disquette **ESS1674**, permettent de comprendre comment s'y prendre pour intégrer les routines du «*driver*» dans un programme écrit dans l'un des langages de programmation courants tels que le C, le Pascal ou l'assembleur. C'est sans aucun doute la méthode la plus rapide pour saisir la plupart des subtilités du sujet.

Caractéristiques techniques de la carte d'E/S, A/N et N/A
- Travaille par l'intermédiaire du bus I²C,
- 4 entrées analogiques (256 pas),
- 1 sortie analogique (256 pas),
- 8 ports d'E/S (bidirectionnels),
- Possibilité de connexion d'un maximum de 8 platines sur un bus I²C,
- Ajustage possible entre 2,8 et 4,1 V de la tension de référence,
- Complet avec logiciel sous DOS,
- Mise à disposition du code source pour sa conversion éventuelle vers C, Pascal ou langage machine.

Le matériel

La **figure 1** donne le schéma de l'électronique présente sur la carte de conversion A/N et N/A à port d'E/S commandée par l'intermédiaire du PC.

Les composants les plus importants de cette carte sont 2 circuits intégrés de Philips, le **PCF8574** (circuit intégré d'E/S) et le **PCF8591** (convertisseur A/N–N/A). Quelques rares composants externes additionnels complètent ce schéma. Est-il preuve plus éclatante de la capacité du bus I²C de permettre la conception et la réalisation d'interfaces extrêmement compactes ?

Le port d'E/S à 8 bits est d'une simplicité renversante. Via le connecteur K1, ses 8 lignes peuvent être reliées à des grandeurs numériques du monde extérieur.

3 lignes d'adresses, A0, A1 et A2, sont reliées à la ligne positive de l'alimentation via une résistance de forçage au niveau logique haut (*pull-up*). 3 des contacts du sextuple interrupteur DIL S1 peuvent servir à définir la partie variable de l'adresse du circuit intégré d'E/S. Cet interrupteur DIL permet la connexion simultanée au bus I²C d'un maximum de 8 exemplaires du PCF8574 au bus I²C.

Comme c'est le cas pour tous les circuits intégrés devant être reliés, leurs adresses sont, en partie du

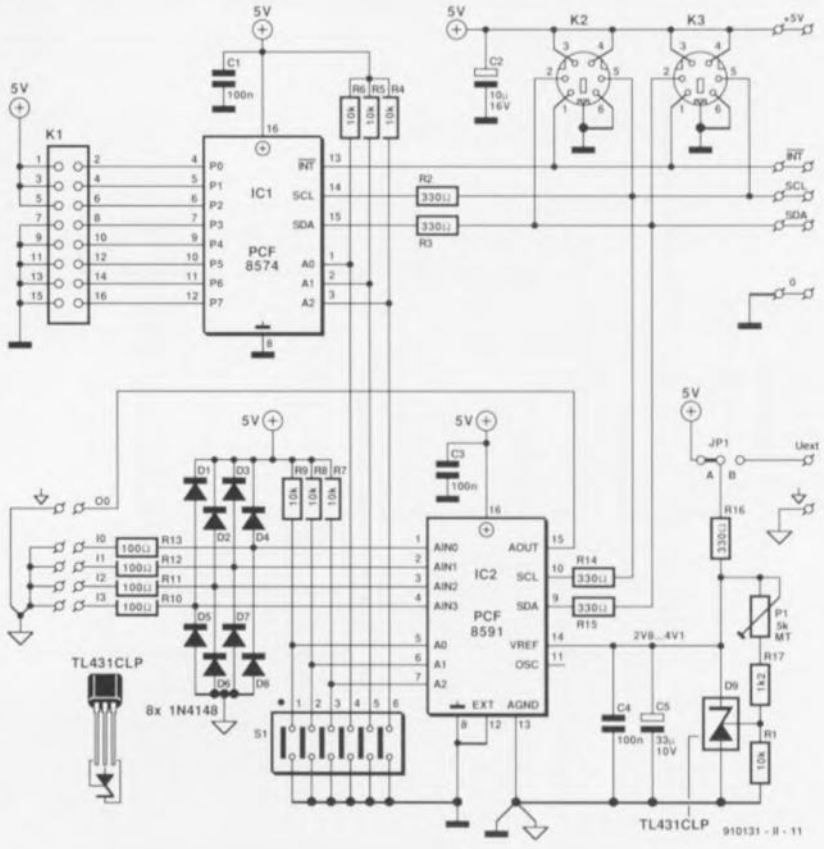


Figure 1. L'électronique de la carte de conversion pour bus I²C comporte un port d'E/S et un convertisseur bidirectionnel, A/N et N/A. La commutation avec le PC se fait via l'interface I²C pour PC décrite le mois dernier.

moins, définies sur le circuit même, lors du processus de fabrication. Les adresses des 2 types de circuits utilisés ici sont :

PCF8574 = 0100 A2 A1 A0 R/$\overline{W}$
PCF8791 = 1001 A2 A1 A0 R/$\overline{W}$

Dans les 2 cas, les 4 premiers bits sont définis sur la puce, les 3 bits suivants pouvant être définis à l'aide des contacts correspondants de l'interrupteur DIL, le dernier bit déterminant lui le traitement de la donnée, écriture ou lecture.
Si la ligne R/$\overline{W}$ est à «1» c'est que l'on se trouve en présence d'un processus de lecture, si elle est à «0» c'est qu'il s'agit d'une écriture.
Les routines standard présentes sur la disquette supposent que tous les contacts de l'interrupteur DIL sont fermés et que les circuits intégrés se trouvent donc respectivement aux adresses hexadécimales \$40, & \$41 et \$90 & \$91.
Si vous envisagez d'utiliser d'autres adresses il vous faudra penser à modifier ces adresses dans le logiciel, opération qui ne devrait pas poser de problème vu la documentation dont vous disposez.

Le port d'E/S quasi-bidirectionnel PCF8574 dont on retrouve la structure interne en **figure 2**, ne possède qu'un registre d'écriture et de lecture. En effet, selon l'application concernée, il est également possible d'utiliser une sortie en entrée. Toute sortie est capable de drainer (*sink*) jusqu'à 25 mA, sachant qu'elle peut fournir (*source*) un courant de 0,4 mA au maximum. Lorsque l'on veut utiliser une broche en entrée on commence par y écrire un «1» logique. Pour ce faire, on fait passer la sortie au niveau logique haut. Il suffit ensuite d'une interrogation de la broche concernée pour voir si la sortie utilisée en entrée se trouve au niveau haut ou au niveau bas. Si la broche présente un niveau bas, cela signifie qu'elle est mise au niveau bas par un élément externe. Étant donnée la structure interne du circuit (sortie en drain ouvert) il n'y a pas de risque de faire de dégâts si l'on procède de cette façon.
Notons en passant que les microcontrôleurs utilisent souvent ce type de sortie.

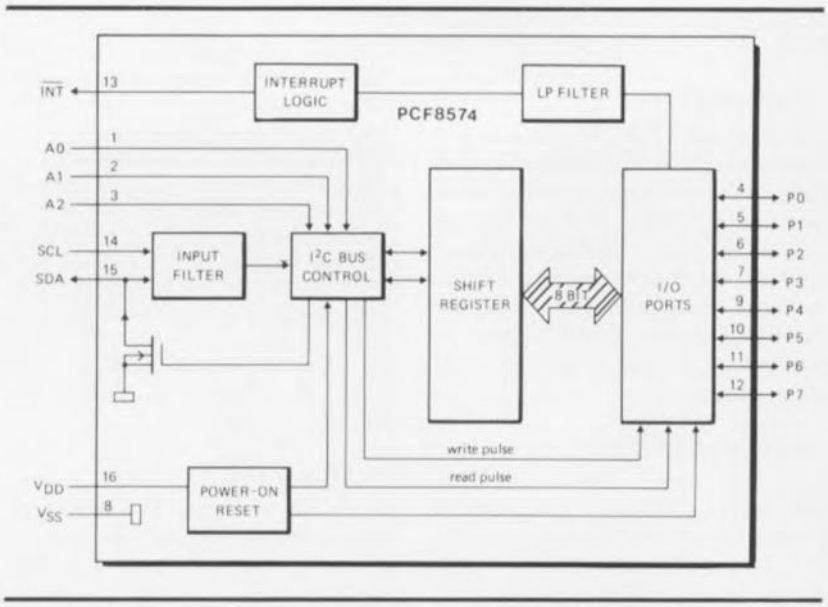


Figure 2. Structure interne du PCF8574, le circuit intégré d'E/S utilisé sur cette carte d'extension.

La circuiterie entourant le PCF8591 semble, à première vue, plus complexe mais, rassurez-vous, les apparences sont trompeuses. Un grand nombre des composants présents (R10 à R13 et D1 à D8) servent à protéger les entrées analogiques du convertisseur A/N – N/A contre d'éventuelles surtensions. Ici à nouveau, les 3 lignes d'adresse (A0 à A2) sont reliées aux contacts d'un sextuple interrupteur DIL, grâce auxquels l'utilisateur peut définir lui-même la partie variable de l'adresse.

Il nous faut en outre quelques composants pour générer une tension de référence parfaitement stable. D9, un TL431CLP, est une diode zener ayant l'incomparable avantage d'être tout à la fois précise et peu onéreuse. En association avec les résistances R1, R16, R17 et l'ajustable P1, cette diode permet de définir une tension de référence comprise entre 2,8 et 4,1 V. L'utilisateur peut opter pour n'importe quelle tension située à l'intérieur de ce domaine et choisir ainsi la taille

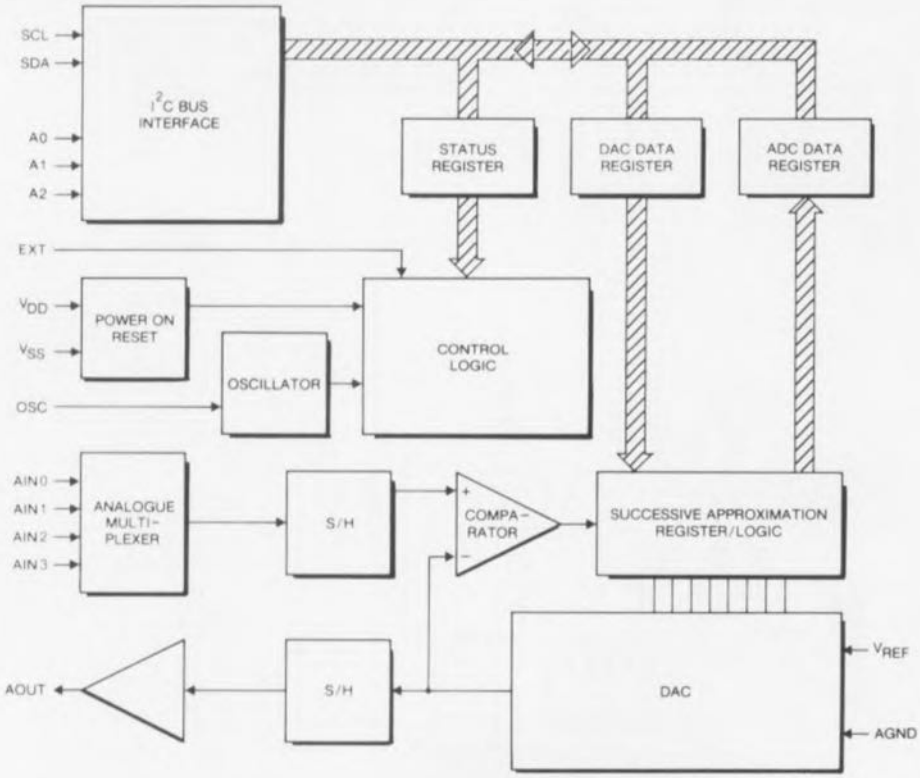


Figure 3. Le convertisseur bidirectionnel N/A et A/N PCF8591 présente une structure interne plus complexe que le PCF8574. Une source de tension de référence externe permet la définition de la taille de pas du convertisseur A/N & N/A.

de pas pour le convertisseur A/N et N/A. Un pas correspond en effet à $V_{ref}/256$. Les condensateurs C4 et C5 débarrassent la tension de référence des dernières petites irrégularités qu'elle pourrait comporter. Le cavalier de court-circuit **JP1** offre la possibilité de faire appel à une source de tension externe destinée à fournir la tension nécessaire à la diode de référence. Cette approche ne se justifie vraiment que lorsque l'on envisage de procéder à des mesures précises et qu'il existe un risque réel de présence de pics de tension parasite sur la ligne +5 V. Il n'est donc pas, d'habitude, nécessaire de prendre de précautions particulières.

La structure interne du convertisseur A/N–N/A, illustrée par la **figure 3**, est elle également plus complexe que celle du port d'E/S. L'émission des données en direction du PCF8591 est, de par l'existence d'un nombre de possibilités plus important, quelque peu plus compliquée que dans le cas du PCF8574 décrit quelques lignes plus haut.
Il faut à ce circuit intégré, outre les octets de données aussi un octet de commande (*control byte*) chargé de définir un certain nombre de conditions.
La **figure 4** –que nous avons laissée en anglais non par fainéantise, mais pour vous permettre de vous «faire la main» en vue de la lecture de recueils de caractéristiques– décrit la fonction de chacun des bits de notre octet.

Le quartet (*nibble*) de poids fort de l'octet de commande (*control byte*) sert à définir la configuration des entrées analogiques (2 entrées différentielles ou 4 entrées normales) et à effectuer la validation de la sortie analogique. Le quartet de poids faible sert à la sélection de l'une des 4 entrées A/N et peut servir à la validation de l'indicateur d'auto-incrément (*autoincrement flag*).
Le troisième octet à être envoyé vers le circuit intégré, après l'octet d'adresse et l'octet de commande donc, est stocké dans le registre du convertisseur N/A, la valeur se trouvant précédemment dans ce registre étant alors convertie en une tension continue analogique qui fait son apparition à la sortie du convertisseur N/A. La tension de sortie augmente, comme nous le disions plus haut, par pas de $V_{ref}/256$. Si la valeur évoquée tout juste est de 00, on trouve à la sortie une tension de 0 V, si au contraire elle est de 255, on aura à la sortie d'une tension égale aux 255/256 de la tension de référence.

La prise en compte des signaux du convertisseur A/N se fait quelque peu différemment. Le processus de conversion A/N est démarré par le flanc montant de l'impulsion d'acquittement (*acknowledge*) envoyée après la mise du circuit intégré en mode de lecture par l'émission d'une instruction de lecture. Pendant le temps nécessaire à l'émission des données de la conversion précédente, le convertisseur effectue une nouvelle conver-

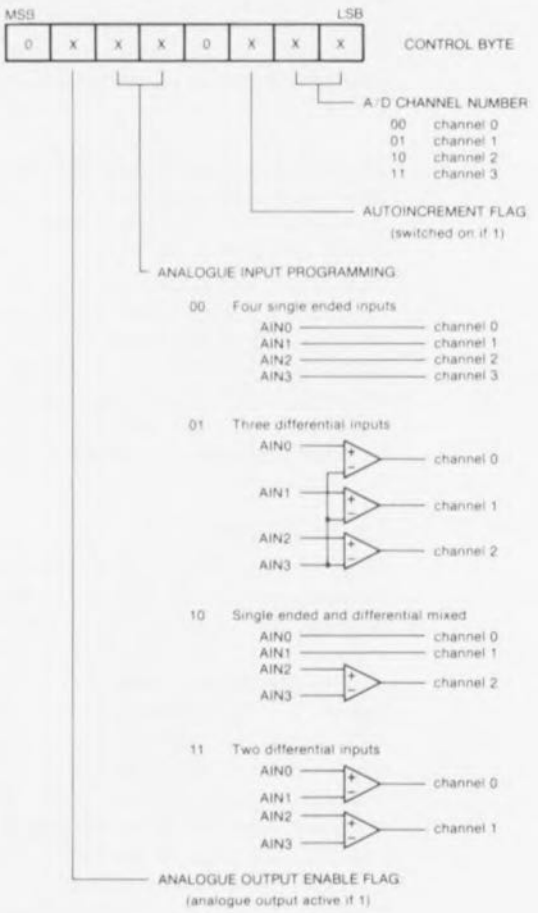


Figure 4. L'octet de commande nécessaire au PCF8591 permet le choix de l'une des différentes options possibles offertes par ce circuit intégré.

Liste des composants

Résistances:
R1,R4 à R9 = 10 kΩ
R2,R3,R14 à R16 = 330 Ω
R10 à R13 = 100 Ω
R17 = 1kΩ2
P1 = 5 kΩ ajustable multitour

Condensateurs:
C1,C3,C4 = 100 nF
C2 = 10 µF/16 V
C5 = 33 µF/10 V

Semi-conducteurs:
D1 à D8 = 1N4148
D9 = TL431CLP (Texas Instruments)
IC1 = PCF8574 (Philips)
IC2 = PCF8591 (Philips)

Divers:
K1 = connecteur HE-10 mâle, 2 x 8 contacts coudés, encartable
K2,K3 = embase DIN miniature, 6 contacts, encartable
S1 = interrupteur DIL, 6 contacts
2 fiches DIN miniatures, 6 contacts

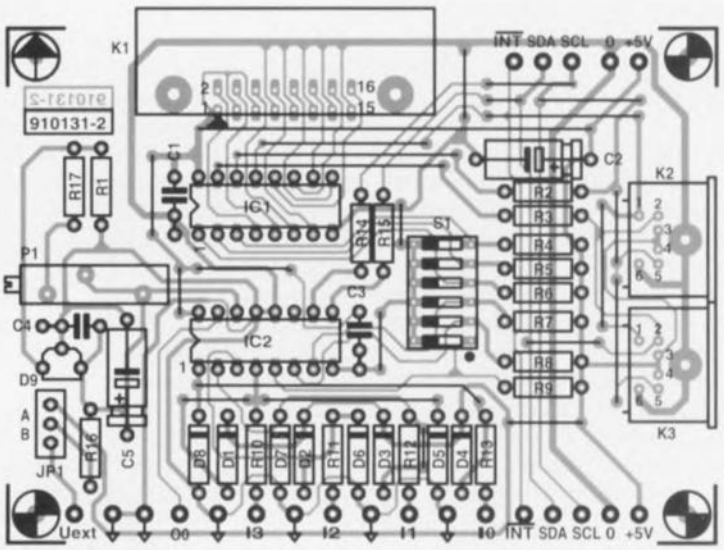


Figure 5. Représentation de la sérigraphie de l'implantation des composants de la platine dessinée pour cette réalisation. Les mini-embases DIN à 6 broches facilitent énormément l'interconnexion des différents circuits imprimés à l'interface I²C pour PC.

sion A/N. Lors de l'impulsion de départ (*start pulse*) de conversion on a échantillonnage du niveau de tension présent sur l'entrée choisie, le résultat de ce processus étant ensuite converti en un code binaire à 8 bits.

Les tensions d'une entrée différentielles sont converties en un code de complément à 2 (*2 complement*) à 8 bits. Le résultat est stocké dans le registre de donnée du convertisseur A/N, d'où il peut ensuite être émis vers une autre destination. Si l'indicateur d'auto-incrément est activé, on a sélection ensuite de l'entrée suivante. C'est ainsi que l'on réalise un échantillonnage successif de toutes les entrées.

La réalisation

Nous avons conçu, à l'intention de ce montage, un dessin de circuit imprimé dont on retrouve la représentation de la sérigraphie de l'implantation des composants en **figure 5**. L'utilisation d'embases DIN miniatures transforme «en un jeu d'enfant» la connexion entre l'interface présente dans le PC et notre **convertisseur A/N – N/A pour bus I²C**. On peut se contenter, en principe, de ne monter qu'une des 2 embases (K2 ou K3) sur la platine, sachant que la seconde embase sert à la mise en série de plusieurs montages I²C.

Comme l'illustre la photo de la couverture du numéro du mois dernier, il est plus économique, dès que l'on envisage de procéder à la connexion de plusieurs platines à l'intérieur d'un même coffret, d'interconnecter les lignes +5 V, 0, SCL, SDA et $\overline{\text{INT}}$ à l'aide de faibles longueurs de fil de câblage semi-rigide.

Le brochage choisi pour le connecteur K1 est tel qu'il correspond très exactement à celui du connecteur présent sur l'**amplificateur de mesure universel pour convertisseurs A/N** (Elektor n°163, janvier 1992). On pourra utiliser un petit morceau de câble plat pour effectuer les connexions nécessaires entre cette platine et l'amplificateur de mesure. Il devient possible ensuite de commander l'amplificateur de mesure uniquement par logiciel par l'intermédiaire de la carte décrite dans cet article.
L'examen attentif du code source du programme de test **ADIO** aura vite fait de vous apprendre qu'il est extrêmement facile de l'utiliser pour constituer un système de mesure à calibration automatique.

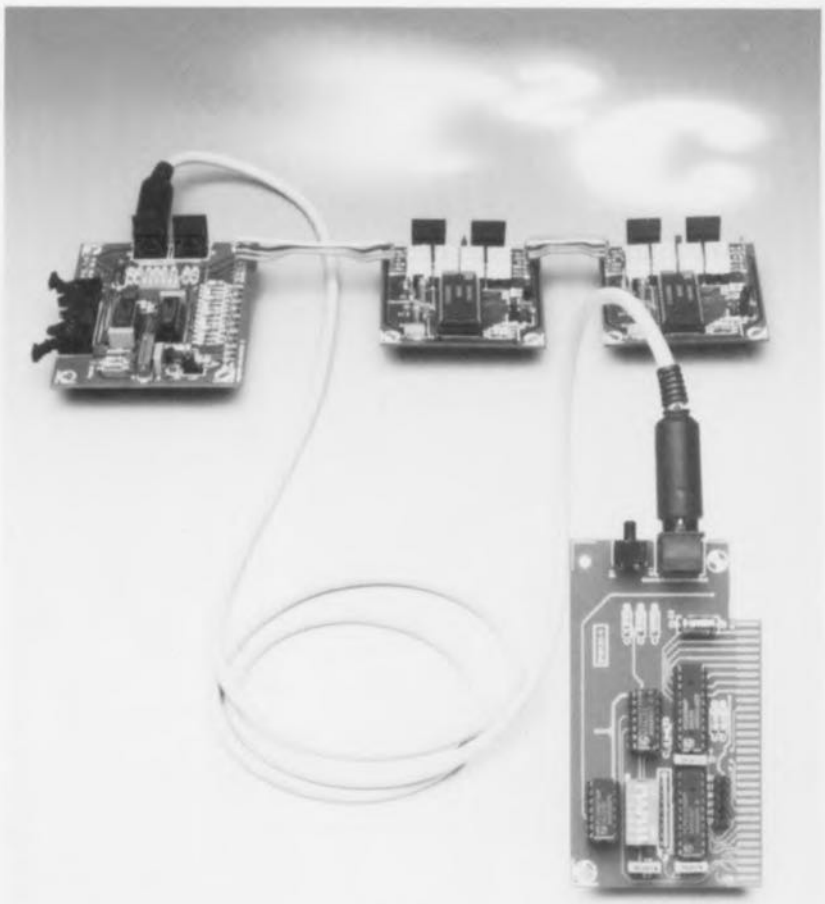


Figure 6. Cette interconnexion de démonstration montre comment s'y prendre pour relier les différents modules, par l'intermédiaire d'une unique interface. Le circuit de commande d'affichage sera décrit dans un prochain article.

Les contacts du sextuple interrupteur DIL définissent les adresses des 2 circuits dans le système I²C. On pourra laisser tous les contacts fermés pour définir l'adressage standard utilisée par le programme de test.

On ajuste, par une action sur la résistance ajustable P1, la tension de référence à la valeur requise. Pour mesurer précisément la valeur de cette tension on branchera un multimètre numérique – placé en calibre voltmètre bien entendu – aux bornes du condensateur C5.

Le programme **ADIO** présent sur la disquette considère que la tension de référence adoptée est de 4,0 V. Il est donc recommandé, pour les premiers essais, d'opter pour une tension de référence de la dite valeur. Lorsque tout marche à souhait, on pourra choisir une valeur de tension de référence différente (en n'oubliant pas alors d'adapter en conséquence la déclaration de variable dans le programme).

Vous venez de mener à bonne fin les étapes de construction et d'étalonnage de ce montage qui est prêt maintenant à remplir la tâche pour laquelle il a été conçu.

Il ne nous reste plus qu'à tester l'interface. Cette opération constitue en même temps un cours accéléré (du Berliz-minute en quelque sorte) d'utilisation du bus I²C en combinaison avec Turbo Pascal et C.

C'est d'ailleurs là la raison de la présence sur la disquette d'un certain nombre de modules de programmes bien documentés, écrits en Turbo Pascal et en C. Le programme de test ne cesse d'exécuter les mêmes routines. On a ainsi lecture des niveaux présents sur les entrées d'E/S b4 à b7 et mise sur les sorties b0 à b3 des niveaux logiques mesurés.
On pourra, lors des essais, mettre les entrées b4 à b7 au niveau logique haut en s'aidant de résistances de forçage de 10 kΩ (*pull up*) et se servir de 4 boutons-poussoirs, pris entre chacune des entrées et la masse, pour mettre l'entrée au niveau bas par action sur l'une de ces touches. Si maintenant on connecte une LED à chacune des sorties b0 à b3, (cathode reliée au port d'E/S, l'anode l'étant au +5 V

via une résistance de 330 Ω), on verra s'allumer, lorsque l'on appuiera sur la touche concernée, la LED correspondant à la touche connectée.

Le processus de test du convertisseur A/N – N/A présente certaines similitudes avec la technique d'essais utilisée plus haut. Le logiciel lit les niveaux de tension présents aux entrées analogiques I0 à I3 et place la valeur de I3 sur la sortie analogique O0.

Si tous les tests ont été passés avec succès et que donc votre réalisation fonctionne comme il le faut, vous disposez maintenant d'un montage utilisable pour d'autres applications (qu'il vous faudra concevoir vous-même le cas échéant). Bonne chance !

Si, sur le Minitel, vous envoyez un message à Elektor, n'ayez pas de fausse honte et indiquez au minimum votre nom de manière à ce que la rédaction puisse répondre publiquement à vos questions.

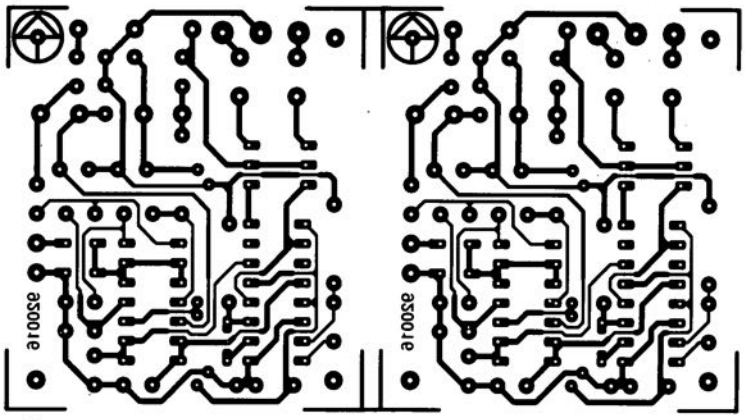
Signé La Rédaction

```
begin (* TestADDA *)
  Ctrl:=GetControlByte(1);          {-Load control byte with option# 1}
  Address(ADA_Addr);                {-Because the R/W_ bit (= LSB ADA_Addr)
                                      equals zero, the PCF8591 enters the
                                      write-mode. Therefore, the next trans-
                                      mitted byte is interpreted as a
                                      control byte.}
  write(bus,Ctrl);                  {-Transmit control byte.}
  with AD do                        {-The next bytes sent to the PCF8591
                                      would be stored in the DAC register.
                                      But at this point, we switch to
                                      read-mode...}
    read(Bus,Dummy,Data[0],Data[1],Data[2],Data[3]); {-I2CDRIV.SYS now
                                      generates a repeated start (same
                                      address, but with R/W_ = 1) and reads
                                      five AD-conversion bytes from the
                                      PCF8591. This is done by using the
                                      channel auto-increment function of
                                      the chip. The first read byte (Dummy)
                                      is the convertion result code of the
                                      previous cycle! We are not interested
                                      in that sample, so it is thrown away.}

  write(bus,Ctrl,AD.Data[Chan3]);   {-I2CDRIV.SYS generates a repeated
                                      start condition; same address, but
                                      with R/W_ = 0. Thus, the PCF8591 is
                                      in write-mode again and expects a
                                      control byte and one or more data-
                                      bytes. All databytes are stored in
                                      the DAC-register, but the analogue
                                      output voltage is always calculated
                                      from the previous DAC-register
                                      contents.}

  for Channel:=Chan0 to Chan3 do
    ShowVoltage(Channel,AD.Data[Channel]);
  ShowVoltage(4,AD.Data[Chan3]);
end;  (* TestADDA *)
```

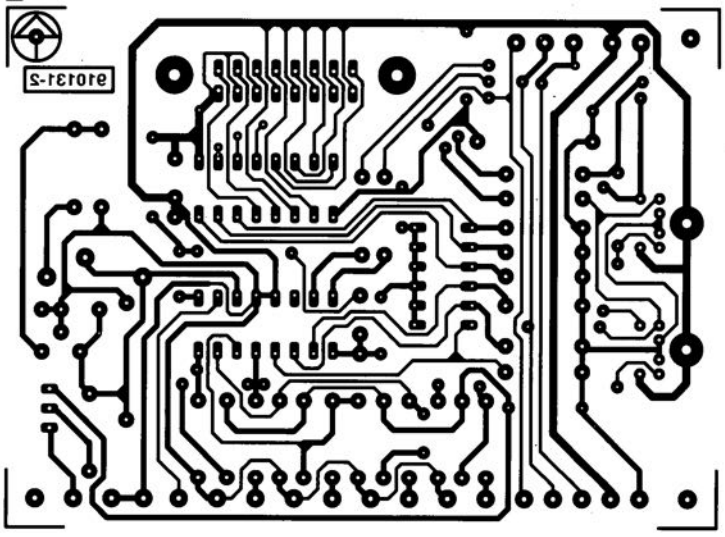
Figure 7. (Petit) extrait du programme ADIO écrit en Turbo Pascal, langage grâce auquel il est relativement facile de commander le bus I²C.

1 témoin d'écrêtage
2 convertisseur A/N-N/A pour I²C
3 convertisseur RS-232-A/N

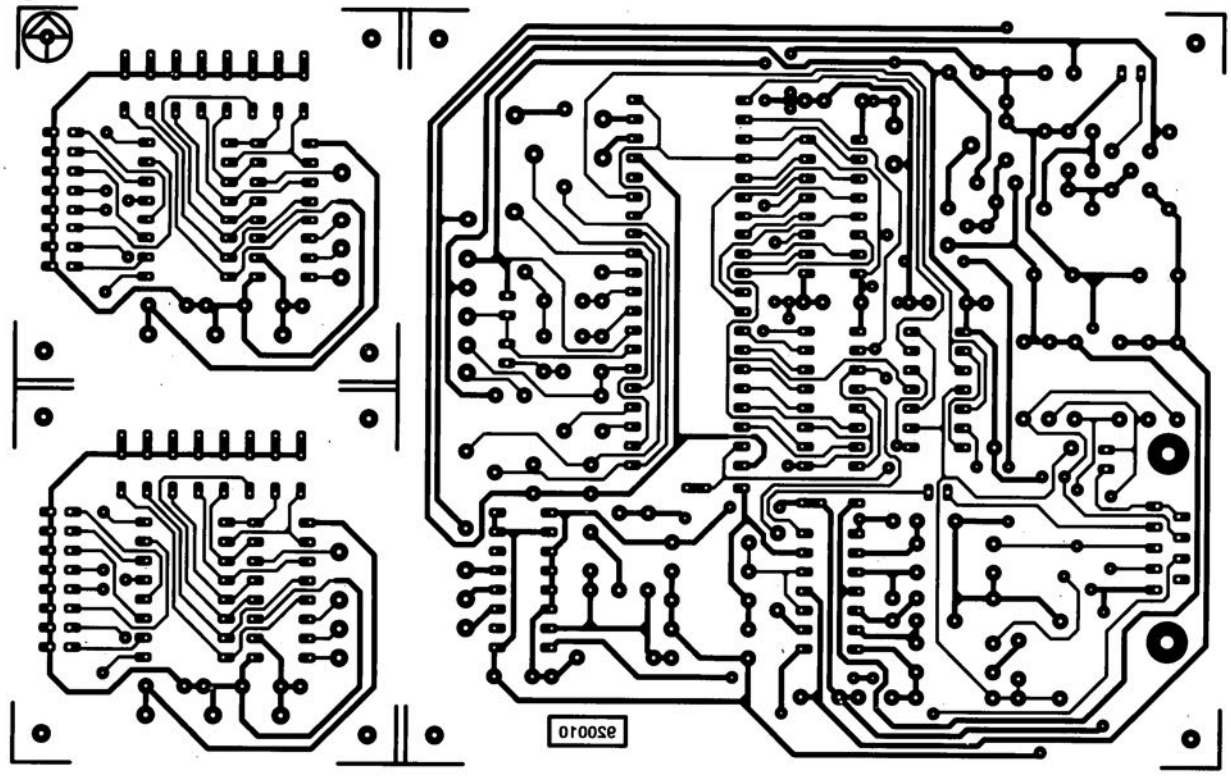
1



2

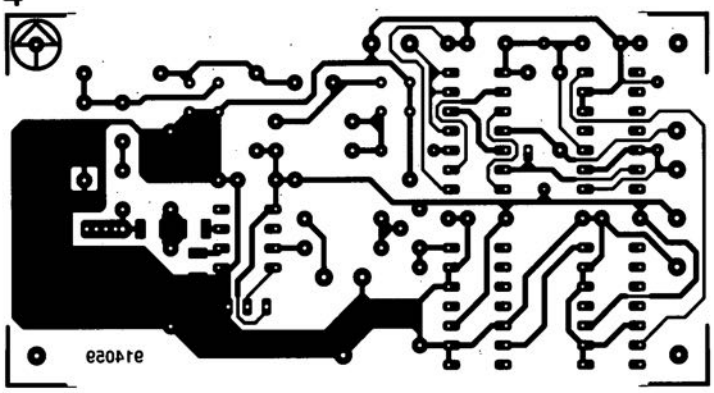


3

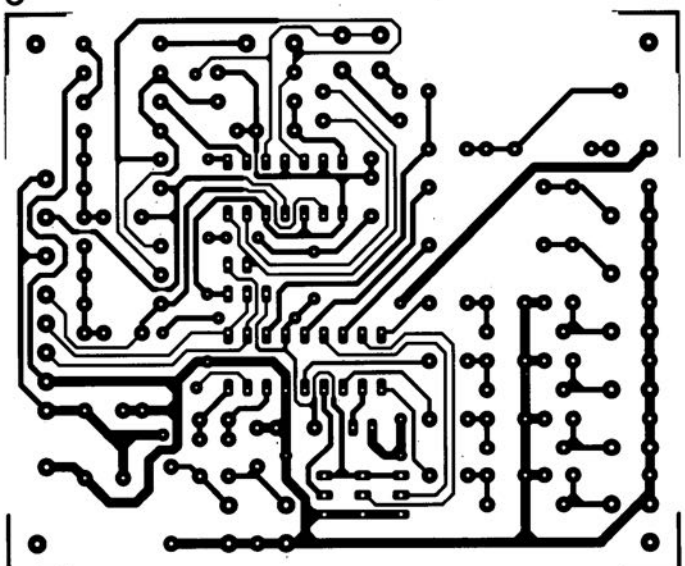


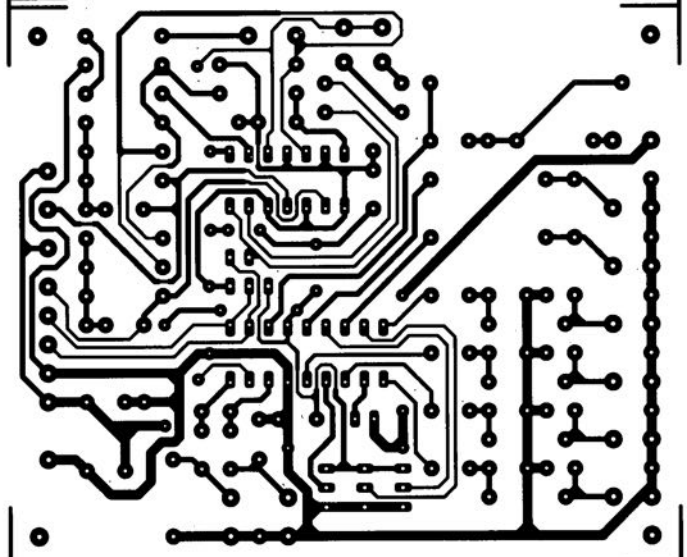
4 prédiviseur 1,3 GHz
5 commutateur audio/vidéo électronique
6 émulateur de 8751 : côté composants
7 émulateur de 8751 : côté pistes

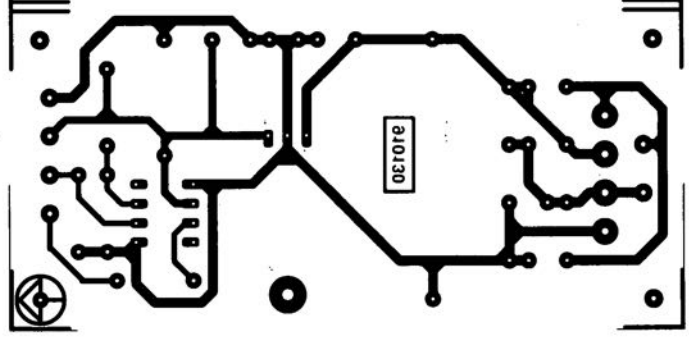
4



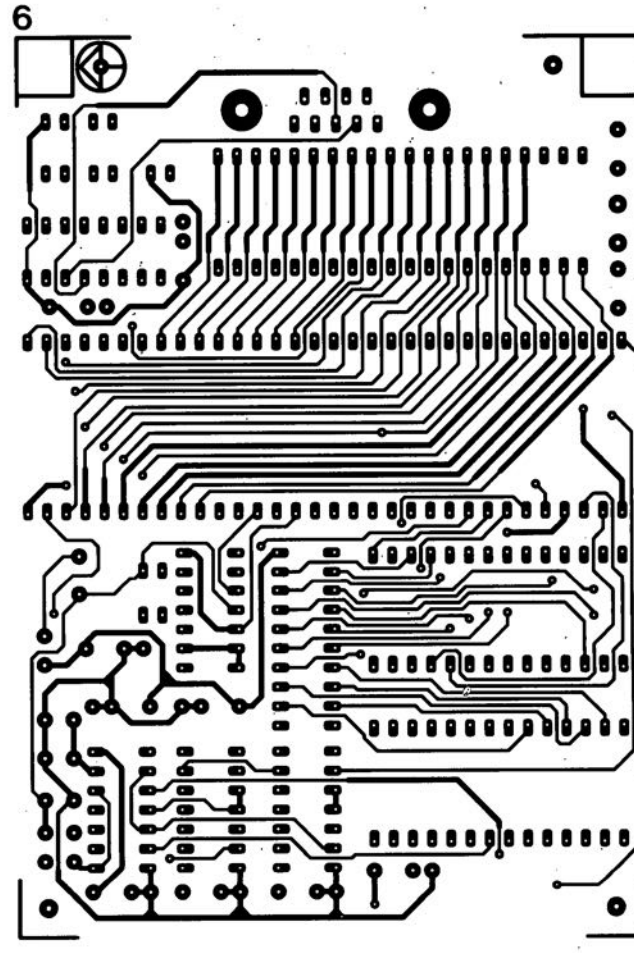
5



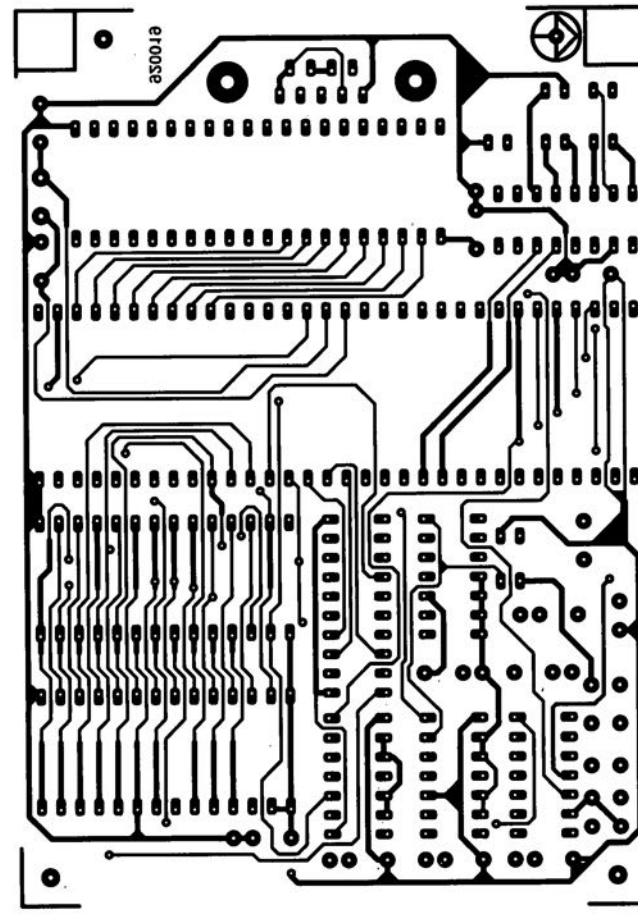




6



7



Seules des questions ou des remarques présentant un intérêt général pour les lecteurs d'Elektor et concernant des montages publiés dans Elektor au cours des 2 dernières années peuvent entrer en ligne de compte pour une éventuelle réponse. La rédaction se réserve le droit de procéder à une sélection des lettres ou d'en raccourcir le contenu. Vu l'abondant courrier nous arrivant chaque semaine, il n'est malheureusement pas possible de répondre à toutes les lettres et nous est totalement impossible de répondre aux souhaits et demandes d'adaptation ou d'information additionnelle concernant des projets décrits dans ce magazine.

Des questions ou des remarques?

Elektor service «Courrier Lecteurs» BP 59 59850 Nieppe

Interface I²C

Il ne se passe pas de semaine où ne recevons pas l'une ou l'autre lettre de lecteur ayant des problèmes avec leur interface I2C. Ceci nous pousse à récapituler les situations à problème et les messages d'erreur qu'elles peuvent générer.

Interface encartable I²C pour PC :

Dans l'article du mois de février intitulé « convertisseur A/N–N/A et E/S pour I²C » publié dans le numéro 164 (février 1992) nous indiquions comment utiliser le logiciel accompagnant la dite interface décrite elle dans le numéro 161 (janvier 1992). Il y était dit d'intégrer la ligne DEVICE=I2CDRIV.SYS dans le fichier de configuration CONFIG.SYS. Notons que la nouvelle version de ce programme s'appelle I2CDRIV2.SYS, il faudra donc faire, aujourd'hui, DEVICE=I2CDRIV2.SYS. Il n'est pas nécessaire, dans la plupart des cas, d'indiquer de paramètres additionnels; la version actuelle est la version 1.1 qui permet la définition de 3 paramètres; il se peut que certains PC à bus interne très rapide nécessitent la définition du 3ème paramètre. On se trouve alors apparaître la ligne suivante (qui est d'ailleurs également celle que l'on obtient en l'absence de définition de paramètres : DEVICE=I2CDRIV2.SYS B:300 C:2 D:00FF

Après mise en fonction de l'ordinateur on voit apparaître le message suivant :

I2C-bus driver V1.1-ESS182X installed
Base I/O address=300h
Clock freq I2C-bus=9.8kHz
Nops in PCD8584 IO=00FFh

N.B. Ce message apparaît même en l'absence de l'interface encartable à 8584 et même si en l'absence de sélection de l'adresse correcte. Si l'on lance le programme en Pascal ADIO.EXE et que la carte à 8584 est absente, non fonctionnelle ou située à une adresse incorrecte on verra apparaître le message suivant :

I/O error code 162
Probably I2CDRIV.SYS not installed. Add it to CONFIG.SYS and reboot.

Ce message peut prêter à confusion vu qu'il suggère, dans certains cas à tort, l'absence du driver – qui, notons-le en passant, s'appelle, depuis sa version 1.1, I2CDRIV**2**.SYS. En cas d'apparition de ce message il faudra commencer par vérifier le platine de l'interface et s'assurer que l'adresse définie à l'aide de S2 n'est pas déjà utilisée par une autre carte implantée dans le PC. On a défini l'adresse à 300HEX lorsque l'on a fermé tous les 7 interrupteurs DIL de S2. Si l'on veut définir une autre adresse il faudra entrer le code binaire correspondant. Supposons que l'on veuille fixer l'adresse 304HEX à l'aide de S2 : il faudra dans ce cas-là ouvrir l'interrupteur n°3 de S2. En cas de choix d'une adresse différente il faudra utiliser cette nouvelle adresse comme paramètre B dans la ligne DEVICE=I2CDRIV2.SYS du fichier CONFIG.SYS.

Les cartes I²C.

Si l'interface encartable est en bon état, mais qu'il y a des problèmes du côté du bus I²C dûs, par exemple à l'absence de réaction du convertisseur AN/NA, on aura apparition du message suivant :

DOS error code 21
Probably IC not present, start() not generated or wrong chip address

Il vous faudra dans ce cas-là bien vérifier les platines de l'interface I²C encartable et du convertisseur A/N-N/A et E/S pour I²C. Portez votre examen sur l'éventuelle absence de l'un ou l'autre pont de câblage sur la seconde platine, vérifiez la présence de la tension d'alimentation et assurez-vous que l'adresse définie est l'adresse correcte. Notons qu'il existe, pour certains des circuits intégrés, plusieurs versions; on a ainsi une version 8574A pour le 8574, ayant d'autres adresses ! Attention donc ! Si l'on veut faire tourner les programmes en Pascal et en C tels qu'ils sont à l'origine, sans les avoir modifiés donc, il faudra avoir mis les circuits intégrés à l'adresse « 0 » à l'aide des interrupteurs DIL, ce qui implique que tous les interrupteurs doivent être fermés. Si vous optez pour une adresse différente il vous faudra modifier en conséquence les adresses des programmes en Pascal et en C. Ceci est l'une des raisons de la présence sur la disquette des codes-source des programmes en question. Attention, les programmes en Pascal et en C ne sont pas nécessairement identiques. La description du programme ADIO donnée dans le second article concerne la version en Pascal de ce programme. Le programme en C se traduit par une effet de chenillard sur le bus parallèle K1. On a en plus incrémentation d'un compteur binaire à 8 bits dont la valeur est envoyée au convertisseur N/A. On produit ainsi une tension en dents de scie sur la sortie analogique 00.

(La rédaction)

Pocket applications

C'est parce que j'apprécie beaucoup votre revue que je me permets de vous adresser ce courrier, afin de vous présenter mon fanzine. Celui-ci est né voilà plus de 2 ans.... ...Maintenant que la « revue » possède une structure adéquat (association loi 1901), il ne lui manque qu'une chose : se faire vraiment connaître. Ainsi, si le numéro joint vous paraît justifier une mention, même infime, dans vos colonnes, je serais très heureux que mon fanzine fasse sa percée grâce à votre magazine. Peut-être est-il utile d'ajouter qu'il paraît tous les 3 mois et que le dernier numéro est disponible contre 3 timbres à 2,80 FF.

W. Klinger,
le Petit Tinzin 38210 Tullins

Seules des questions ou des remarques présentant un intérêt général pour les lecteurs d'Elektor et concernant des montages publiés dans Elektor au cours des 2 dernières années peuvent entrer en ligne de compte pour une éventuelle réponse. La rédaction se réserve le droit de procéder à une sélection des lettres ou d'en raccourcir le contenu. Vu l'abondant courrier nous arrivant chaque semaine, il n'est malheureusement pas possible de répondre à toutes les lettres et nous est totalement impossible de répondre aux souhaits et demandes d'adaptation ou d'information additionnelle concernant des projets décrits dans ce magazine.

Des questions ou des remarques?

Elektor service «Courrier Lecteurs» BP 59 59850 Nieppe

Interface I²C

Il ne se passe pas de semaine où ne recevons pas l'une ou l'autre lettre de lecteur ayant des problèmes avec leur interface I2C. Ceci nous pousse à récapituler les situations à problème et les messages d'erreur qu'elles peuvent générer.

Interface encartable I²C pour PC :

Dans l'article du mois de février intitulé « convertisseur A/N–N/A et E/S pour I²C » publié dans le numéro 164 (février 1992) nous indiquions comment utiliser le logiciel accompagnant la dite interface décrite elle dans le numéro 161 (janvier 1992). Il y était dit d'intégrer la ligne DEVICE=I2CDRIV.SYS dans le fichier de configuration CONFIG.SYS. Notons que la nouvelle version de ce programme s'appelle I2CDRIV2.SYS, il faudra donc faire, aujourd'hui, DEVICE=I2CDRIV2.SYS. Il n'est pas nécessaire, dans la plupart des cas, d'indiquer de paramètres additionnels; la version actuelle est la version 1.1 qui permet la définition de 3 paramètres; il se peut que certains PC à bus interne très rapide nécessitent la définition du 3ème paramètre.

On se trouve alors apparaître la ligne suivante (qui est d'ailleurs également celle que l'on obtient en l'absence de définition de paramètres : DEVICE=I2CDRIV2.SYS B:300 C:2 D:00FF

Après mise en fonction de l'ordinateur on voit apparaître le message suivant :

I2C-bus driver V1.1-ESS182X installed
Base I/O address=300h
Clock freq I2C-bus=9.8kHz
Nops in PCD8584 IO=00FFh

N.B. Ce message apparaît même en l'absence de l'interface encartable à 8584 et même si en l'absence de sélection de l'adresse correcte. Si l'on lance le programme en Pascal ADIO.EXE et que la carte à 8584 est absente, non fonctionnelle ou située à une adresse incorrecte on verra apparaître le message suivant :

I/O error code 162
Probably I2CDRIV.SYS not installed. Add it to CONFIG.SYS and reboot.

Ce message peut prêter à confusion vu qu'il suggère, dans certains cas à tort, l'absence du driver – qui, notons-le en passant, s'appelle, depuis sa version 1.1, I2CDRIV**2**.SYS. En cas d'apparition de ce message il faudra commencer par vérifier le platine de l'interface et s'assurer que l'adresse définie à l'aide de S2 n'est pas déjà utilisée par une autre carte implantée dans le PC. On a défini l'adresse à 300HEX lorsque l'on a fermé tous les 7 interrupteurs DIL de S2. Si l'on veut définir une autre adresse il faudra entrer le code binaire correspondant. Supposons que l'on veuille fixer l'adresse 304HEX à l'aide de S2 : il faudra dans ce cas-là ouvrir l'interrupteur n°3 de S2. En cas de choix d'une adresse différente il faudra utiliser cette nouvelle adresse comme paramètre B dans la ligne DEVICE=I2CDRIV2.SYS du fichier CONFIG.SYS.

Les cartes I²C.

Si l'interface encartable est en bon état, mais qu'il y a des problèmes du côté du bus I²C dûs, par exemple à l'absence de réaction du convertisseur AN/NA, on aura apparition du message suivant :

DOS error code 21
Probably IC not present, start() not generated or wrong chip address

Il vous faudra dans ce cas-là bien vérifier les platines de l'interface I²C encartable et du convertisseur A/N-N/A et E/S pour I²C. Portez votre examen sur l'éventuelle absence de l'un ou l'autre pont de câblage sur la seconde platine, vérifiez la présence de la tension d'alimentation et assurez-vous que l'adresse définie est l'adresse correcte. Notons qu'il existe, pour certains des circuits intégrés, plusieurs versions; on a ainsi une version 8574A pour le 8574, ayant d'autres adresses ! Attention donc !

Si l'on veut faire tourner les programmes en Pascal et en C tels qu'ils sont à l'origine, sans les avoir modifiés donc, il faudra avoir mis les circuits intégrés à l'adresse « 0 » à l'aide des interrupteurs DIL, ce qui implique que tous les interrupteurs doivent être fermés. Si vous optez pour une adresse différente il vous faudra modifier en conséquence les adresses des programmes en Pascal et en C. Ceci est l'une des raisons de la présence sur la disquette des codes-source des programmes en question. Attention, les programmes en Pascal et en C ne sont pas nécessairement identiques. La description du programme ADIO donnée dans le second article concerne la version en Pascal de ce programme. Le programme en C se traduit par une effet de chenillard sur le bus parallèle K1. On a en plus incrémentation d'un compteur binaire à 8 bits dont la valeur est envoyée au convertisseur N/A. On produit ainsi une tension en dents de scie sur la sortie analogique O0.

(La rédaction)

AD 7886

Convertisseur analogique numérique 12 bits 750 kHz ne consommant que 350 mW

Analog Devices commercialise un CAN 12 bits monolithique, avec échantillonneur-bloqueur intégré, ne consommant que 350 mW, la plus faible consommation qui soit pour un CAN aussi rapide. L'AD 7886 consomme 33% de moins que les solutions monolithiques comparables et nécessite moins de composants externes pour fonctionner et s'interfacer à un DSP.

Le temps de conversion de ce nouveau CAN est de 1 µs et le temps d'accès aux données de 57 ns. Il trouvera des applications dans l'instrumentation rapide ou dans les systèmes multivoies basses fréquences.

L'AD 7886 utilise une architecture flash triple passes. En plus de ses excellentes caractéristiques statiques, il est entièrement spécifié en dynamique:

La non-linéarité intégrale maximum est de ±2 LSB,

Il est garanti sans code manquant,

Le rapport signal sur (bruit + distorsion) est de 65 dB et la distorsion harmonique totale de 75 dB, ce qui sera apprécié pour les applications d'analyse spectrale comme la reconnaissance vocale par exemple.

La distorsion d'intermodulation d'ordre deux et trois est typiquement de −80 dB.

L'AD 7886 nécessite des tensions d'alimentation de ±5 V; la plage de tension d'entrée, quant à elle, est de 0 à 5 V, 0 à 10 V ou ±5 V (sélectionnable). Ce circuit est disponible en boîtier 28 broches DIP ou LCC.

Analog Devices
3, rue Goerges Besse
CE 27
92182 Antony Cedex
tél.: (1).46.66.25.25
fax.: (1).46.66.24.12

Tampon pour imprimante 1 ou 4 Moctets

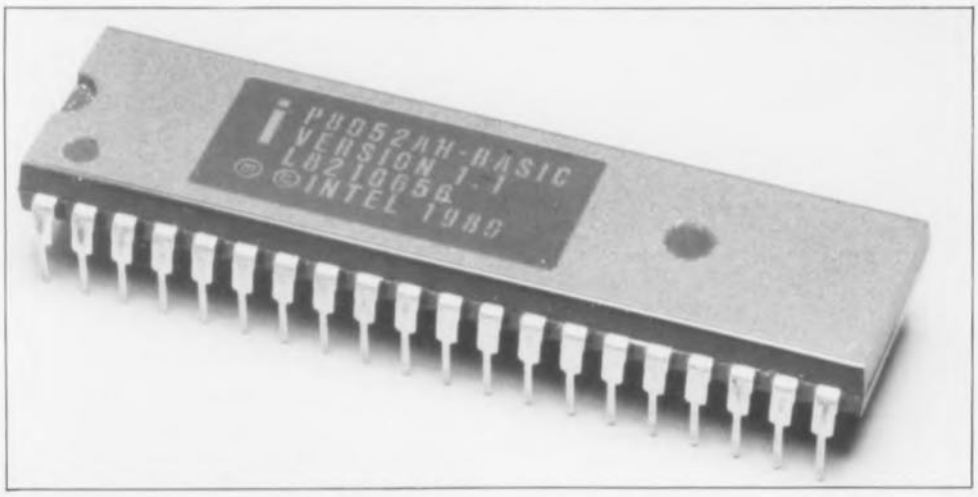
Elektor n°161, novembre 1991, page 46

Le tampon pour imprimante connaît le succès auquel nous nous attendions. Qui dit succès sous-entend presque inévitablement problèmes. Il semblerait que la première version de l'ESS 6044 ait quelques déconvenues avec les ordinateurs à base de 80486 nous disent plusieurs lecteurs. Nous n'en avons pas détecté avec des systèmes à autres µP. Si donc vous rencontrez un problème avec votre tampon pour imprimante, qu'il soit relié à un 486 ou non, et que vous pensez avec un bon pourcentage de certitude que cela puisse être dû à l'EPROM, renvoyez-la sous enveloppe à PUBLITRONIC avec une enveloppe affranchie au port convenable pour le retour et nous vous la ferons reprogrammer dans les plus brefs délais.

améliorer le MCS® BASIC-52 V1.1

2e partie

Z. Stojsavljevic & D. Mudric



Dans l'article du mois dernier nous avons examiné et corrigé la routine du noyau de virgule flottante et celle de la conversion hex-BCD de l'interpréteur BASIC MCS-52; dans ce second et dernier (court) article nous allons nous intéresser à certains problèmes rencontrés lors d'opérations de multiplication.

910128-2-11

Figure 1. Ordinogramme de la routine chargée de la multiplication de 2 nombres.

Il est relativement facile de déceler l'existence d'un certain "flou artistique" lorsque le MCS BASIC-52 d'Intel procède à la multiplication de 2 nombres. Il suffit d'exécuter les 3 embryons de programmes données ci-dessous pour constater que chacun de ces programmes fournit une réponse fausse.

```
10  a=1.E-65
20  b=1.E-65
30  ?a*b        (résultat: 1.0E+126)

10  a=1.E-65
20  b=1.E-64
30  ?a*b        (résultat: 0)

10  a=1.E-64
20  b=1.E-64
30  ?a*b        (résultat: 1.0E-0)
```

Dans chacun de ces 3 cas, l'interpréteur aurait dû afficher le message suivant:

ERROR: ARITH. UNDERFLOW – IN LINE 30

Les exemples ci-dessus mettent l'accent sur des problèmes apparaissant dans les cas-limites de multiplication de 2 nombres.

L'examen critique du mode de travail de l'interpréteur BASIC d'Intel nous a permis de découvrir un algorithme de multiplication à la structure du type de celle illustrée en **figure 1**.

Il semble apparemment que les erreurs découvertes lors de l'exécution des opérations indiquées plus haut, soient dues à la routine de choix de l'exposant, routine dont on retrouve le listage en **figure 2**.

Il est malheureusement impossible de corriger en code machine BASIC-52 les erreurs découvertes dans cette routine sachant qu'une telle correction entraînerait une adjonction de code machine. Cela signifie qu'il faudrait ajouter un

```
1A9AH 9175      ACALL  1C75H
1A9CH BE0002    CJNE   R6,#00H,1AA1H
1A9FH 61B8      AJMP   1BB8H
1AA1H 8D2F      MOV    2FH,R5
1AA3H EF        MOV    A,R7
1AA4H 60F9      JZ     1A9FH
1AA6H 2E        ADD    A,R6
1AA7H 20E705    JB     ACC.7,1AAFH
1AAAH 10D706    JBC    CY,1AB3H
1AADH 61B2      AJMP   1BB2H
1AAFH 5002      JNC    1AB3H
1AB1H 61A1      AJMP   1BA1H
1AB3H 9481      SUBB   A,#81H
1AB5H FE        MOV    R6,A
1AB6H 718B      ACALL  1B8BH
1AB8H 7B04      MOV    R3,#04H
1ABAH AC01      MOV    R4,01H
1ABCH 8C01      MOV    01H,R4
1ABEH E3        MOVX   A,@R1
1ABFH FA        MOV    R2,A
1B30H 7834      MOV    R0,#34H
1B32H E6        MOV    A,@R0
1B33H FE        MOV    R6,A
1B34H 6003      JZ     1B39H
1B36H 717F      ACALL  1B7FH
1B38H 18        DEC    R0
1B39H 08        INC    R0
1B3AH 7408      MOV    A,#08H
1B3CH F9        MOV    R1,A
1B3DH 28        ADD    A,R0
1B3EH F8        MOV    R0,A
1B3FH B60500    CJNE   @R0,#05H,1B42H
1B42H 4013      JC     1B57H
1B44H D3        SETB   C
1B45H E4        CLR    A
1B46H 18        DEC    R0
1B47H 36        ADDC   A,@R0
1B48H D4        DA     A
1B49H D6        XCHD   A,@R0
1B4AH 30E409    JNB    ACC.4,1B56H
1B4DH D9F5      DJNZ   R1,1B44H
1B4FH 18        DEC    R0
1B50H 7601      MOV    @R0,#01H
1B52H 717F      ACALL  1B7FH
1B54H 8006      SJMP   1B5CH
1B56H 19        DEC    R1
1B57H E9        MOV    A,R1
1B58H C3        CLR    C
1B59H C8        XCH    A,R0
1B5AH 98        SUBB   A,R0
1B5BH F8        MOV    R0,A
1B5CH 792B      MOV    R1,#2BH
1B5EH E6        MOV    A,@R0
1B5FH C4        SWAP   A
1B60H 08        INC    R0
1B61H D6        XCHD   A,@R0
1B62H 4206      ORL    06H,A
1B64H F7        MOV    @R1,A
1B65H 08        INC    R0
1B66H 09        INC    R1
1B67H B92FF4    CJNE   R1,#2FH,1B5EH
1B6AH EE        MOV    A,R6
1B6BH 7003      JNZ    1B70H
1B6DH 753000    MOV    30H,#00H
1B7FH 0530      INC    30H
1B81H E530      MOV    A,30H
1B83H 70F9      JNZ    1B7EH
1B85H D0E0      POP    ACC
1B87H D0E0      POP    ACC
1B89H 61A1      AJMP   1BA1H
```

910128-2-12

Figure 2. Listage original de la partie du programme concernant l'ajustage de l'exposant.

```
        ORG     1A9AH

        ; TOS    - Top Of Arithmetic Stack
        ; NXTOS  - NeXt TOS (position behind)

        ; TOS_MUL1
        ; routine for multiplication of two FP numbers, one of which
        ; is located on TOS and the other on NXTOS (NXTOS*TOS)

TOS_MUL1:ACALL  PREP_MUL; clearing of FP working space and reg. prep.
         CJNE   R6,#0,NXTNM_0   ; is NXTOS equal 0?
TOSM_0:  AJMP   ZERO_MANTISSA   ; in case TOS or NXTOS are 0
NXTNM_0: MOV    SIGN,R5         ; result mark in SIGN
         MOV    A,R7    ; is TOS equal 0?
         JZ     TOSM_0
         ADD    A,R6    ; addition of exponent degree (exp. multipl.)
         JB     ACC.7,CMPM_EXP  ; exp. with different mark or overflow?
         JBC    CY,CORM_EXP     ; result is bigger than 0.1
         AJMP   UNDERFLOW       ; in this case exp. sum is < -127
CMPM_EXP:JNC    CORM_EXP; in this case result is smaller than 0.1
OVER_MD: AJMP   OVERFLOW; in this case exp. sum is > 127
CORM_EXP:CLR    MUL_LIMIT_CASE  ; flag of multiplication limit case
         SUBB   A,#82H  ; exp. multipl. results are within the limits
         INC    A       ;  deduction  with  82H  because  of  reduced
         JNC    NMARK_L ;  conditions
         SETB   MUL_LIMIT_CASE  ; limit case
NMARK_L: MOV    R6,A    ; exp. adjustment to real value
         ACALL  BCD2_1  ; disassembl. of TOS mantissa in LEN_MANTISSA
         MOV    R3,#LEN_BYTE    ; acc. (one number in each byte)
         MOV    R4,AR1  ; R1 is pointer of NXTOS
MUL_NOOV:MOV    AR1,R4  ; @R1 contains few numbers and disassembled
         MOVX   A,@R1   ; mantissa is successively multiplied by each
         MOV    R2,A    ; of  them  in  argument  accumulator  (R2  is
                        ; auxilliary register)

-----------------------------------------

         ; TRANSFER
         ; routine  for  adjustment  of  multiplication  and  division
         ; results of  FP  numbers  in  argument  stack  and  mantissa
         ; conversion in BCD packed  format  in  argument  accumulator

TRANSFER:MOV    R0,#ASCII_DEC   ; pointer to MSB result number
         MOV    A,@R0   ; MSB result number is in acc.
         MOV    R6,A
         JZ     MSB_EQ_0; is MSB result number equal 0?
         ACALL  INC_EXP ; exp=exp+1 because from the start it was
         DEC    R0      ; taken that MSB result number is equal 0
MSB_EQ_0:INC    R0      ; pointer to MSB-1 number
         MOV    A,#LEN_MANTISSA ; positioning to LSB-1 number
         MOV    R1,A    ; R1 is counter
         ADD    A,R0
         MOV    R0,A    ; R0 contains the address of LSB-1 number
         CJNE   @R0,#05,ROUND1  ; test of remainder to rounding
ROUND1:  JC     NO_ROUND; in case remainder is smaller than 5 there
C_BCD:   SETB   C       ; is no rounding
         CLR    A       ; carry takes one because of rounding
         DEC    R0      ; transfer of carry into higher byte
         ADDC   A,@R0   ; addition of 1 to higher byte
         DA     A       ; adjustment to BCD format
         XCHD   A,@R0   ; result storing in higher byte
         JNB    ACC.4,NO_C_BCD  ; transfer in higher BCD nibble?
         DJNZ   R1,C_BCD; is @R0 the address of MSB number?
         DEC    R0      ; transfer memorizing in the place MSB+1
         MOV    @R0,#01
         ACALL  INC_EXP ; transfer is outside decimal point frame
         SJMP   BCD1_2
NO_C_BCD:DEC    R1
NO_ROUND:MOV    A,R1    ; address return in R0 to the first bigger BCD
         CLR    C       ; number which is not equal 0
         XCH    A,R0
         SUBB   A,R0
         MOV    R0,A
BCD1_2:  MOV    A,EXPONENT      ; test of exceeding in limit case
         JNB    MUL_LIMIT_CASE,N_LCASE
         JZ     UNDER_MD; message about underflow
         INC    A
         JZ     UNDER_MD; message about underflow
N_LCASE: MOV    R1,#ARGUMENT_ACC; repacking of BCD number in one byte
XFER_1:  MOV    A,@R0   ; in two BCD numbers in one byte
         SWAP   A       ; higher number in BCD packed  format is in
         INC    R0      ; upper nibble
         XCHD   A,@R0   ; BCD packed format is in ACC
         ORL    AR6,A   ; the indication if mantissa is 0 is in R6
         MOV    @R1,A   ; packing in argument accumulator mantissa
         INC    R0      ; two BCD packed numbers in one byte
         INC    R1
         CJNE   R1,#SIGN,XFER_1
         MOV    A,R6
         JNZ    RESULT_MAT_OP_TOS       ; is mantissa equal 0?
         MOV    EXPONENT,#0

-----------------------------------------

INC_EXP: INC    EXPONENT; exp. adjustment by 1 (consequence of
         MOV    A,EXPONENT      ; transfer after addition)
         JNZ    EXP_OK  ; test of exponent exceeding
         POP    ACC     ; removal from return address stack
         POP    ACC
         JBC    MUL_LIMIT_CASE,UNDER_MD ; in limit case it is
         AJMP   OVERFLOW; underflow
UNDER_MD:AJMP   UNDERFLOW
                                                    910128-2-13
```

Figure 3. Code-source de la routine de multiplication améliorée. Les lignes verticales identifient les lignes corrigeant les erreurs de multiplication de cas limite mentionnées dans cet article.

certain nombre de lignes au fichier source de l'assembleur pour introduire cette compensation.

Il apparaît un problème lorsque le résultat de l'instruction SUBB A,#81H est 0FFH, ce qui correspond à un exposant de E+127. Si la partie de programme de multiplication de mantisse comporte un résultat commençant par un 0 après la virgule, l'exposant n'est pas incrémenté restant à 0FFH. Après la mise en forme du résultat (voir figure 1) et la routine de sortie sérielle (affichage), on a un exposant de E+126.

Nous avons développé, pour éviter les problèmes et les erreurs lors de multiplication de cas limite, le programme listé en **figure 3**. Les lignes identifiées par les barres verticales, en particulier, corrigent les erreurs de multiplication de cas-limite. Il existe bien entendu d'autres approches possibles pour ce type de manipulation. Nous sommes persuadés que les professionnels du 8052/8051 qui se rencontrent sur leur serveur dédié pourraient nous apprendre beaucoup de choses.

Sur la base des informations publiées dans cet article et celui du mois dernier nous avons donc développé, pour le 8051, une arithmétique de virgule flottante (FP = *Floating Point*) améliorée, ne comportant pas d'erreur, en nous basant sur le noyau de virgule flottante du 8052AH-BASIC. Ce nouveau noyau FP utilise une technique modifiée d'accès à la pile arithmétique et possède un code plus court et plus rapide pour un certain nombre d'algorithmes. Il permet de choisir entre 2 et 16 chiffres la longueur de la mantisse d'un nombre à virgule flottante, sachant que la cartographie complète des variables de l'arithmétique de virgule flottante se trouve dans l'espace mémoire interne du microcontrôleur 8051.

Le mois prochain:

- un module à afficheurs 7 segments à LED pour le bus I²C,
- UNIVOLT, module d'alimentation compact à tension de sortie définissable par l'utilisateur,
- un L/C-mètre à compensation automatique de résistance de perte,
- un système mono-carte «mini-Z80»,
- un générateur de test pour les Hautes-Fréquences,
- de l'utilisation des condensateurs en audio, voici quelques-uns des articles que vous pouvez vous attendre à trouver dans le numéro de mars 1992.

émulateur de 8751

un système de développement complet: matériel + logiciel

O. Bailleux

Si vous êtes un lecteur assidu de ce mensuel consacré à l'électronique sous toutes ses formes et que vous vous intéressiez de près à tout ce qui touche à la micro-informatique vous aurez sans doute noté la publication dans Elektor, à intervalles plus ou moins réguliers, de simulateurs et autres émulateurs de tout genre.
Nous vous avons proposé, par le passé, RAMSAS, un simulateur d'EPROM universel (n°99), un simulateur d'EPROM (de la 2764 à la 27512, n°137, novembre 1989), un émulateur de 2764 (n°157/158) et bien d'autres réalisations dans ce domaine fort intéressant qu'est celui du développement de «firmware» c'est-à-dire la création de programmes en (EP)ROM, où que se trouve cette dernière.

Mais quelle est donc l'utilité de tous ces émulateurs et, en particulier, celle d'un **émulateur de 8751**, ne manqueront pas de se demander un certain nombre de nos lecteurs.
Un émulateur de XXXX est un outil de développement qui se substitue à un composant du type XXXX qu'il simule au cours d'opérations complexes de programmation permettant ainsi de faciliter la mise au point de programmes de gestion de systèmes électroniques à base de microprocesseurs ou de microcontrôleurs. Il comporte une sonde, le plus souvent un connecteur DIL au nombre de broches convenable, qui vient se connecter directement sur l'application, à la place du microcontrôleur dans le cas présent. Ce processus peut également être utilisé avec des EPROM, des microprocesseurs et, comme ici, des microcontrôleurs de tous types et de toutes origines.

Depuis l'ordinateur d'exploitation, en l'occurrence un compatible PC, auquel est relié l'émulateur, on peut:
– Modifier rapidement, et sans la moindre action mécanique –extraction, effacement, reprogrammation et remise en place du composant– car c'est bien là que le bât blesse, le programme en cours de mise au point, écrit en assembleur symbolique,
– Placer dans le programme des points d'arrêt,
– Visualiser le contenu des registres et de la mémoire,
– Exécuter le programme instruction par instruction, en mode «pas à pas»,
– Modifier le contenu de certains registres.

On aura compris à la lecture de ces quelques lignes, qu'un émulateur n'est pas un gadget, mieux encore, on peut dire qu'il s'avère indispensable dès lors que l'on conçoit des programmes assez longs (1 Koctet et plus), en particulier sur des systèmes à EPROM ou à microcontrôleurs «monochip».

Généralités

Une remarque préliminaire importante: il nous paraît plus que judicieux de ne pas se lancer dans un tel projet si l'on ne dispose pas de la moindre documentation concernant le composant que l'on envisage d'émuler. Il vaut donc la peine d'essayer de mettre la main sur le Data Book du 8751 (*Microcontroller Handbook* d'Intel par exemple), car il nous est impossible ici d'entrer dans le détail de la description et des possibilités du 8751, de ses instructions etc... encore que le fichier LISEZ.MOI présent sur la disquette de logiciel accompagnant ce montage vous propose un certain nombre des informations les plus importantes.
Résumons:

Le 8751 est un microcontrôleur qui intègre:
– un microprocesseur à 8 bits,
– une RAM de 128 octets,
– 4 ports d'E/S parallèles à 8 bits,
– 1 port de communication série,
– 2 temporisateurs (*timers*) programmables,
– 2 interruptions extérieures.

L'espace d'adressage 8 bits de

256 octets du 8751 est divisé en 2 parties:
– la RAM Utilisateur: adresses 00_{HEX} à $7F_{HEX}$ qui contient notamment les registres R0 à R7 (adresses 00_{HEX} à 07_{HEX}) et la pile-système. Une zone de 16 octets est adressable par bit. La zone «Registres de contrôle»: adresses 80_{HEX} à $0FF_{HEX}$. Les ports d'E/S, les registres de contrôle des timers et autres ressources se trouvent dans cette zone. Cette zone est accessible exclusivement avec le mode d'adressage direct.

Le 8751 ne connaît pas moins de 256 instructions différentes obtenues par la combinaison aux instructions de base de différents opérandes et de modes d'adressage divers. En un mot comme en cent, on se trouve ici en présence d'un processeur performant et donc (relativement) complexe.

Caractéristiques techniques:
- Émulation en temps réel du 8751
- Fréquence d'horloge fixe: 8 MHz
- Tous les ports d'E/S sont disponibles
- Toutes les interruptions et ressources internes (timers, port série...) sont disponibles
- Alimentation à partir de la carte en cours de développement, la consommation n'étant que de 100 mA sous 5 V
- Commande (contrôle) à partir d'un ordinateur compatible PC (liaison série 9600 bauds)
- Points d'arrêt - Exécution pas à pas
- Visualisation du contenu des registres et de la RAM interne
- Modification possible de la valeur de certains registres et de la RAM interne
- Assembleur symbolique 8051
- Éditeur pleine page avec localisation des erreurs
- Production de fichiers binaires ou de format HEX (format INTEL)
- Impression formatée du programme source.

Limitations:
- Mode "monochip" exclusivement (mémoire programme interne, pas de mémoire données externe)
- Bit d'E/S P3.6 et P3.7 strictement réservés au système
- Bancs de registre 0 exclusivement
- Utilisation de 12 octets sur la pile par le système
- Interruptions masquées pendant le traitement des points d'arrêt.

Le principe de fonctionnement

L'électronique
Comme l'illustre le synoptique de la figure 1, un système à microprocesseur est, en règle générale, constitué d'un microprocesseur, d'une mémoire programme, d'une mémoire de données et de périphériques d'E/S, ces divers éléments étant interconnectés grâce à un bus d'adresses, un bus de données et un bus de commande – souvent appelé à tort bus de contrôle de par sa dénomination anglaise de «*control bus*».
Lorsque le 8751 travaille en mode "monochip", tous ces éléments sont à l'intérieur du composant et seuls les ports d'E/S sont accessibles depuis l'extérieur (4 ports de 8 bits).

Pour émuler un 8751 dans ce mode de fonctionnement, il faut donc utiliser un composant réunissant les caractéristiques suivantes:
– Possibilité de placer en RAM le code exécutable (et non en EPROM interne) de manière à pouvoir le modifier rapidement,
– 4 ports d'E/S et des périphériques identiques à ceux du 8751,
– un jeu d'instructions compatible avec celui du 8751.

Résultat: le composant répondant à ce cahier des charges et donc retenu ici a été le 80C451.
Nous vous proposons, à titre d'information, le brochage du SC80C451 dans la marge.
L'émulateur bâti autour de ce composant et comprend:
– Une EPROM contenant le programme de lancement,
– Une RAM de 32 Koctets contenant les routines système et le programme utilisateur en cours de mise au point.
– Un dispositif de commutation de mémoire comportant 2 états:
- État de démarrage: l'EPROM est implantée à l'adresse 0000_{HEX} et la RAM à l'adresse 8000_{HEX}.
- État d'exécution: la RAM est en 0000_{HEX} et l'EPROM en 8000_{HEX}.
– Une interface TTL/RS232 pour la connexion au port série d'un PC.
La procédure d'initialisation est la suivante:

1) Initialisation (*Reset*) et passage en état démarrage. Exécution du programme en EPROM à l'adresse 0000_{HEX}.

2) Le programme en EPROM assure le chargement du programme système en RAM grâce à un protocole de communication simplifié avec le PC de gestion.

3) Le programme en EPROM actionne un bit d'E/S (réservé) qui provoque le passage en état d'exécution et un nouveau Reset.

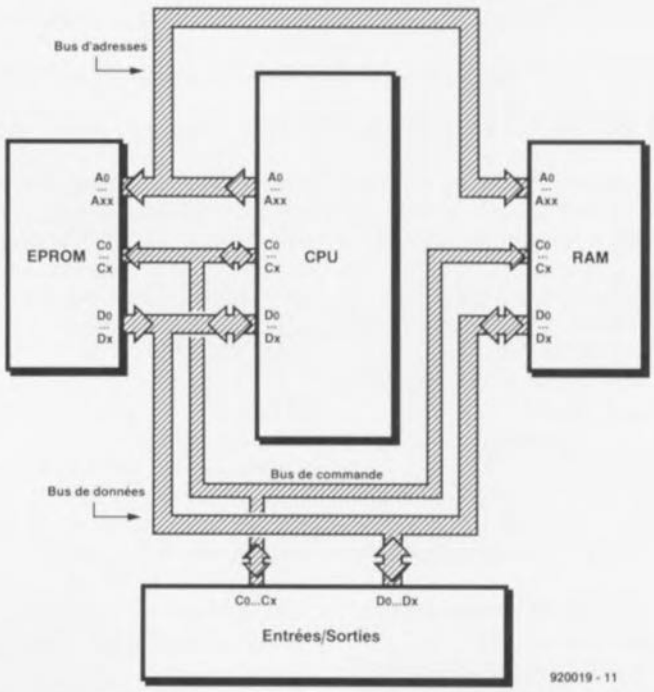


Figure 1. Synoptique du système à microprocesseur le plus rudimentaire. Un certain nombre de microcontrôleurs intègre pratiquement la quasi-totalité de cette circuiterie dans un unique composant.

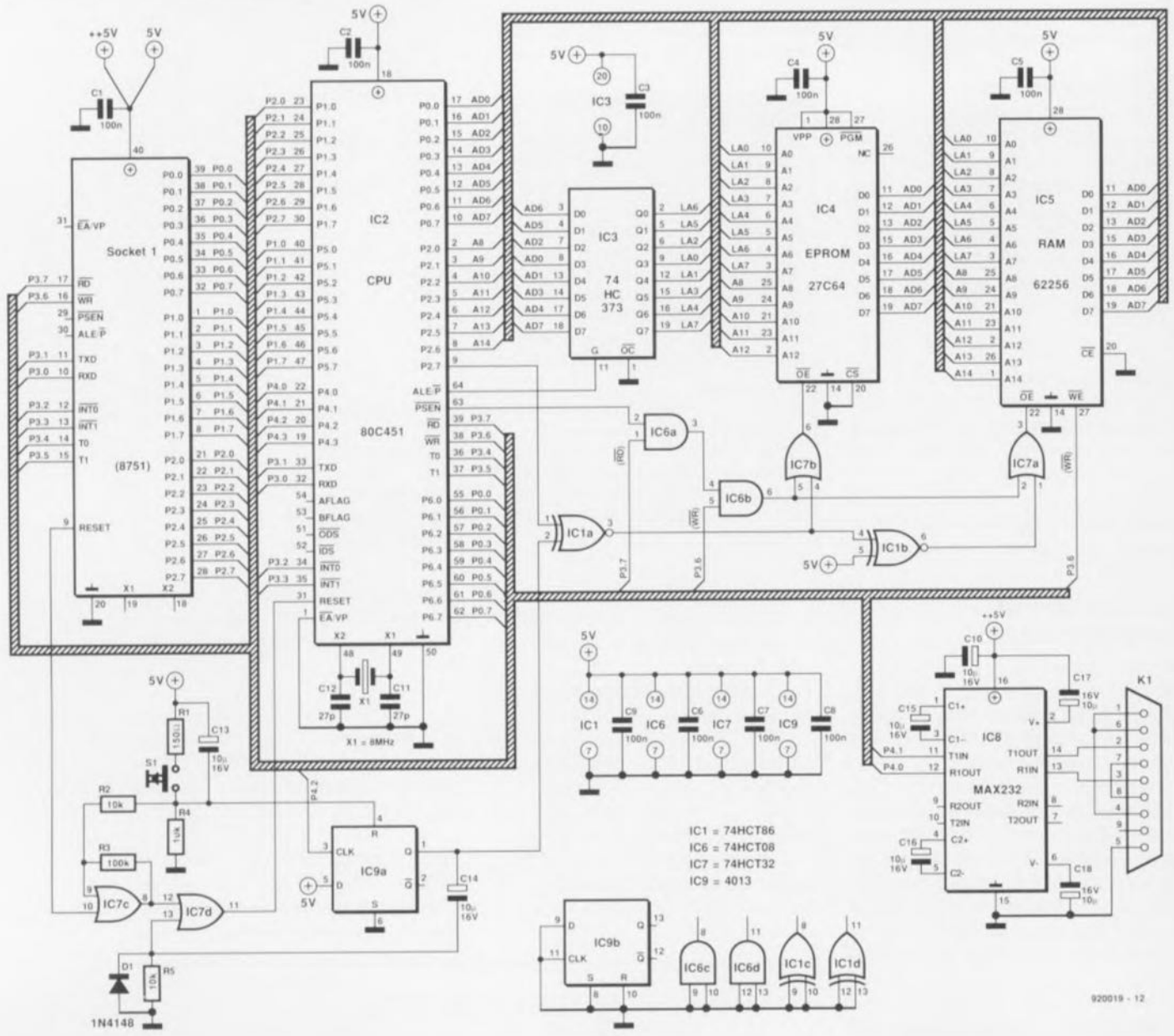


Figure 2. Schéma de l'émulateur de 8751. Un régal pour les amateurs. Il reste un certain nombre de portes inutilisées.

4) Le programme système s'exécute à l'adresse 0000_{HEX}, il se duplique et se réexécute à la fin de la RAM en 7800_{HEX}. De là, il assure la communication avec le PC grâce à une routine de télécommande qui permet, depuis le PC:

– La réinitialisation logicielle du microcontrôleur,

– La lecture du contenu de la RAM de l'émulateur,

– La modification du contenu de la RAM de l'émulateur,

– La modification du pointeur de pile,

– Le chargement d'un nouveau programme utilisateur,

– Le lancement ou la poursuite de l'exécution du programme utilisateur.

5) Depuis le PC, le programme de gestion assure le contrôle à distance de l'émulateur. Dans un premier temps, il va transférer le programme utilisateur à tester. Ce programme comporte bien sûr au moins un point d'arrêt, c'est-à-dire un appel à une routine système de gestion de point d'arrêt.

6) Le programme utilisateur s'exécute jusqu'à ce qu'un point d'arrêt soit rencontré. La routine de gestion de point d'arrêt transfert les valeurs de registres et de la RAM interne du 8751 dans une zone tampon de la RAM externe de l'émulateur, en 7000_{HEX}, puis appelle la routine de télécommande.

7) Le programme de gestion intervient à sa guise dans la RAM externe de l'émulateur puis demande la poursuite de l'exécution du programme utilisateur – jusqu'au prochain point d'arrêt. À ce stade, il est possible de provoquer depuis le PC la réinitialisation du 8751 ou le chargement et le lancement d'un nouveau programme utilisateur.

8) Les valeurs de la zone tampon en 7000_{HEX} – qui ont pu être modifiées par le programme de gestion – sont rechargées dans la RAM interne et les registres du 8751 et la routine de traitement de point d'arrêt se termine. Le programme utilisateur reprend son exécution jusqu'au prochain point d'arrêt.

La routine de télécommande est la clé du système. Elle permet au programme de gestion de modifier le programme utilisateur et le contenu des registres du 8751. Cette fonctionnalité permet d'installer des

points d'arrêt dynamiques (temporaires) utilisés notamment dans le mode d'exécution pas à pas.

Le logiciel

prend la forme d'une disquette 5''¼ de 360 Koctets pour IBM et comporte les fichiers suivants:

DEV.EXE Environnement de programmation
CONFIG.EMU Fichier de configuration
SYS8751.C51 Fichier d'initialisation
LISEZ.MOI Informations additionnelles importantes
plus 2 ou 3 autres fichiers.

2 des caractéristiques les plus intéressantes de ce système sont, comme mentionné quelques lignes plus haut,:
– La possibilité de définir des points d'arrêt dynamique et
– L'existence d'un mode pas à pas.

Le fichier LISEZ-MOI vous donnera de plus amples informations à ce sujet. Soyons brefs!
Par point d'arrêt dynamique on entend point d'arrêt placé par l'utilisateur ou le programme de gestion dans le programme utilisateur pendant son exécution, au moment où un point d'arrêt (statique ou dynamique) est en cours de traitement. Rappelons que l'utilisateur ou le programme de gestion ne peuvent agir sur l'émulateur que pendant le traitement d'un point d'arrêt dont l'utilisateur peut définir la position dans le programme source. Le programme de gestion calcule l'adresse correspondante, sauvegarde dans une variable les 3 octets de code qui se trouvent à cette adresse et met à la place les 3 octets servant à coder l'instruction d'appel de la routine de point d'arrêt. Après différentes manipulations on se retrouve dans les conditions de départ et l'on peut passer au point d'arrêt suivant.

En mode pas à pas, le programme de gestion de l'émulateur place, pendant le traitement d'un point d'arrêt, un point d'arrêt dynamique juste après l'instruction suivante, puis redonne la main au programme utilisateur qui va donc exécuter une instruction et ''tomber'' sur ce nouveau point d'arrêt. Ce cycle peut être répété indéfiniment.

L'électronique

Même le lecteur le plus distrait ne manquera pas d'identifier le coeur de ce montage dont nous vous proposons le schéma en **figure 2**. Ce composant est très rarement utilisé, et ce en dépit de ses nombreuses caractéristiques fort intéressantes. Peut-être que nous aurons aidé à mieux le faire connaître.
Le SC80C451, puisque c'est de lui qu'il s'agit, est un microcontrôleur ayant pour origine Intel, mais fabriqué par Signetics, la branche américaine de Philips.
Nous vous en proposons la structure interne en **figure 3**.
À y regarder de plus près, il s'agit en fait d'une version étendue du 80C51, un microcontrôleur de la famille 80C31/51, ne comportant pas moins de 3 ports d'E/S additionnels, ce qui lui en fait 6, et 4 lignes de commande d'E/S supplémentaires.

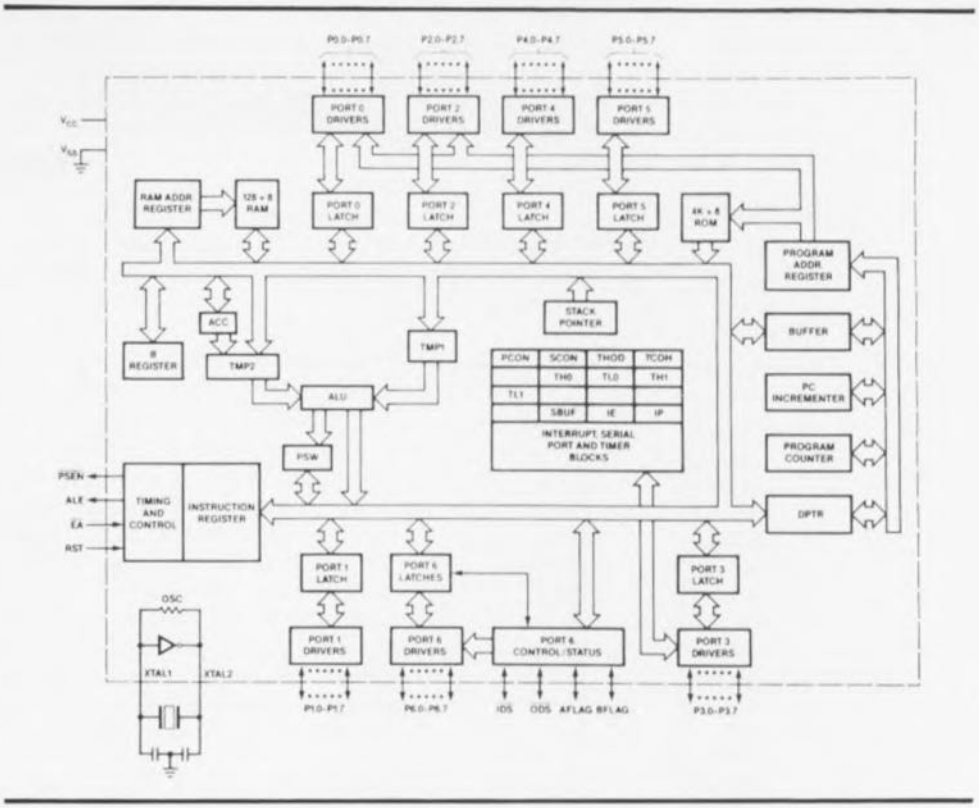


Figure 3. Structure interne du 80C451. On notera la présence d'un nombre important de ports.

Le 80C451 ne comporte pas de ROM interne; il va donc chercher ses instructions dans une (EP)ROM externe. Il comporte également 128 octets de RAM et la version DIP utilisée ici ne possède pas moins de 6 ports à 8 bits et 1 port à 4 bits.

Notons en passant qu'il existe une version PLCC de ce composant, dotée elle de 68 broches, les 4 broches additionnelles étant autant de lignes de commande d'Entrées/Sorties, associées au port 6, et destinées à faciliter les fonctions d'E/S asynchrones à vitesse élevée. La version PLCC met donc à disposition 7 ports à 8 bits.

Ces 2 types de circuits sont en mesure d'adresser 64 Koctets de ROM et autant de RAM.

La consommation de courant de ce type de circuit est faible: 24 mA sous 5 V à 12 MHz en service, 3 mA au repos et 50 μA en veille (*Power-down mode*).

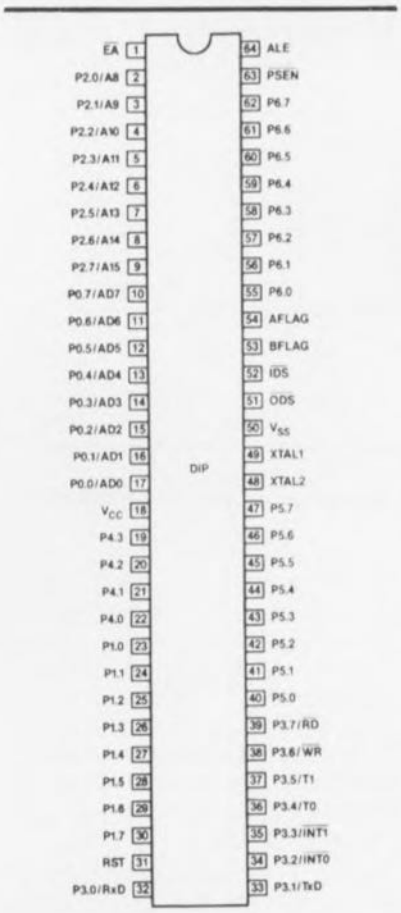
Il existe également une version à ROM programmable par masque de ce circuit, le SC83C451, dotée elle de 4 Koctets de ROM. Les 2 types de circuit possèdent 2 temporisateurs/compteurs à 16 bits.
Leur structure d'interruptions imbriquée connaît 2 niveaux de priorité.
La présence d'un port d'E/S sériel permet ou la réalisation d'un UART (***U**niversal **A**synchronous **R**eceiver/**T**ransmitter* = Émetteur/Récepteur Asynchrone Universel) en duplex vrai (*full duplex*), ou une extension des lignes d'E/S, soit encore l'établissement de communications inter-processeurs; il intègre également sa propre circuiterie d'oscillateur et d'horloge.

L'une des caractéristiques de fonctionnement intéressante de ces 2 types de circuits est l'existence de 2 modes de fonctionnement additionnels sélectables par logiciel et permettant une réduction sensible de la consommation: un mode «ralenti» et un mode «veille». Dans le premier mode la «cépéu» (CPU) est mise en hibernation, le reste du circuit, à savoir la RAM, les temporisateurs, le port sériel et le système

d'interruption, continuant lui de fonctionner. En mode «veille» l'oscillateur est à son tour «gelé», ce qui rend inopérantes le reste des fonctions, le contenu de la RAM restant lui sauvegardé.

La réalisation

Après cette théorie relativement sèche, nous en arrivons à l'étape intéressante de ce montage: sa réalisation.



Brochage du 80C451 (Source Philips).

L'existence d'une platine double face à trous métallisés dessinée à l'intention de ce montage et dont on retrouve la représentation de la sérigraphie de l'implantation des composants en figure 4, transforme la réalisation de ce montage en jeu d'enfant (et le pire, dans tout ceci..., c'est que nous sommes persuadés de ce que nous écrivons là).

Un certain nombre des composants passifs, résistances et condensateurs sont montés verticalement.
Il suffira de bien faire attention au sens d'implantation des différents circuits intégrés, sachant qu'ils ne sont pas tous orientés de la même façon, à la polarisation des condensateurs et de la diode.
Le bouton-poussoir à contact travail S1 pourra être positionné sur le boîtier d'où il sera relié aux picots correspondants présents sur la platine.

Le SC80C451 mérite bien évidemment d'être placé sur un support constitué, par exemple, par la mise en série de 32 contacts tulipe fournis par des barrettes autosécables. La RAM et l'EPROM méritent sans doute elles aussi un traitement de faveur.
La solution la plus simple consiste à doter le connecteur SOCKET1 d'un support à 40 broches dans lequel viendra s'enficher un connecteur DIL terminant un morceau de câble plat à 40 conducteurs, câble dont l'autre extrémité est elle aussi dotée d'un connecteur DIL destiné à venir s'enficher à l'emplacement prévu pour le 8751 sur le système en cours de développement. Attention, ce type de connecteur est très fragile, on a vite fait d'en replier une patte avec un risque non négligeable de la briser lors de son redressement.
Il ne reste plus, pour terminer cette réalisation, qu'à implanter l'**émulateur de 8751** dans un boîtier de caractéristiques convenables doté d'une fente destinée au passage du câble d'interconnexion.

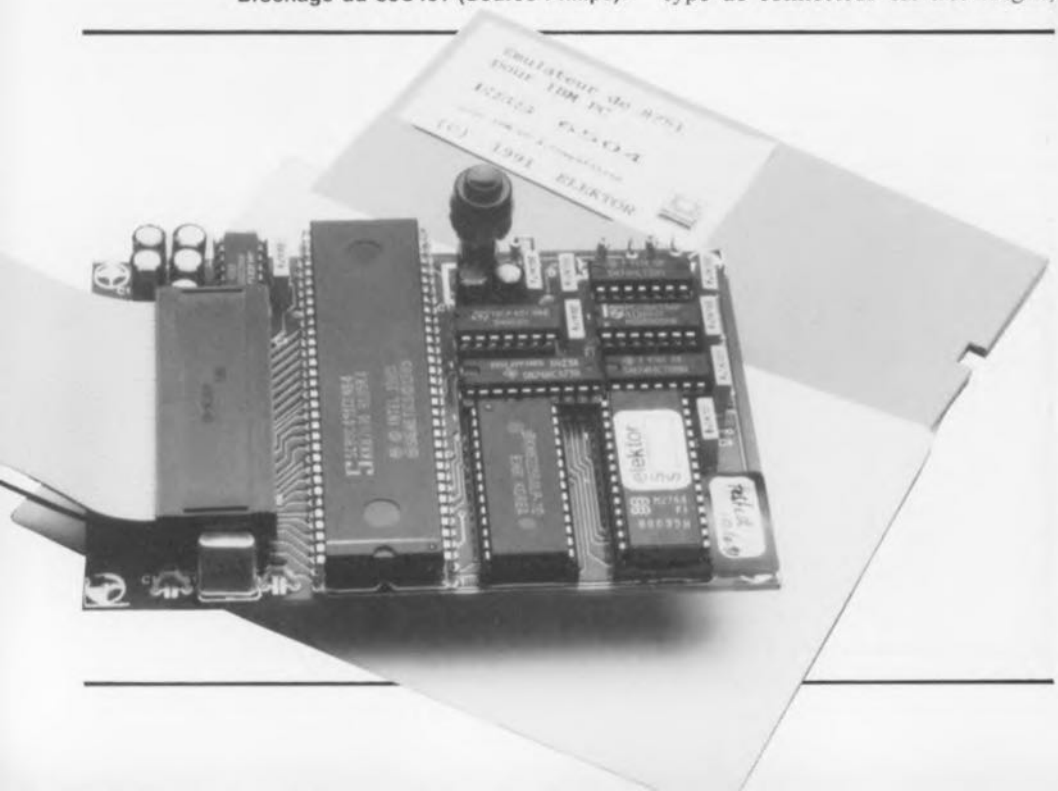
Mode d'emploi

Mise en service de l'émulateur

On commencera par s'assurer que le PC et l'application ne sont pas alimentés. Il n'y a pas de risque de ce côté-là en ce qui concerne l'émulateur puisqu'il tire, normalement, son alimentation de l'application.
On relie l'émulateur au PC par l'intermédiaire du câble RS232. On pourra utiliser les ports COM1 ou COM2.
On adapte le cas échéant le fichier de configuration CONFIG.EMU selon les indications données un peu plus loin.
Faire un essai «en l'air» en appliquant une tension de 5 V entre les broches 40 et 20 du support DIL (+5 V à la broche 40, masse aux broches 20 et 9).
Lancer le programme en faisant «**DEV**» et vérifier qu'il n'y a plus affichage du message «Émulateur non connecté ou planté PRESSEZ UNE TOUCHE».

Il doit vous être possible d'émuler le programme d'exemple (voir la syntaxe dans le fichier **LISEZ.MOI**) sans qu'il ne soit nécessaire d'enficher le support dans le système d'application.
Si tout s'est passé comme prévu, on pourra couper l'alimentation pour ensuite enficher la sonde de l'émulateur sur le support convenable de l'application (à la place du futur 8751 donc).
On alimente ensuite l'application avant de mettre le PC en marche.

Si, maintenant, vous envisagez d'essayer l'**émulateur de 8751** sans le connecter à un quelconque système, il faudra bien entendu en assurer l'alimentation par l'intermédiaire du connecteur DIL à 40 broches destiné à finir à la place du 8751 lors de l'émulation. La broche 40 de ce connecteur sera reliée au +5 V de l'alimentation, sa broche 20 l'étant à la masse; pour se mettre à l'abri de tout risque que pourrait entraîner une erreur de polarité lors de la connexion de l'alimentation, on pourra placer envisager l'implantation, entre ces 2 broches, d'une diode (1N4001) dont la cathode sera reliée au plus.



Lancement du logiciel

Comme nous le disions plus haut, le logiciel fonctionne (en partie) même si l'émulateur n'est pas connecté, voire pas alimenté. Dans ce cas on ne dispose bien évidemment que d'une partie des fonctions, à savoir éditer et assembler des programmes. Quoi qu'il en soit, le répertoire de travail de votre système doit contenir les fichiers suivants recopiés de la disquette ESS6054:

DEV.EXE: Environnement de développement
CONFIG.EMU Fichier de configuration
SYS8751.C51: Programme système de l'émulateur.

Le fichier **CONFIG.EMU** contient 2 caractères:

1e caractère : **M** pour **M**onochrome (Hercules)
C pour **C**ouleur (CGA, EGA, VGA)

2e caractère : **1** pour RS232 sur COM**1**
2 pour RS232 sur COM**2**

Ce fichier ne doit contenir aucun autre caractère (ni espace, ni retour à la ligne). Il peut être modifié avec l'éditeur de texte de l'environnement de développement ou tout autre programme permettant une telle manipulation, l'utilitaire PC Tools ou tout traitement de texte ASCII.

Il faudra donc commencer, si nécessaire, par modifier le fichier **CONFIG.EMU** pour qu'il réponde aux caractéristiques de votre système: on aura donc, selon la configuration du système, soit **M1**, soit **M2**, soit **C1**, soit **C2** dans ce fichier. Le fichier **CONFIG.EMU** présent sur la disquette est du type **C1**, c'est-à-dire **C**ouleur avec COM**1** en sortie série.

On peut alors lancer le programme **DEV.EXE** en faisant DEV depuis le répertoire où se trouve ces différents programmes. Le menu principal offre plusieurs choix, accessibles grâce aux touches «flèches» du clavier.

Si maintenant on travaille avec un émulateur de 8751 sous tension, il faudra commencer par l'initialiser par une action sur S1 avant de lancer le programme **DEV.EXE**.

La disquette vous proposant tous les programmes nécessaires, **l'ESS6054** disponible auprès des sources habituelles, comprend en outre un fichier **LISEZ.MOI** qui donne des informations additionnelles concernant l'utilisation du logiciel de gestion. Nous nous limitons donc ici au strict nécessaire.

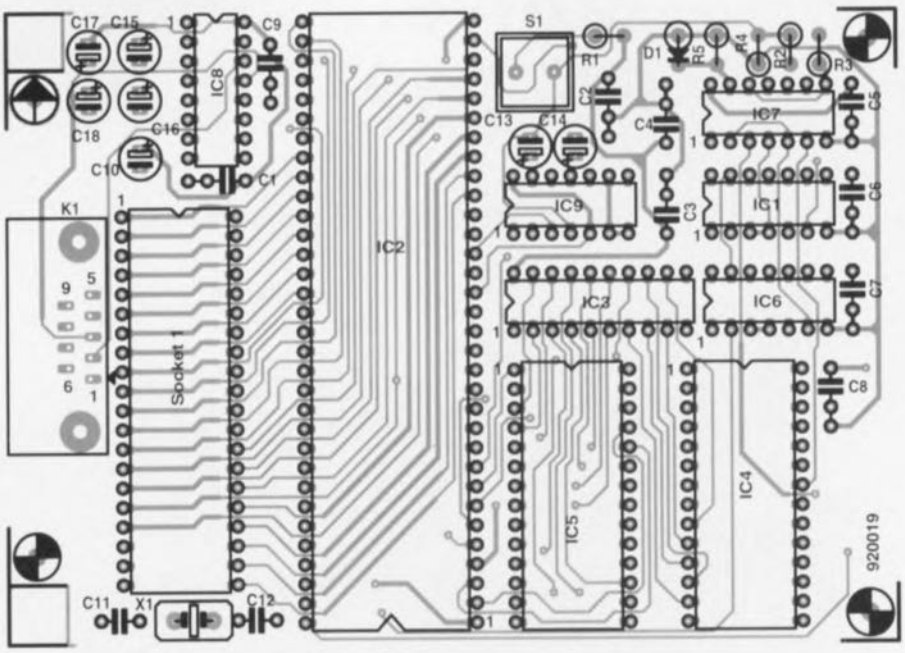


Figure 4. Représentation de la sérigraphie de l'implantation des composants du circuit imprimé double face à trous métallisés dessiné pour cette réalisation.

En cas de problème, émulateur non alimenté par exemple, le programme signale cette situation sans pour autant se «planter». On peut fort bien poursuivre les essais d'un certain nombre des fonctions de cette réalisation.

Le fichier LISEZ.MOI, à lire impérativement, donne toutes les informations concernant le protocole de communication. Pour des raisons de place disponible nous ne pouvons pas ici reprendre ces milliers de caractères de texte.

Dernière remarque: l'auteur, Mr Bailleux, recommande, pour éviter d'éventuels problèmes, de mettre les broches 52, 53 et 54 du 80C451 à la masse, en particulier la broche 52, IDS, sinon, le port 6, émulant le port 0, ne fonctionne pas correctement. Nous n'avons cependant pas constaté de problème de ce côté-là, les broches 53 et 54 étant dotées de résistances de forçage au niveau haut internes.

Vous voici donc équipés d'un système de développement pour 8751 dont le rapport qualité prix est fort intéressant, car ce type de matériel coûte plusieurs dizaines de milliers de francs, comme nous l'ont appris nos demandes de documentation au dernier Salon «COMPONIC '91».

Faites bon usage pour créer des automatismes en tout genre. Nous attendons de vos nouvelles... ■

Liste des composants

Résistances:
R1 = 150 Ω
R2,R4,R5 = 10 kΩ
R3 = 100 kΩ

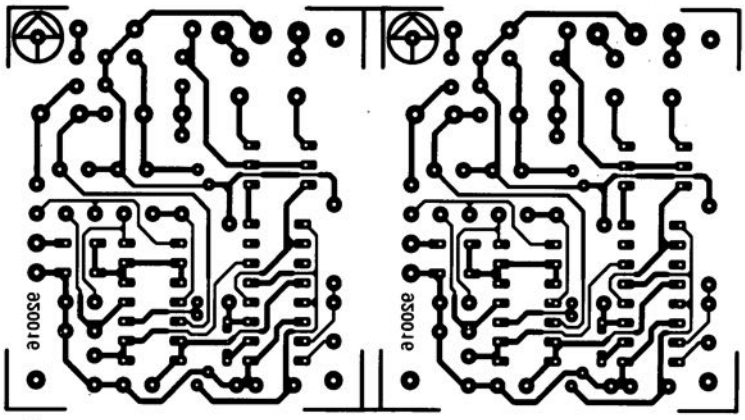
Condensateurs:
C1 à C9 = 100 nF
C10,C13 à C18 = 10 μF/16 V radial
C11,C12 = 27 pF

Semi-conducteurs:
D1 = 1N4148
IC1 = 74HCT86
IC2 = SC80C451CCN64 (Philips, Signetics)
IC3 = 74HC373
IC4 = 27(C)64 programmée (ESS 6054)
IC5 = 62256
IC6 = 74HCT08
IC7 = 74HCT32
IC8 = MAX232 (Maxim)
IC9 = 4013

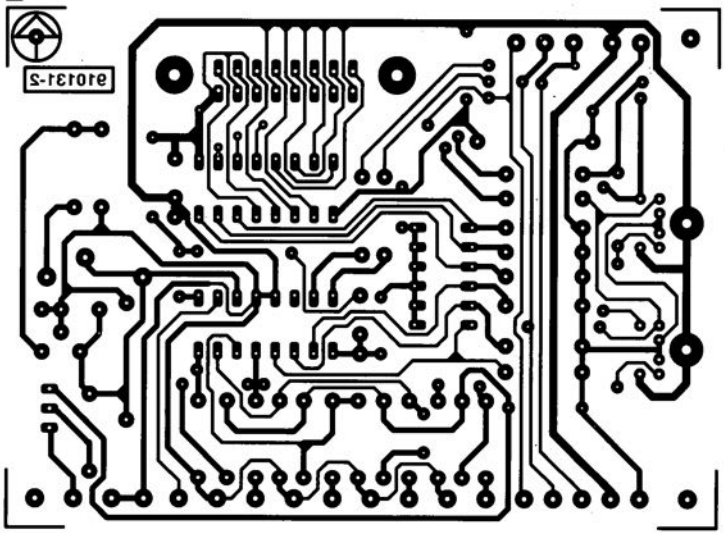
Divers:
K1 = embase Sub-D femelle à 9 contacts en équerre encartable
S1 = bouton-poussoir à contact travail (RAZ)
X1 = quartz 8 MHz
64 contacts sous forme de barrette autosécable (pour IC2)
H8 = support à 40 broches
morceau de câble en nappe à 40 conducteurs doté à ses 2 extrémités d'un connecteur auto-dénudant DIP à 40 contacts
un boîtier tel que, par exemple, Pactec HP kit 92 x 146 x 28 mm (OKW)

1 témoin d'écrêtage
2 convertisseur A/N-N/A pour I²C
3 convertisseur RS-232-A/N

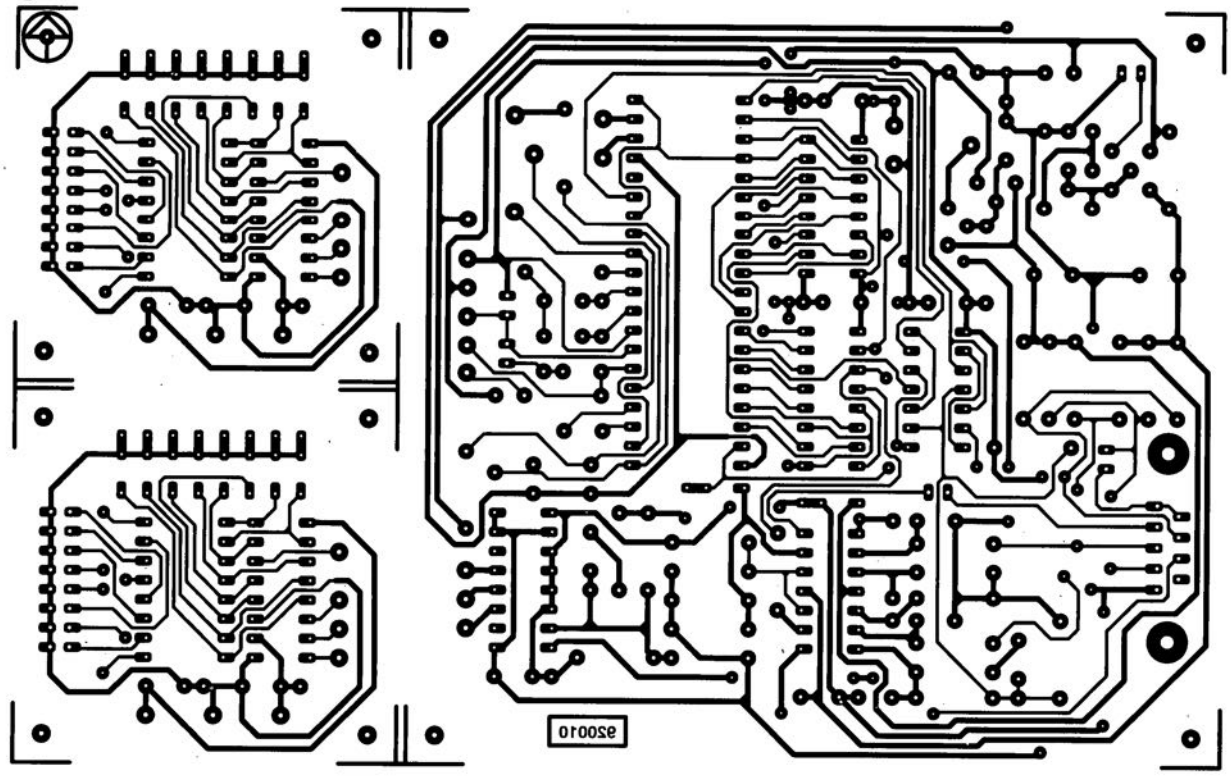
1



2

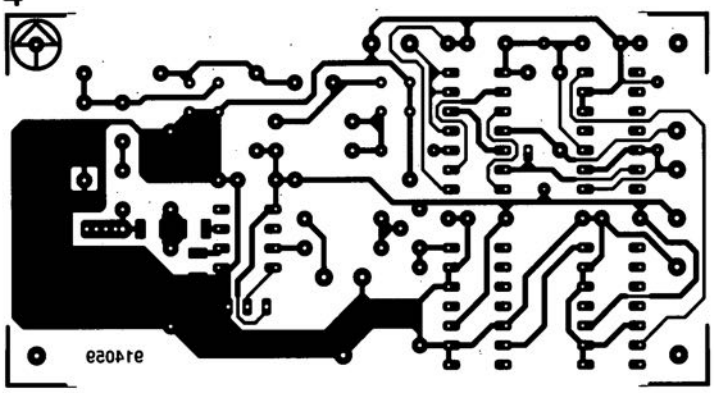


3

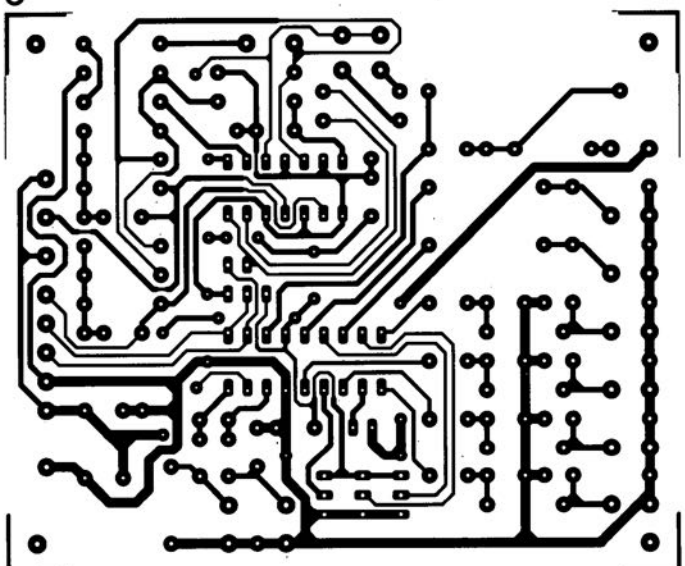


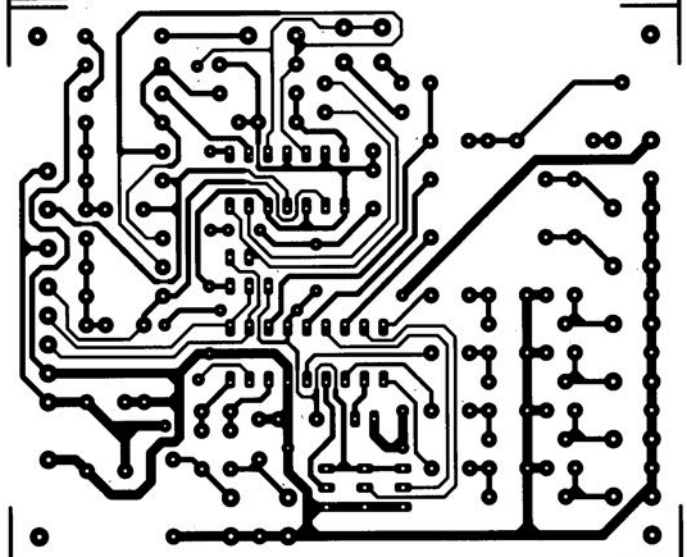
4 prédiviseur 1,3 GHz
5 commutateur audio/vidéo électronique
6 émulateur de 8751 : côté composants
7 émulateur de 8751 : côté pistes

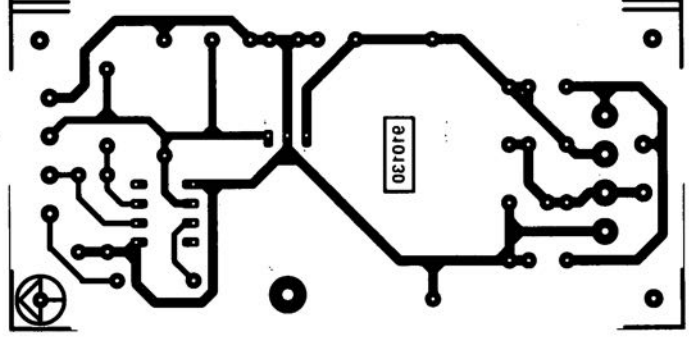
4



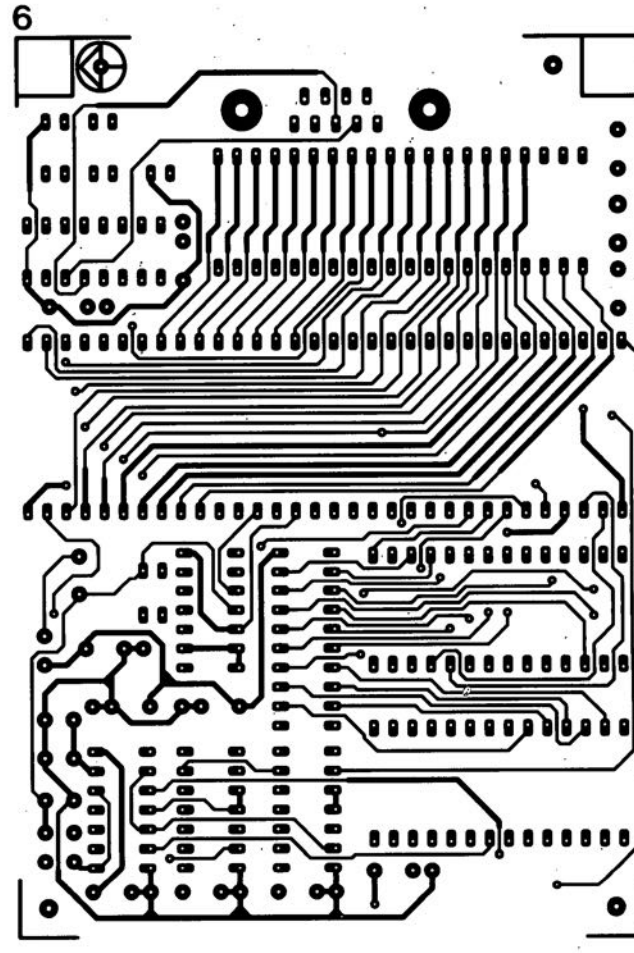
5



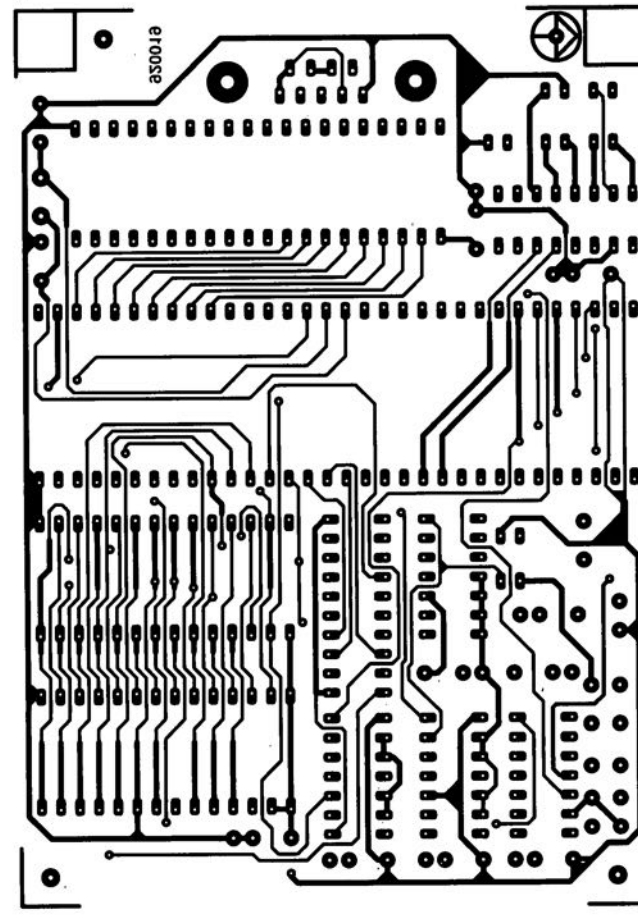




6



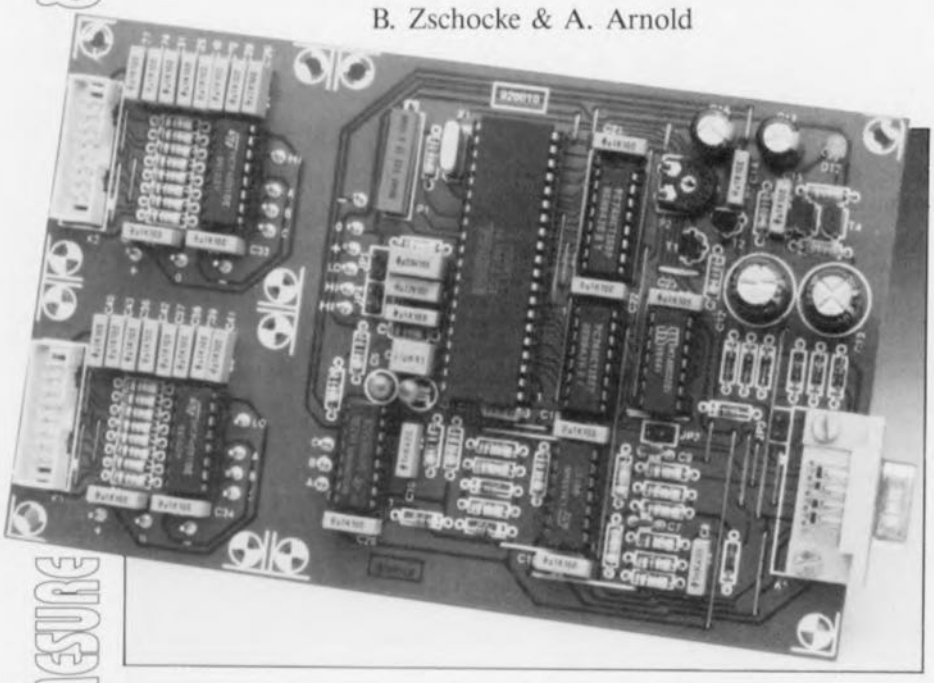
7



convertisseur RS-232 – A/N

7109 à interface pour micro-ordinateur

B. Zschocke & A. Arnold



Est-il encore nécessaire de vous présenter le 7106, un convertisseur que l'on trouve au coeur de plusieurs générations de multimètres. Il existe une version spéciale de ce circuit intégré, à la précision identique et au même coût abordable, mais doté d'une interface parallèle à la place de la circuiterie de commande d'afficheurs à 7 segments. Il ne faut lui ajouter que fort peu de composants additionnels pour en faire une «boîte noire» de mesure à interface RS-232 qu'il est extrêmement aisé de commander par logiciel grâce à l'utilisation de quelques astuces techniques.
Nous pouvons proposer aux possesseurs d'ordinateurs du type IBM-PC & Compatibles (et pour ces types seulement malheureusement) un logiciel permettant la commande de ce convertisseur RS-232 – A/N.

Rassurez-vous, nous n'avons pas fait, dans l'introduction, la confusion classique entre parallèle et sériel, sachant que le ICL 7109 de Harris (qui a repris Intersil, le fabricant d'origine de ce composant) possède bien une interface pouvant travailler soit en mode parallèle soit en mode sériel.

Cette remarque faite, passons aux choses sérieuses. Le synoptique de la structure interne représenté en **figure 1** ouvre un premier hublot sur le concept à la base de cette réalisation.

Sur la gauche on découvre 3 blocs d'entrée à comportement de filtre passe-bas, dénommés **HI'**, **HI** et **LO**. Les 2 blocs **HI** et **LO** symbolisent 8 canaux d'entrée que l'on peut sélecter individuellement à l'aide du bloc de commande «*SELECT*» placé en-dessous. Le bloc «*SELECT*» vertical placé en aval des sélecteurs d'entrée permet le choix entre l'entrée directe **HI'** et les entrées commutables.
Le convertisseur Analogique/Numérique (A/N, *A/D Converter* en anglais) monté à la suite de ce bloc de sélection transmet, par l'intermédiaire d'un bus parallèle, les données numériques à un convertisseur parallèle-série. L'information disponible à la sortie du convertisseur n'est pas reliée à l'entrée de données RxD (***R**eceive **D**ata*) de l'interface RS-232, mais à l'entrée de commande CTS (***C**lear **T**o **S**end*). Cette technique de connexion permet une commande bien plus aisée de la boîte (noire) de mesure que ne l'aurait permis une connexion effectuée selon les normes.
Le bloc d'alimentation («*POWER SUPPLY*») représenté dans la partie supérieure droite du schéma tire son énergie directement et exclusivement de l'interface sérielle de l'ordinateur. Il fournit 4 tensions d'alimentation régulées : ±5 V pour le convertisseur et ±12 V pour l'interface.

Problèmes de connexion

Nous avons, pour ce montage, opté pour l'interface sérielle pour la simple et bonne raison que l'interface parallèle n'est pas en mesure de fournir un courant suffisant pour l'alimentation de la totalité du circuit du **convertisseur RS-232 – A/N**. L'expérience nous a en outre appris qu'il est plus fréquent, sur un ordinateur du type PC, de trouver une seconde interface sérielle de disponible qu'une interface parallèle inutilisée : les ordinateurs à 2 interfaces parallèles sont extrêmement rares. Avantage additionnel : une interface sérielle permet l'établissement de liaisons plus longues que celles qu'aurait autorisé une interface parallèle.

À l'inverse, la transmission de données via une interface sérielle prenant notablement plus de temps que leur émission par l'intermédiaire d'une interface parallèle, seuls pourront être utilisés efficacement des convertisseurs relativement

Caractéristiques techniques :

- Travaille avec tout PC doté d'une interface sérielle
- 16 canaux d'entrée avec utilisation d'un multiplexeur
- convertisseur A/N à 12 bits
- Support logiciel à menu déroulant avec traitement des données et leur affichage en fenêtre
- Se passe d'alimentation propre
- Coût de réalisation faible de par l'absence de circuit intégré «exotique»

lents. Cette caractéristique ne constitue pas nécessairement un inconvénient, sachant, primo, qu'il est facile de trouver des convertisseurs A/N plus lents mais présentant cependant une résolution élevée et (relativement) bon marché et, secundo, qu'il n'est pas impératif d'échantillonner à un taux frisant le kHz des grandeurs qui ne varient que lentement.

Notre montage fait appel, comme nous le disions plus haut, à un ICL 7109 qui ne diffère du ICL 7106 plus courant utilisé dans de nombreux montages et multimètres que par son interface numérique. La commutation des entrées est pratiquement identique sur ces 2 composants.
Lors de la mise en oeuvre selon les normes d'une interface sérielle, ce sont les paramètres d'interfaçage qui peuvent être la source de problèmes. Il faut en effet alors non seulement que l'émetteur et le récepteur respectent le même format de données (nombre des bits de données, celui des bits de début/d'arrêt) mais encore qu'ils utilisent le même taux de transmission (*baudrate*). Tous ceux d'entre vous ayant déjà eu affaire à une interface sérielle savent combien de temps il faut avant d'avoir trouvé la bonne disposition à donner aux nombreux contacts de l'interrupteur DIL, si tant est qu'il n'y en ait pas plusieurs, et avant d'avoir défini le chemin à suivre parmi les multiples menus de configuration nécessaires à la définition des paramètres.

Nous avons opté, bien entendu diront les mauvaises langues, pour une transmission des données **non conforme** à la norme.
La transmission des données proprement dite fait appel à la ligne d'état **CTS**, le signal d'horloge étant fourni par la ligne d'état **RTS**. Ce n'est pas sans de bonnes raisons que nous avons choisi cette approche.
■ Le logiciel détermine l'instant de transmission des données et la vitesse à laquelle se fera ce processus, il n'est donc pas nécessaire de procéder à la configuration de l'interface. Tout l'agrément de cette solution apparaît lorsque l'on utilise cette interface pour plusieurs appareils externes.
■ Il faut, si l'on veut effectuer une transmission selon les normes, utiliser soit des circuits intégrés spécialisés gros consommateurs de courant dotés d'un oscillateur interne ou associés à un oscillateur externe, soit une circuiterie discrète à la complexité redoutable.

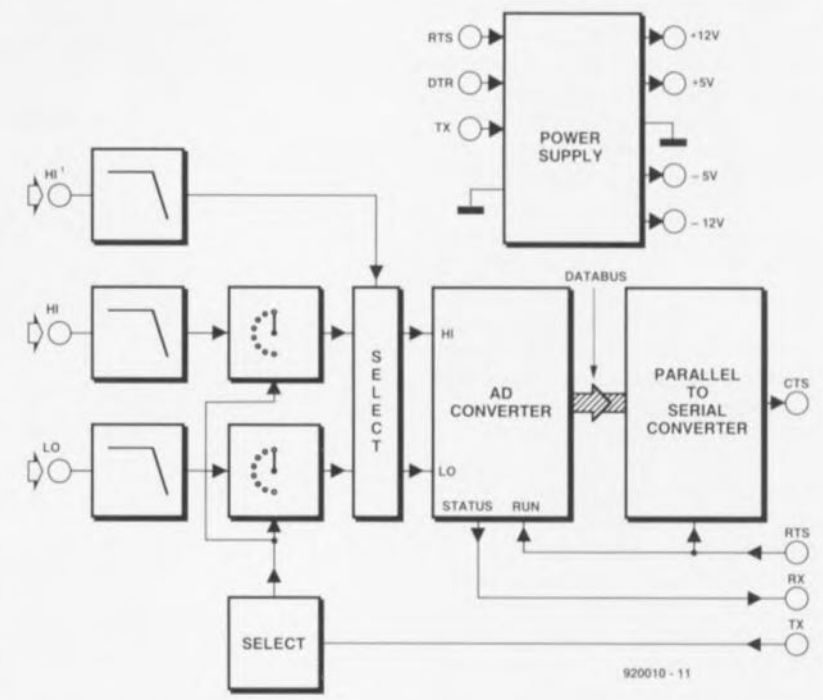


Figure 1. Synoptique de la structure interne du 7109.

■ Il est possible d'utiliser ce montage avec d'autres interfaces ne respectant pas la norme RS-232 au pied de la lettre ou encore avec des microcontrôleurs.
■ La complexité de la réalisation en est notablement réduite.

Les détails

On peut subdiviser le schéma représenté en **figure 2** en 4 sous-ensembles fonctionnels. Faisons un tour rapide d'horizon :
le quadruple amplificateur opérationnel IC6 se transforme en interface sérielle, IC4, IC7 et IC8 constituent un multiplexeur d'entrée rustique, IC1 le convertisseur A/N proprement dit. Les 2 circuits du type 74HC166, IC2 et IC3 forment à eux deux un registre à décalage à 16 bits servant à la conversion parallèle-série, les 4 portes EXOR intégrées dans le 74HC02, IC5, démarrent la conversion A/N et génèrent le signal d'horloge pour les registres à décalage.
Il reste, pour finir, l'alimentation qui rassemble les différentes tensions présentes sur les lignes de l'interface RS-232.

Le processus de conversion est lancé par un changement de niveau à l'entrée **RTS** (***R**equest **T**o **S**end* = demande d'émission). L'amplificateur opérationnel IC6a donne au signal **RTS** un niveau TTL; son signal de sortie entraîne le basculement de la bascule bistable (flip-flop) constituée par les portes NOR (NI) IC5c et IC5d. Le niveau logique haut présent sur la broche 10 de IC5 lance la conversion A/N effectuée par IC1.
Une fois le processus de conversion terminé, le 7109 charge, par l'intermédiaire de sa ligne **$\overline{\text{LOAD}}$** et via les portes NOR IC5a et IC5b, l'octet de poids fort du résultat de mesure et les signaux d'état dans le registre à décalage IC2 puis l'octet de poids faible de ce résultat dans le registre à décalage IC3.

La mise en oeuvre des signaux $\overline{\text{HBEN}}$ (***H**igh **B**yte **EN**able* = validation de l'octet de poids fort) et $\overline{\text{LBEN}}$ (***L**ow **B**yte **EN**able* = validation de l'octet de poids faible), tous 2 actifs au niveau bas comme l'indique la barre horizontale qui surplombe leur dénomination, est plus délicate.
Lorsqu'ils présentent un niveau logique bas ils commutent le mode de travail du registre à décalage concerné le faisant passer de la fonction de décalage à celle de chargement (*Load*).
Le fait primo qu'il soit impossible que ces 2 signaux présentent un niveau logique bas au même instant et secundo que leurs entrées d'horloge sont interconnectées, a pour conséquence que, lors du chargement de l'un des registres, l'autre se trouve en mode de décalage (*Shift*). Si cela est sans importance lors du chargement de l'octet de poids fort, on a décalage de l'octet de poids fort dans IC3 lors du char-

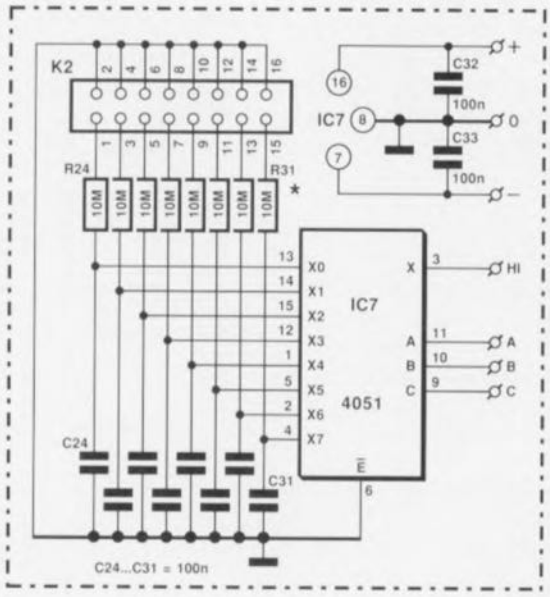
gement de l'octet de poids faible. Les bits 7 et 8 de l'octet de poids fort inutilisés par le convertisseur sont forcés respectivement au niveau logique bas et haut et entraînent, après décalage, un changement de niveau de la ligne **CTS** après le chargement de l'octet de poids faible.

On pourra, sur un PC, utiliser ce changement de niveau pour le déclenchement d'une interruption ou, par son interrogation constante (*polling*), pour s'en servir comme signal d'état indiquant la fin d'une conversion A/N.
Avant que le logiciel ne puisse, par l'application de 16 impulsions d'horloge sur la ligne **RTS**, lire le contenu du registre à décalage bit après bit, il faudra commencer par patienter pendant une durée égale à 1,5 cycle d'horloge (soit 30 µs environ), jusqu'à ce que le signal $\overline{\text{LBEN}}$ soit repassé au niveau logique haut. On pourra utiliser cette durée pour, par exemple et entre autres choses, basculer le multiplexeur sur le canal requis. Le **tableau 1** récapitule la signification de chacun des bits.

Impulsion sur RTS	Valeur
1	Dépassement (OR)
2	Polarité (POL)
3	B12
4	B11
5	B10
6	B9
7	Haut
8	B8
9	B7
10	B6
11	B5
12	B4
13	B3
14	B2
15	B1
16	Haut

Tableau 1. Il faut à chaque fois 16 cycles d'horloge pour la prise en compte d'une valeur de mesure complète. Signification de chacun des bits.

La première impulsion d'horloge positionne (met à «1») la bascule RS bistable IC5c/IC5d. Ce faisant, l'entrée *RUN* est remise au niveau logique haut, situation qui se traduit par le lancement d'un nouveau processus de conversion. Le bistable est remis à zéro par le niveau haut présent sur la ligne d'état de IC1 (*STATUS*, broche 2).
Le niveau logique haut à l'entrée sérielle du registre à décalage de l'octet de poids faible garantit que la sortie prend en tout cas, après la lecture, un niveau haut (soit −12 V pour la ligne **CTS**, n'oublions pas que RS-232 travaille en logique négative).



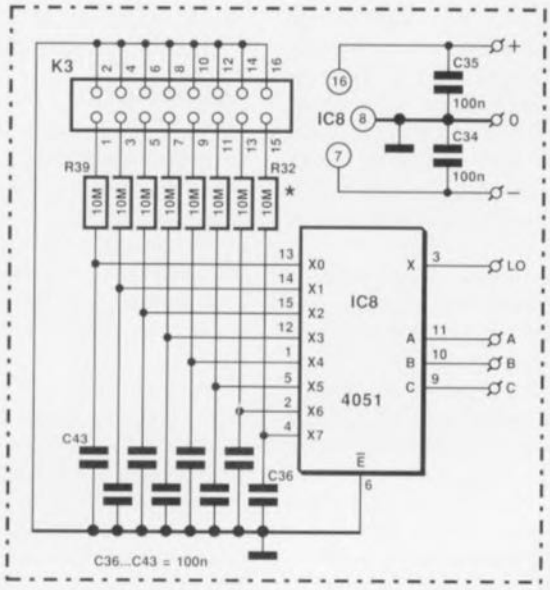
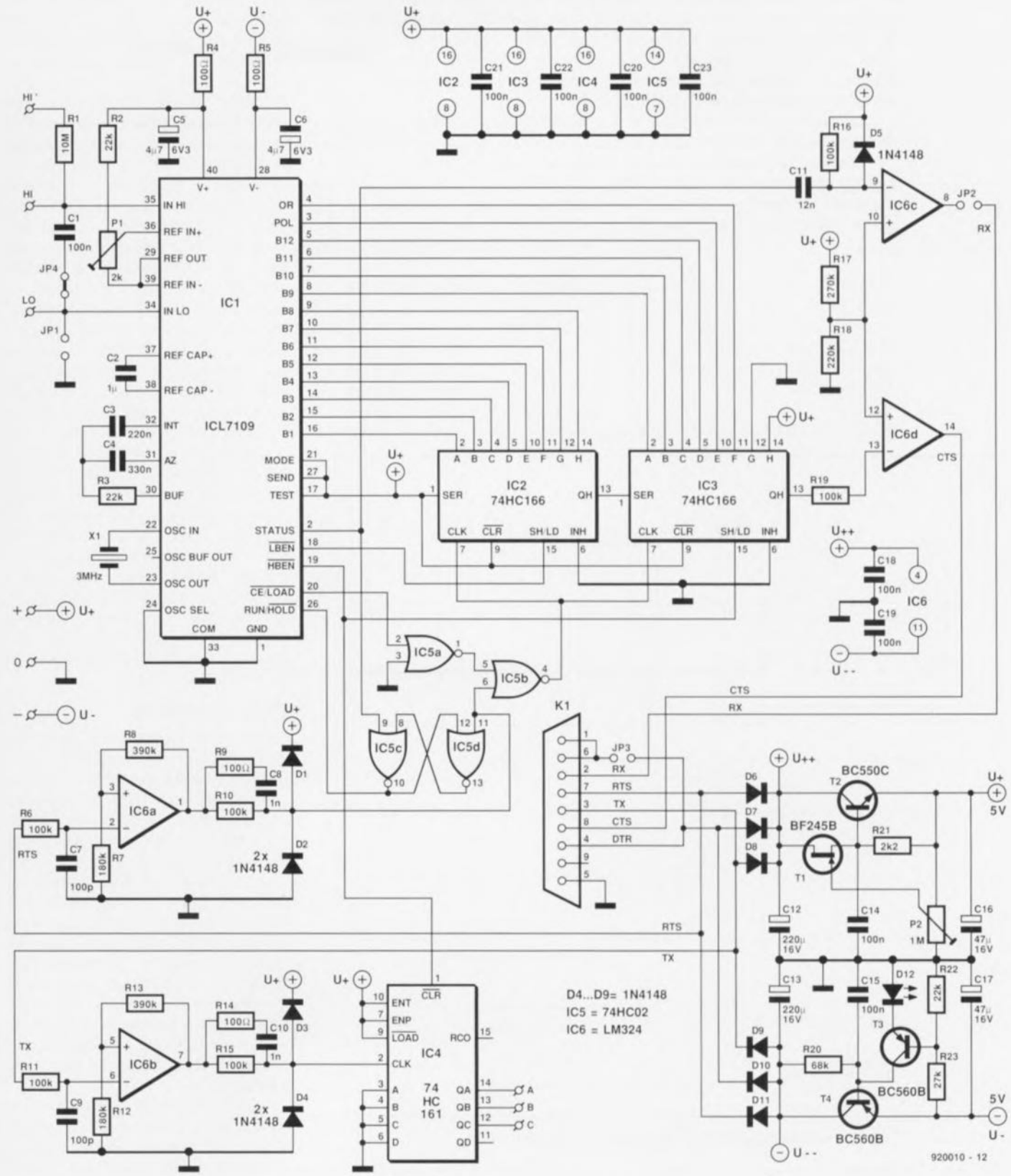


Figure 2. Il est relativement simple de retrouver sur ce schéma les sous-ensembles du synoptique. L'alimentation est passée dans le coin inférieur droit.

L'ICL 7109

est un convertisseur A/N à 12 bits à pente double (relativement) peu coûteux (une bonne centaine de francs, voire moins) doté de sorties additionnelles fort intéressantes telles que détection de polarité et de dépassement.
Nous allons faire appel aux chronodiagrammes des **figures 3** et **4** pour essayer de comprendre le fonctionnement de la partie numérique du convertisseur.



Commençons par la figure 3. Le premier signal, à partir du haut, rend le niveau de la tension présente aux bornes du condensateur C3. On retrouve ensuite un certain nombre des signaux du 7109, le signal d'horloge interne (*Internal Clock*), le signal d'horloge du verrou de sortie (*Internal Latch*), celui présent à la sortie d'état (*STATUS OUTPUT*) et à l'entrée **RUN/HOLD**. La mise au niveau haut de l'entrée **RUN** démarre un processus de conversion.

Tout au long de la conversion la sortie d'état reste au niveau logique haut. La première étape du processus consiste à effectuer une charge du condensateur de l'intégrateur, C3, à un courant proportionnel à la tension d'entrée ((U_{HI} – U_{LO})/R3), et ce pendant 2 048 impulsions d'horloge (fréquence de l'oscillateur/58) avant que ne soit

lancé un processus de décharge de ce condensateur à un courant constant ($U_{ref}/R3$). Lors de la décharge on en mesure la durée par un comptage des impulsions d'horloge interne. Le premier flanc montant du signal d'horloge interne se présentant après que la tension aux bornes du condensateur soit tombée en-dessous de zéro produit un transfert du contenu du compteur dans le registre de sortie, la sortie d'état passe au niveau logique bas, indiquant ainsi la fin de la conversion. On a :

Contenu du compteur = $2\,048 \cdot U_{in}/U_{ref}$,

formule dans laquelle U_{in} est la tension d'entrée et U_{ref} la tension de référence.

Si l'entrée **RUN** reste au niveau logique haut tout au long de la conversion, on aura, après une phase d'auto-zéro, le lancement d'une nouvelle conversion; sinon, on aura d'abord 7 impulsions d'horloge après la remise au niveau logique haut de cette entrée.

Le chronodiagramme de la figure 4 montre les signaux d'acquittement (*handshaking*) utilisés par le ICL7109 lors de sa communication avec le monde extérieur lorsque les entrées «*Mode*» et d'émission («*Send*») se trouvent au niveau haut.
Outre les signaux déjà présents sur le chronodiagramme de la figure 3 nous retrouvons ici les signaux $\overline{LOAD}$, $\overline{HBEN}$, $\overline{LBEN}$ et les lignes de données. La mise au niveau bas de la ligne $\overline{HBEN}$ indique la présence à la sortie de l'octet de poids fort du contenu du compteur et des lignes d'état de dépassement (*OverRun*) et de Polarité. Cet octet peut être pris en compte par le flanc montant du signal $\overline{LOAD}$. On retrouve le même scénario pour le transfert de l'octet de poids faible.

L'entrée du ICL7109 peut être considérée, à l'intérieur du domaine de la tension d'alimentation, comme étant symétrique; elle présente une résistance (impédance d'entrée) supérieure à 1 GΩ. Si l'on veut disposer d'une réserve de charge suffisante pour les commutateurs de la partie analogique, il faudra tamponner la source de tension à mesurer à l'aide d'un condensateur ayant une tension de fuite la plus faible possible. En cas de mesure sans multiplexeur il faut de ce fait, à l'aide du cavalier de court-circuit **JP4**, mettre le condensateur C1 en parallèle sur les 2 entrées. Lors d'une mesure symétrique il faut que les tensions appliquées aux 2 entrées se situent, en prenant la masse comme référence, à l'intérieur du domaine des 2 tensions d'alimentation de l'ICL7109; dans le cas d'une mesure asymétrique on pourra mettre l'entrée **LO** à la masse par la mise en place du cavalier de court-circuit **JP1**. La résistance R1 empêche une décharge via la source de tension de mesure.

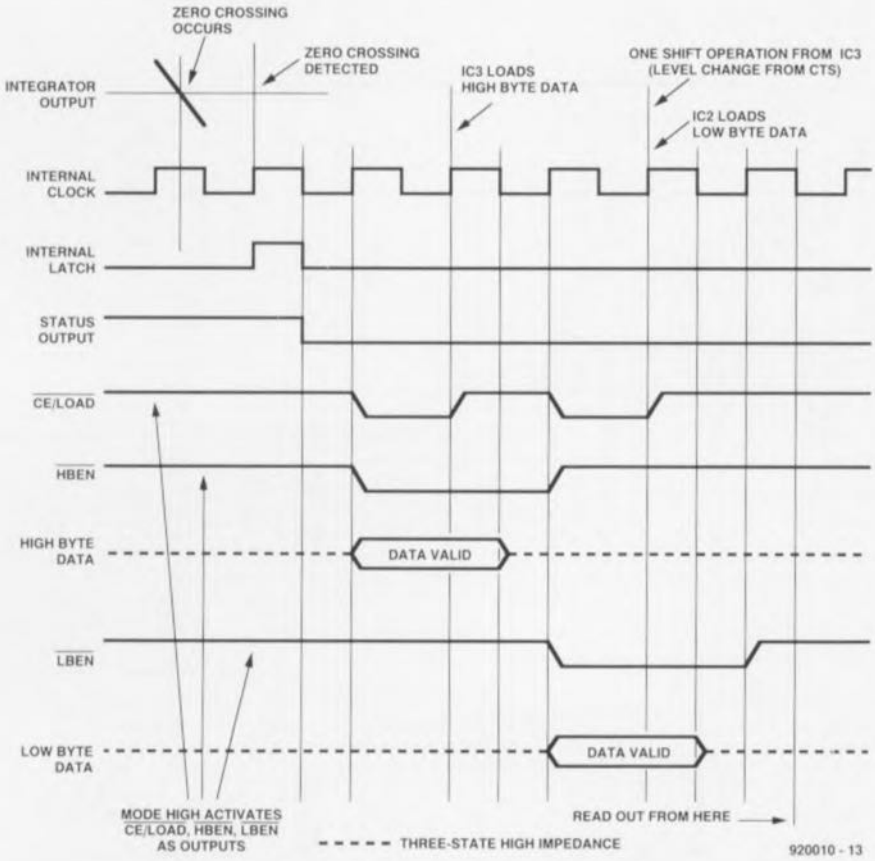


Figure 3. Chronodiagramme d'un processus de conversion.

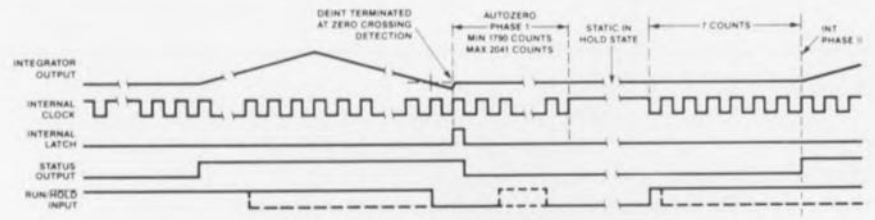


Figure 4. Ce chronodiagramme explicite le processus d'acquittement suivi par le convertisseur A/N.

Plus d'entrées

Nous avons imaginé, à l'intention de ceux d'entre vous qui voudraient mesurer plus d'une source, un multiplexeur rustique. Il prend la forme de IC7 pour la sélection de la ligne **HI** et de IC8 pour celle de la ligne **LO**. Ces 2 circuits sont commandés par IC4 qui compte le nombre d'impulsions véhiculées par la ligne **TX**. Il faut, pour obtenir la sélection d'un canal, que le logiciel mette les caractères convenables sur la ligne **TX** de manière à ce que le nombre total d'impulsions corresponde au canal requis (on consultera l'exemple de programmation «PrepCanal» donné dans le **Listing 1**).
Le choix du taux de transmission ne joue pas de rôle ici; il faudra cependant tenir compte de l'utilisation d'une éventuelle fonction de parité (rarement utilisée en règle générale). Le compteur est automatiquement, après chaque conversion, remis sur le canal 1 de manière à ce que l'on se trouve toujours dans une situation de départ parfaitement définie (initialisation).

En cas d'utilisation d'un multiplexeur il faut déconnecter le condensateur C1 de l'entrée

(suppression du cavalier **JP4**) et éliminer la référence par rapport à la masse (enlever le cavalier **JP1**).
Les condensateurs C24 à C31 et C36 à C43 reprennent alors la fonction de C1.
Il est également possible, avec un multiplexeur, d'effectuer des mesures symétriques à l'intérieur du domaine de la tension d'alimentation du convertisseur.
En cas de mesures asymétriques il faudra relier les entrées concernées (!) du multiplexeur IC8 à la masse par la mise en place des cavaliers correspondants sur l'embase K3.

Le 4051 présente malheureusement, en raison de l'impédance d'entrée élevée du convertisseur, une tension de dérive (*offset*) notable qui varie d'un canal à l'autre.
La structure symétrique réduit bien dans une certaine mesure l'effet de cette tension, mais ne peut malheureusement pas l'éliminer complètement. Si donc on envisage d'effectuer des mesures exigeant une précision élevée, il faudra, soit se passer de multiplexeur, soit utiliser un multiplexeur d'un autre type. Pour les applications domestiques on pourra se contenter de procéder, après avoir fait un étalonnage du zéro, à une correction par logiciel de la tension mesurée.

Les adaptateurs de niveau

Les 4 amplificateurs opérationnels intégrés dans IC6 servent d'adaptateur de niveau. Les amplificateurs sont montés en commutateurs de valeur de seuil inverseurs et éliminent de ce fait l'inversion, caractéristique de la norme RS-232. Les diodes D1 à D4 servent, en association avec les résistances R10 et R15 respectivement, à la limitation du niveau du signal à +5 V et la masse respectivement. Les réseaux RC R9/C8 et R14/C10 servent à éviter que les capacités d'entrée de IC5 et de IC4 respectivement ne doivent être chargées via les résistances R10 et R15 respectivement et limitent ainsi les conséquences de la pente relativement faible de l'amplificateur opérationnel.

L'amplificateur opérationnel IC6c n'est pas, à strictement parler, indispensable au bon fonctionnement du circuit. Si cependant vous désirez déclencher une interruption dans votre ordinateur via la ligne **RX**, vous pourrez implanter le cavalier de court-circuit **JP2**; vous pourrez

Liste des composants

Résistances :
R1,R24 à R39 = 10 MΩ
R2,R3,R22 = 22 kΩ
R4,R5,R9,R14 = 100 Ω
R6,R10,R11,R15,
R16,R19 = 100 kΩ
R7,R12 = 180 kΩ
R8,R13 = 390 kΩ
R17 = 270 kΩ
R18 = 220 kΩ
R20 = 68 kΩ
R21 = 2kΩ2
R23 = 27 kΩ
P1 = 2 kΩ ajust multitour
P2 = 1 MΩ ajust multitour

Condensateurs :
C1,C14,C15,C18 à C43 = 100 nF
C2 = 1 μF
C3 = 220 nF
C4 = 330 nF
C5,C6 = 4μ7F/6V3 tantale
C7,C9 = 100 pF
C8,C10 = 1 nF
C11 = 12 nF
C12,C13 = 220 μF/16 V radial
C16,C17 = 47 μF/16 V radial

Semi-conducteurs :
D1...D11 = 1N4148
D12 = LED verte
T1 = BF245B
T2 = BC550C
T3,T4 = BC560B
IC1 = ICL7109 (Harris Semiconductor)
IC2,IC3 = 74HC166
IC4 = 74HC161
IC5 = 74HC02
IC6 = LM324
IC7,IC8 = 4051

Divers :
K1 = connecteur sub-D femelle coudée à 9 contacts encartable
K2,K3 = connecteur HE-10 à 2 x 16 contacts mâle
X1 = quartz 3 MHz

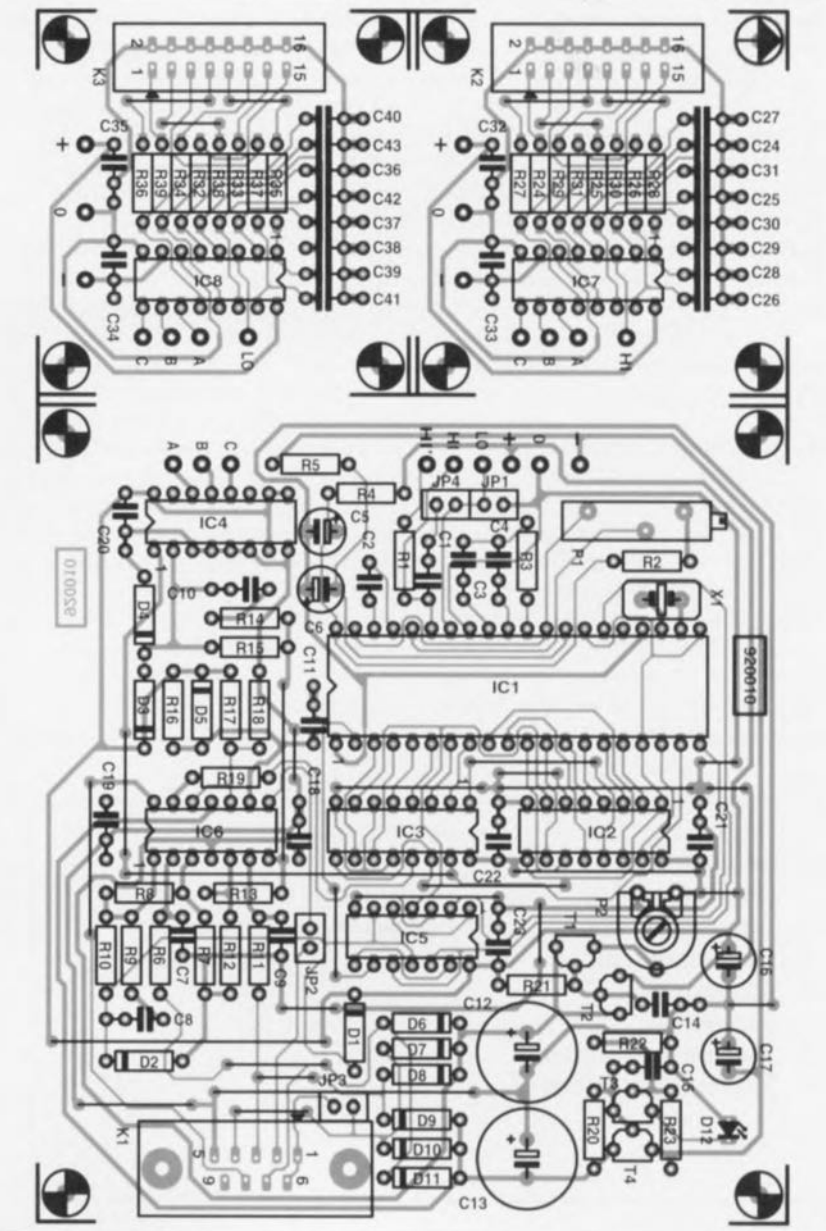


Figure 5. Représentation de la sérigraphie de l'implantation des composants du circuit imprimé dessiné à l'intention de cette réalisation. Il se subdivise en 3 parties : la platine principale et les 2 platines des multiplexeurs.

alors observer une impulsion sur la ligne **RX** à la fin de chaque processus de conversion.

L'alimentation

se fait via l'interface de l'ordinateur. Pour ce faire il faut que le logiciel commute au minimum une ligne à +12 V et une autre à −12 V. Sachant que dans le cas du respect de la norme RS-232 la ligne **TX** se trouve, au repos à −12 V, seules entrent encore en ligne de compte pour la fourniture du +12 V, les lignes **RTS** ou **DTR**.
Les diodes D6 à D11 recueillent les tensions que tamponnent les condensateurs C12 et C13. Les tensions disponibles aux bornes des dits condensateurs servent à l'alimentation en courant des amplificateurs opérationnels.
L'ouverture du pont fermé par le cavalier **JP3** permet de limiter au strict indispensable la charge pour l'interface au cas où celle-ci ne pourrait pas fournir le courant nécessaire.
La régulation de tension utilisée ici mérite que l'on s'y attarde. En effet elle se caractérise non seulement par la faible chute de tension (*low drop*) qu'elle introduit, caractéristique importante au cas où l'interface ne serait pas en mesure de fournir les 12 V, mais encore par son efficacité d'utilisation du courant fourni: un exemple de frugalité.

La régulation négative prend la forme d'une pseudo-diode zener, la LED verte, et d'un transistor-série, T3. Vu la taille importante du domaine de tension négative du convertisseur A/N, il n'est pas nécessaire de procéder au réglage de la tension négative.
Il en va autrement en ce qui concerne la tension positive, tension pour laquelle c'est la tension de fin de courant de drain (*pinch off*) d'un FET qui sert de référence. Bien que cette tension soit relativement stable, il faut cependant l'ajuster en raison des tolérances existant entre les différents FET, réglage qui se fait par action sur la résistance ajustable P2.
Comme nous voulions limiter le nombre de composants nécessaires à cette réalisation, nous n'avons pas prévu de protection contre des surtensions en cas de réglage erroné. Il faudra donc, avant d'entreprendre l'étalonnage, et donc avant la première mise sous tension, tourner le curseur de cet ajustable vers la masse. Le réglage de la tension de +5 V se fera après avoir mis celle-ci sous charge (résistance de caractéristiques convenables).

Réalisation et étalonnage

La **figure 5** vous propose la représentation de la sérigraphie de l'implantation des composants de la platine simple face dessinée pour ce montage. On voit qu'elle comporte en fait 3 platines : la platine principale et les 2 platines des multiplexeurs.
Une fois l'implantation des composants sur la platine du multiplexeur terminée, il restera à en effectuer l'interconnexion avec la platine principale en respectant les indications de la sérigraphie.
L'entrée **HI'** sera utilisée pour les premiers essais sans multiplexeur. Il est essentiel de veiller à un blindage correct de toutes les lignes véhiculant des signaux, sous peine de se trouver rapidement confronté, en raison de l'importance de l'impédance d'entrée, aux effets de tensions parasites captées par des lignes insuffisamment ou mal blindées.

La meilleure suppression de ronflement est obtenue par l'utilisation d'un quartz de 2,969600 MHz; ce type de quartz n'étant malheureusement pas une valeur standard, nous nous sommes résignés à utiliser un quartz de 3 MHz.

Le réglage du convertisseur par l'intermédiaire de l'ajustable P1 est fonction de l'application envisagée. Il est, en principe, possible d'effectuer l'étalonnage en s'aidant du logiciel. La mise en oeuvre du convertisseur A/N est prévue ici pour une tension d'entrée maximale de quelque 400 mV. Si l'on veut mesurer d'autres tensions il faudra redimensionner la résistance R3 :

$R3 = U_{max}/20\ \mu A$.

La tension U_{ref} (diviseur de tension R2/P1) devrait avoir une valeur légèrement supérieure à $U_{max}/2$, sachant qu'à $U_{max} \geq 2 \cdot U_{ref}$ le convertisseur affiche un dépassement (*overflow* ou *overrun*).

Le logiciel AD232

Bien que les sous-programmes de base proposés en **figure 7** permettent l'écriture de toutes sortes de routines de mesure, il existe aussi pour cette réalisation un logiciel à interface logicielle confortable, **ESS1694**, disponible auprès de adresses habituelles et auquel nous allons nous intéresser d'un peu plus près. Ce programme fonctionne avec tout ordinateur de PC & Compatible, de l'XT au 486 en passant par les 286 et les 386.

L'installation
Le programme fonctionne tout aussi bien (bien que plus lentement) depuis une disquette (faites-en une copie de travail) que depuis le disque dur sur lequel il aura été transféré.
En cas de travail avec une disquette il faudra vérifier l'absence de protection en écriture de la disquette de travail, sachant que le programme y écrit automatiquement un fichier comportant les données de configuration.
Pour la mise sur le disque dur du programme, il n'est pas nécessaire de suivre un processus d'installation quelconque, il suffira, comme cela avait été le cas pour le travail avec une disquette, de recopier les programmes de la disquette originale vers le répertoire requis sur le disque dur.

Lancement et mode d'emploi
Le programme, en anglais, est lancé depuis le niveau du DOS, par

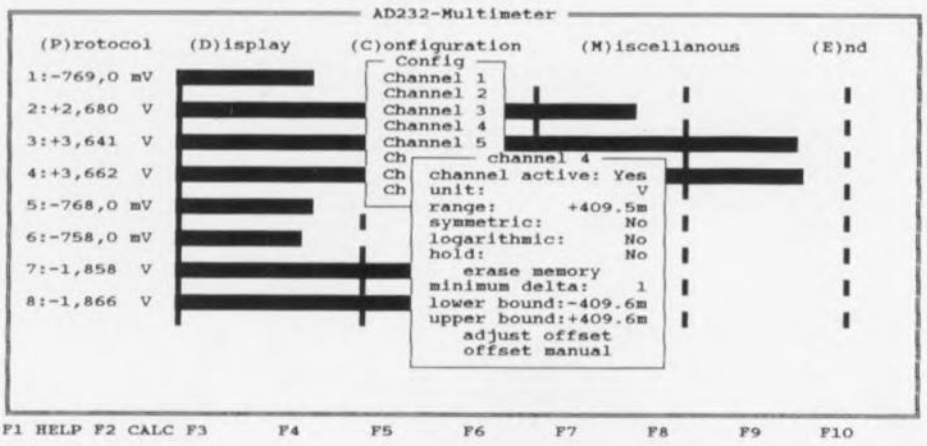


Figure 6. Cette représentation d'écran montre le menu de configuration pour le canal 4.

l'entrée de l'instruction **MULTI**. Lors de la première utilisation du logiciel il faudra indiquer dans le menu «(M)iscellaneous» l'interface (port) à laquelle le **convertisseur RS-232 – A/N** doit être connecté. Après sortie du menu, l'écran principal du programme montre en permanence les valeurs de mesure présentes aux différents canaux. On peut, le cas échéant, supprimer la visualisation d'un canal, voire de plusieurs. Si la visualisation se limite à quelques canaux seulement, le programme agrandit automatiquement les signaux affichés à l'écran.

Options (Menu «(C)onfiguration»)
Les possibilités de choix ne se limitent pas uniquement à la mise en fonction ou non des canaux. Si, de par la présence d'un montage externe additionnel, on utilise un calibre ou une grandeur de mesure différente, le logiciel peut en tenir compte lors de la visualisation. La représentation graphique de la barre peut également se faire sous forme symétrique et/ou logarithmique. Il est possible de la même façon, d'effectuer une compensation de la tension de dérive ou encore d'entrer un facteur d'égalisation servant à tranquilliser l'affichage (le rendre plus stable).

Protocole
Il est possible non seulement d'afficher la valeur de mesure à l'écran, mais encore de la représenter sous une forme ASCII. On pourra envoyer le résultat soit directement vers une imprimante soit encore le stocker dans un fichier de texte pour un traitement ultérieur.

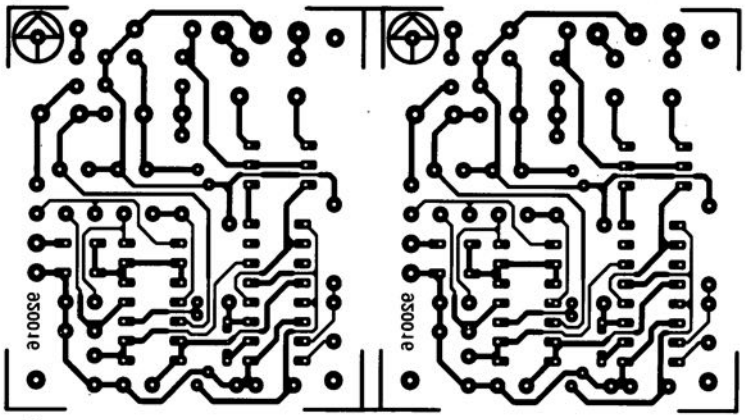
Extras
L'utilisateur a toute liberté quant aux couleurs visualisées à l'écran. Où que l'on se trouve dans le programme, on dispose à tout instant d'une calculatrice RPN –Reverse Polish Notation = à Notation Polonaise Inverse, technique lancée par Hewlett-Packard voici quelques lustres déjà– et d'informations additionnelles variant en fonction du contexte («*Context sensitive*» comme disent si joliment les anglais). Pour cette dernière fonction il faudra bien entendu avoir mis dans le répertoire concerné le fichier de texte en question «**MULTI.HLP**».
Il nous arrive, eh oui même à nous, d'être étonnés de voir l'évolution vers la classe professionnelle de plus en plus des réalisations proposées pour le PC. Il en viendra sans doute d'autres... ■

```
      Uses Crt;

CONST
     PosOver=MaxInt;          {valeur pour dépassement positif}
     NegOver=-MaxInt;         {valeur pour dépassement négatif}

VAR
     SIOAdr:Word;             {Il faudra entrer ici l'adresse de base du 8520}
                              {telle que par exemple COM1: SIOAdr:=$3f8}
                              {                      COM2: SIOAdr:=$2f8}
{-----------------------------------------------------------------------------}
{Préparation du canal pour la conversion suivante}

      PROCEDURE PrepCanal (Canal:Byte);
CONST
     PulseBytes:ARRAY[1..5] OF Byte=(0,14,51,21,85); { -> 1 à 5 impulsions vers TX}

      PROCEDURE AttendTX;
BEGIN
     {attendre jusqu'à ce que le registre à décalage d'émission soit libre}
     REPEAT UNTIL Port[SIOAdr+5] AND 32<>0;
END;
BEGIN
     IF Canal>5 THEN          {nécessite 2 caractères ?}
       BEGIN
         AttendTX; Port[SIOAdr]:=PulseBytes[5];
         AttendTX; Port[SIOAdr]:=PulseBytes[Canal-5];
         AttendTX;
       END
     ELSE IF Canal>0 THEN     {nécessite 1 caractère ?}
       BEGIN
         AttendTX; Port[SIOAdr]:=PulseBytes[Canal];
         AttendTX;
       END
     ELSE;                    {absence d'impulsion pour Canal 0}
END;

{Lecture du contenu des registres à décalage. N'appeler que si la conversion est
terminée !!}

      FUNCTION LireAD:Integer;
BEGIN
     Inline(
$8B/$16/>SIOAdr/ {       mov  dx,[SIOAdr]     ;DX au port SIO }
$83/$C2/$04/     {       add  dx,4            ;et là à ModemCtrl }
$BE/>0002/       {       mov  si,2            ;Pour changement d'adresse }
$33/$DB/         {       xor  bx,bx           ;effacer accumulateur de décalage }
$B5/$10/         {       mov  ch,16           ;14 impulsions de données + 2 fictives }
$B1/$04/         {       mov  cl,4            ;pour le décalage-CTS }
$EC/             { @read:in   al,dx           ;transfert d'impulsion à RTS }
$24/$FD/         {       and  al,0fdh         ;RTS au niveau haut }
$0C/01/          {       or   al,1            ;DTR complémentaire au niveau bas }
$EE/             {       out  dx,al           ;positionnement des lignes }
$51/             {       push cx              ;boucle d'attente brève, }
$B9/>$1F4        {       mov  cx,500          ;pour que les niveaux }
$E2/$FE/         {       loop $               ;se stabilisent }
$59              {       pop  cx              }
$0C/$02/         {       or   al,2            ;révoquer une nouvelle fois }
$24/$FE/         {       and  al,)feh         ;DTR complémentaire au niveau haut }
$EE/             {       out  dx,al           ;remise à zéro des lignes }
$51/             {       push cx              ;garantie pour l'obtention de la }
$B9/>$1F4        {       mov  cx,500          ;durée d'impulsion minimale requise }
$E2/$FE/         {       loop $               }
$59/             {       pop  cx              }
$80/$FD/$0A      {       cmp  ch,10           ;sauter Dummy après 7 impulsions }
$74/$0A/         {       je   @nuse           }
$03/$D6/         {       add  dx,si           ;DX à ModemStatus }
$EC/             {       in   al,dx           ;lecture des bits dn'état }
$D2/$E0/         {       shl  al,cl           ;décalage du bit-CTS dans retenue }
$F5/             {       cmc                  ;prendre en compte l'inversion }
$D1/$D3/         {       rcl  bx,1            ;intégrer dans le résultat }
$2B/$D6/         {       sub  dx,si           ;remise de DX à ModemCtrl }
$FE/$CD/         { @nuse:dec  ch              ;compteur de boucle principale }
$75/$D4/         {       jnz  @read           }
$D1/$EB/         {       shr  bx,1            ;décalé une fois de trop }
$80/$F7/$10/     {       xor  bh,10h          ;inversion du signe }
$F6/$C7/$20/     {       test bh,20h          ;Overrun positionné ? }
$75/$0D/         {       jnz  @Over           ;oui -- }
$F6/$C7/$10/     {       test bh,10h          ;polarité négative ? }
$74/$16/         {       jz   @WErg           ;non -- résultat O.K. }
$80/$F7/$10/     {       xor  bh,10h          ;oui -- efface indicateur de polarité .. }
$F7/$DB          {       neg  bx              ;.. et constitution du complément à 2 }
$EB/$0F/$90/     {       jmp  @WErg           }
$F6/$C7/$10/     { @Over:test bh,10h          ;dépassement négatif ? }
$75/$06/         {       jnz  @ONeg           ;oui -- }
$8B/>PosOver/    {       mov  bx,PosOver      ;constante de dépassement positif }
$EB/$04/$90/     {       jmp  @WErg           }
$BB/>NegOver     { @ONeg:mov  bx,NegOver      ;constante de dépassement négatif }
$89/$5E/$FE);    { @WErg:mov  @result,bx      ;résultat de la fonction = BX }
END;

{Vérification si résultat valable est présent:}

      FUNCTION AD232_Pret:Boolean;
BEGIN
     AD232_Pret:=Port[SIOAdr+6] AND 16<>0;
END;

{Initialisation de AD232.  ATTENTION !!  Absence de délai d'attente !}
{En l'absence de AD232 ceci fait entrer dans une boucle refermée sur elle-même !!}

      PROCEDURE AD232_Init;
VAR
     Dummy:Integer;
BEGIN
     Port[SIOAdr+4]:=2;          {uniquement DTR au niveau bas }
     Delay(100);                   {attendre stabilisation de l'alimentation }

     REPEAT UNTIL AD232_Pret;  {attendre l'arrivée du premier résultat }
     Delay(1);
     Dummy:=LireAD;                {se débarasser du résultat }
END;
```

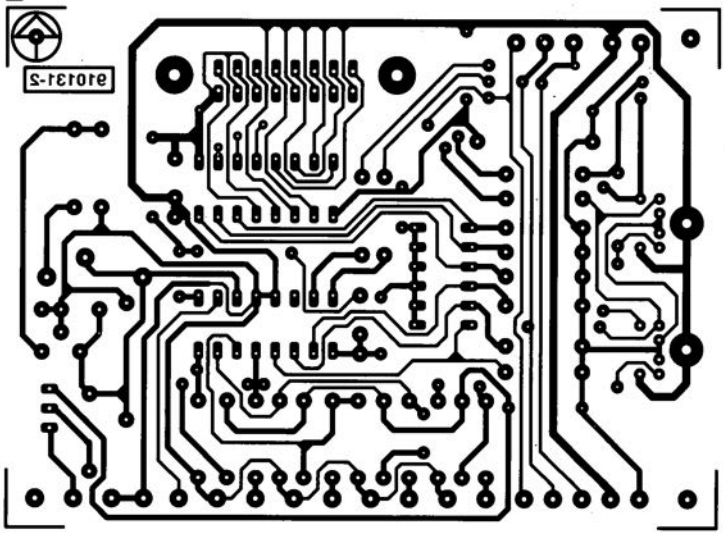
Figure 7. Listing des sous-programmes nécessaires aux premiers essais du montage. La fonction LireAD écrite en assembleur Inline sert à l'interrogation du convertisseur A/N lorsque la chronologie devient plus critique.

1 témoin d'écrêtage
2 convertisseur A/N-N/A pour I²C
3 convertisseur RS-232-A/N

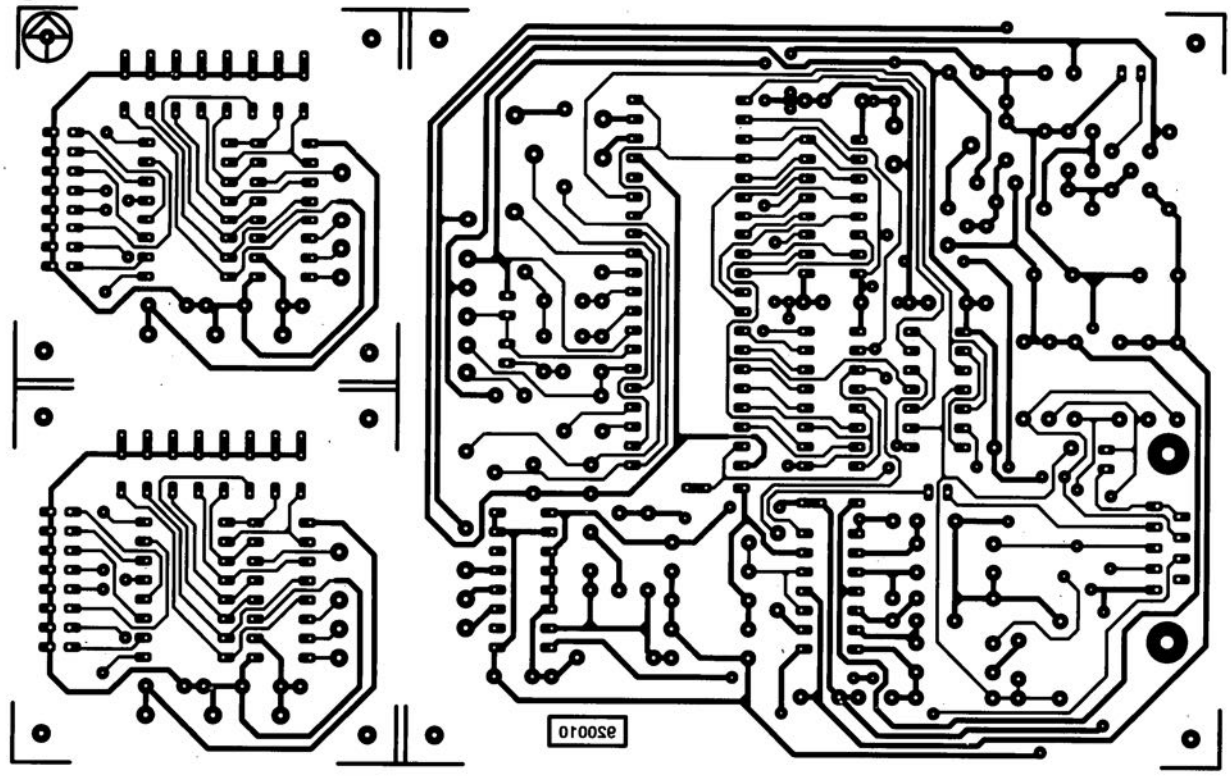
1



2

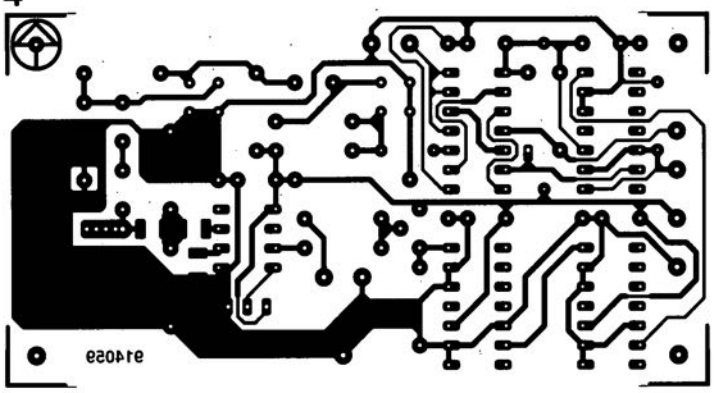


3

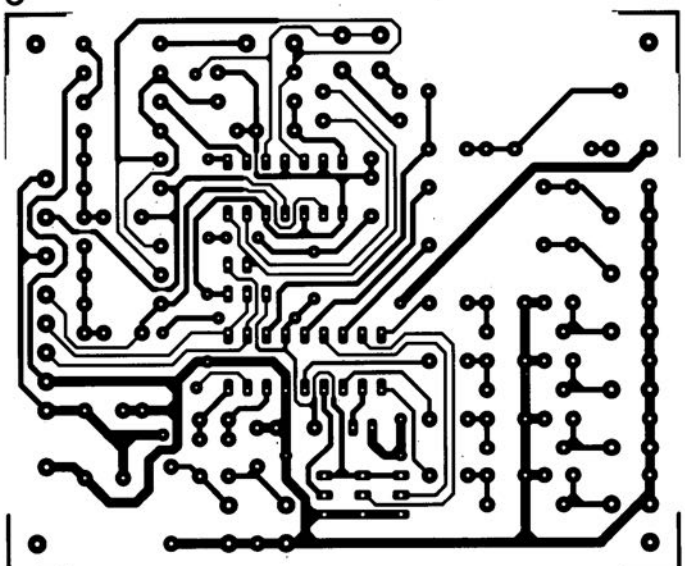


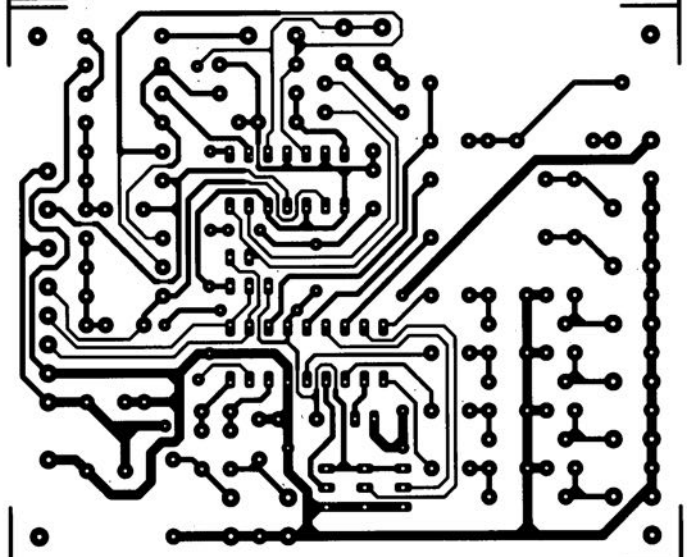
4 prédiviseur 1,3 GHz
5 commutateur audio/vidéo électronique
6 émulateur de 8751 : côté composants
7 émulateur de 8751 : côté pistes

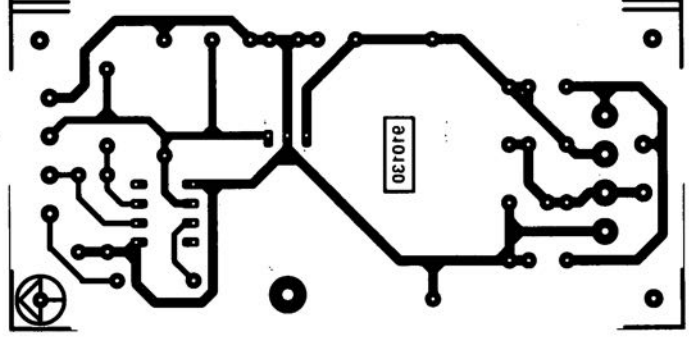
4



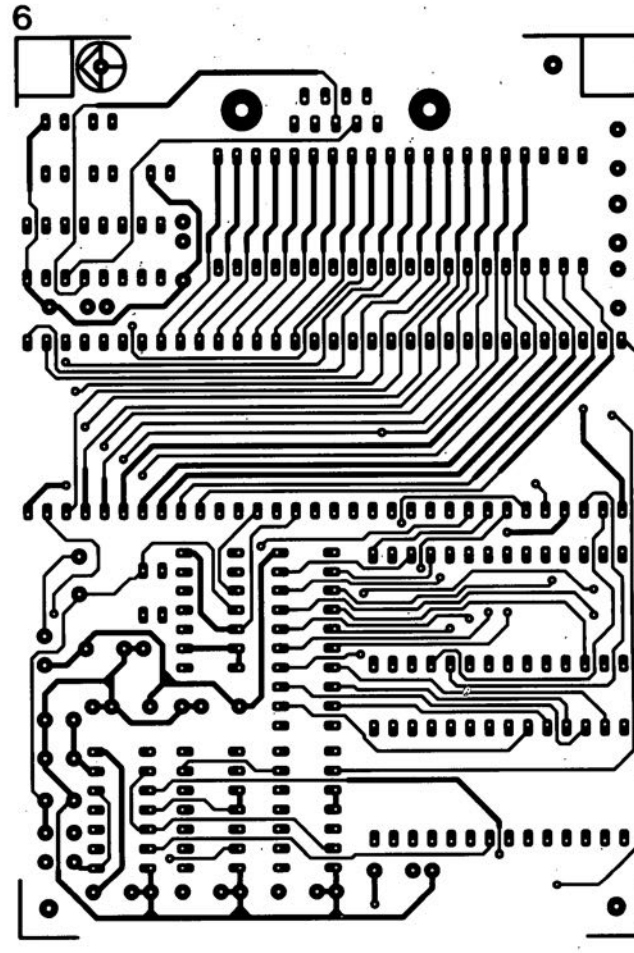
5



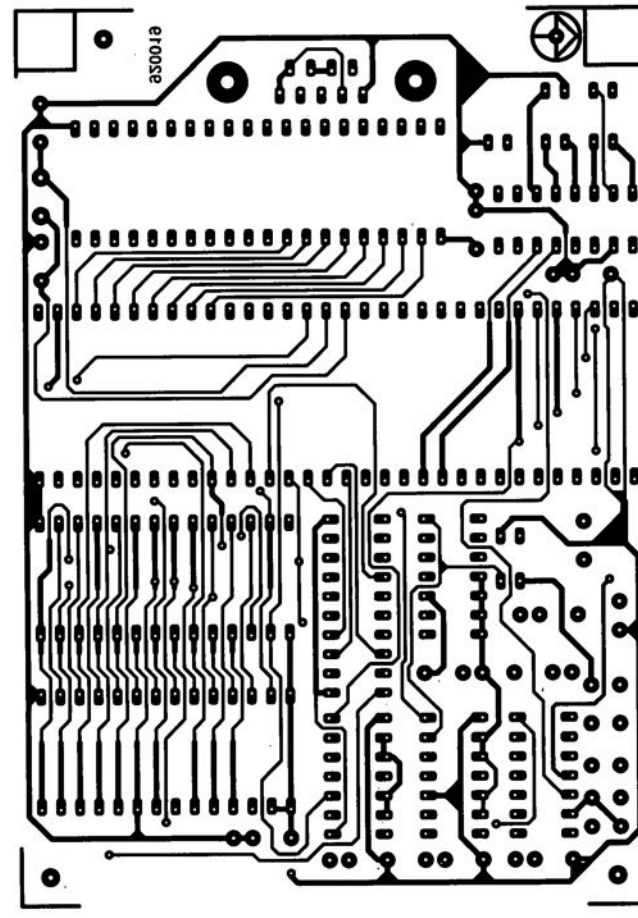




6



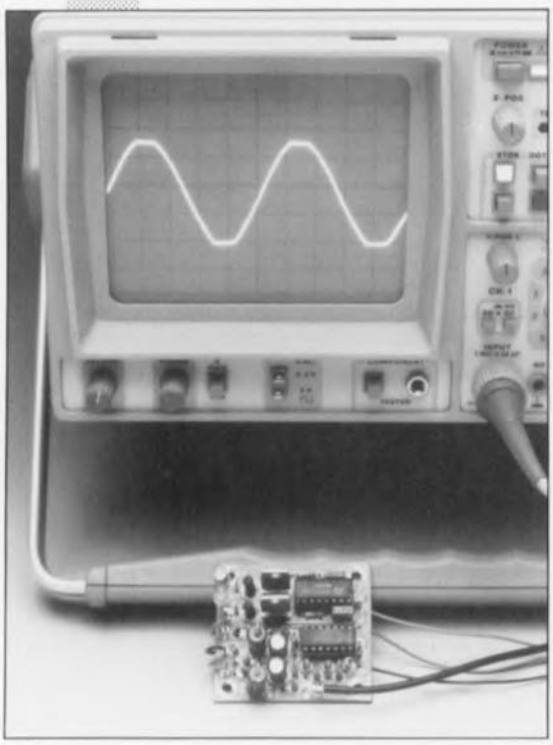
7



témoin d'écrêtage

avec indication −3 dB additionnelle

Le témoin d'écrêtage universel proposé ici peut être connecté, pratiquement, à n'importe quel amplificateur de puissance. Il visualise avec une grande précision les instants où sont atteints, d'abord le point −3 dB de l'amplificateur de puissance et ensuite le point d'entrée en écrêtage de celui-ci. L'utilisateur voit ainsi, lors de la reproduction d'un morceau, d'un seul coup d'oeil quelle est la réserve de puissance disponible.
La LED −3 dB tombe à pic pour indiquer, dans le cas précis du LFA-50-«Optim-A», la limite de domaine de classe A de cet amplificateur haut de gamme.



Nous voulions, à l'origine, vous proposer ce témoin d'écrêtage le mois suivant la publication du second article consacré au **LFA-50-«Optim-A»** (n°160 et 161, octobre et novembre 1991), mais le concepteur de ce projet s'est vu retardé par un certain nombre d'autres projets ayant une priorité encore plus grande.
Nous avons mis à profit ce délai de grâce pour universaliser le concept de cette réalisation, et la débarrasser de tout point de réglage. Le résultat de tout ceci est qu'il vous suffit de construire la (les) platine(s) concernée(s) et de l' (les) implanter ensuite dans l'amplificateur choisi.

Il nous faut revenir au **LFA-50-«Optim-A»** pour mieux saisir les antécédents de notre témoin d'écrêtage.
Comme vous n'êtes pas sans le savoir, fervent lecteur d'Elektor que vous êtes, cet amplificateur fournit une puissance de 25 W dans une charge de 8 Ω en fonctionnement en classe A. Au-delà de ce niveau de puissance, l'amplificateur passe en fonctionnement en classe B et fournit une puissance additionnelle de 25 W, ce qui revient à un total de 50 W. La différence entre 25 et 50 W correspond à 3 dB seulement, de sorte qu'il nous a paru judicieux de concevoir un indicateur d'écrêtage dont les 2 LED indiqueraient le point −3 dB et le point d'entrée en écrêtage de l'amplificateur concerné.
Il est facile ainsi, d'une part de savoir quelle est la réserve de puissance dont on dispose et de l'autre de connaître l'instant de passage en fonctionnement en classe B sachant que la LED −3 dB indique, par son illumination, l'instant où l'amplificateur quitte le fonctionnement en classe A.

Bien qu'à l'origine le témoin d'écrêtage ait été conçu spécifiquement pour le **LFA-50-«Optim-A»**, son concept universel en permet l'utilisation sur tout amplificateur de puissance dont les tensions d'alimentation symétrique sont comprises entre ±30 et ±70 V, ce qui revient à inclure la quasi-totalité des amplificateurs de réalisation personnelle.

La présence d'une LED −3 dB sur un amplificateur travaillant en classe AB n'a rien d'insolite non plus, l'expérience nous ayant appris qu'il n'existe qu'une très faible différence de pression acoustique entre les niveaux −3 dB et 0 dB −qui sonnent d'ailleurs tous deux très forts dès lors que l'on se trouve en présence d'un amplificateur fournissant quelques dizaines de watts. Si, en utilisation, on fait en sorte que la LED −3 dB soit illuminée et que la LED signalant l'entrée en écrêtage ne le soit pas, on s'assure une puissance additionnelle exempte de tout produit de distorsion (dû à une surmodulation). L'illumination de la LED d'écrêtage signale en effet que les choses se sont fortement gâtées et que le signal de sortie comporte une composante en harmoniques non négligeable !

Cette simple constatation nous amène à conseiller l'adjonction de ce montage à tout étage de puissance non doté d'une indication d'écrêtage digne de ce nom.

Comparaison

La **figure 1** vous propose le schéma du **témoin d'écrêtage**. On peut, à première vue, se demander pourquoi un montage aussi simple se doit d'avoir un tel embonpoint, mais l'examen de la platine dessinée à son intention aura de quoi vous rassurer : les dimensions du circuit imprimé sont restées plus que raisonnables.

Le dispositif de mesure prend la forme de 2 paires de comparateurs intégrés dans un unique circuit intégré du type LM339, un quadruple comparateur. Il ne s'agit pas d'un comparateur à la vitesse impressionnante, mais elle est plus que suffisante pour l'application envisagée. L'avantage principal du LM339 est sa faible consommation de courant. Cette dernière caractéristique pèse bien entendu très lourd lorsque l'on envisage d'utiliser ce montage sur une plage de tensions d'alimentation aussi étendue que possible.

L'une des entrées de chacun des comparateurs est reliée à un diviseur de tension chargé de fournir la tension de référence, diviseur de tension constitué par les résistances R3 à R8. Comme ce diviseur est, via les diodes zener D1 et D3, relié directement à la tension d'alimentation de l'amplificateur de puissance, les seuils de commutation des comparateurs sont automatiquement adaptés aux tensions d'alimentation. Les 2 diodes zener de 2,7 V prises aux extrémités du diviseur de tension servent à la compensation de la tension résiduelle de saturation

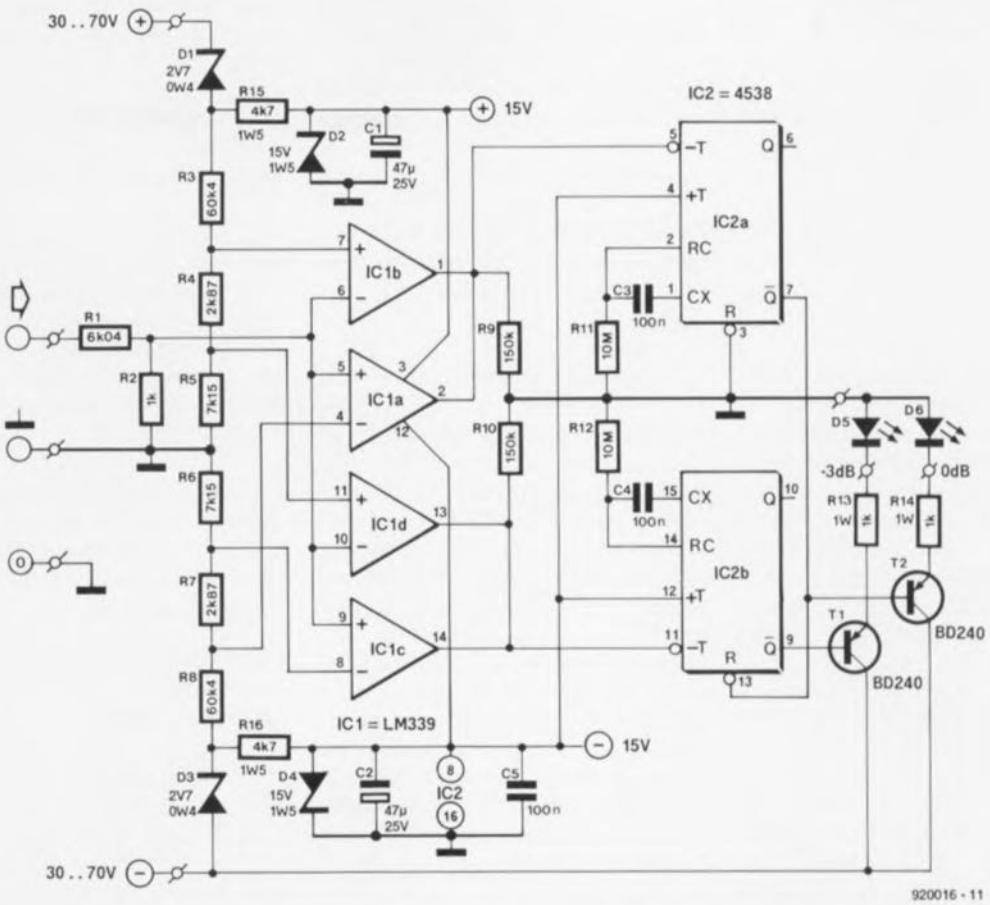


Figure 1. L'embonpoint du schéma du témoin d'écrêtage est, après ces périodes de fêtes, un peu plus grand que ce que nous aurions aimé. Il nous faut 4 comparateurs et 2 multivibrateurs monostables pour visualiser toute impulsion dépassant −3 ou 0 dB.

$U_{DS(sat)}$, relevée entre le drain et la source, présente aux bornes des transistors de puissance et des résistances d'émetteur correspondantes lorsque l'on travaille à pleine modulation −c'est-à-dire transistor en saturation.
Dans ces conditions, la LED de visualisation d'entrée en écrêtage s'illumine juste avant que ne soit atteint le point réel d'entrée en écrêtage.

Sur les amplificateurs de puissance à FETMOS faisant appel à la configuration en source-suiveuse il sera le plus souvent nécessaire, pour les diodes D1 et D3, d'utiliser des diodes zener ayant une tension zener de 10, voire de 12 V, en raison du niveau plus important de la tension résiduelle de saturation présente aux bornes de ce type de transistor (FETMOS).
On peut également envisager de commencer par mesurer la tension de sortie maximale obtenue à charge nominale pour adapter les caractéristiques des diodes D1 et D3 en fonction du résultat des mesures.

Le signal d'entrée du circuit (qui est donc le signal de sortie de l'amplificateur de puissance) y entre par l'intermédiaire d'un diviseur de tension constitué par les résistances R1 et R2, avant d'attaquer ensuite les entrées restantes des comparateurs IC1a à IC1d. Cette atténuation du signal d'entrée a pour but de faire en sorte que les comparateurs n'aient pas à traiter de tension d'entrée supérieure à 10 V −ils sont capables de traiter cette dernière valeur sans le moindre problème vu la tension d'alimentation de ±15 V utilisée ici.

Avec les valeurs de composants proposées ici pour le diviseur de référence et l'atténuateur d'entrée, le circuit peut être utilisé avec des étages de puissance fournissant jusqu'à 300 W dans 8 Ω.

Un coup d'oeil au schéma, il n'en faut pas plus pour saisir le principe de fonctionnement des comparateurs. La sortie de IC1d bascule en cas de dépassement, au cours de la moitié positive de la période du signal, de la moitié de la puissance −ce qui correspond à une tension égale à 0,707 fois la tension de sortie maximale. IC1c réagit de la même manière, mais pour le signal de la moitié négative de la période cette fois.

À pleine modulation, on aura basculement de la sortie de IC1b lorsque cette situation se produit pendant la partie positive du signal. De la même façon, IC1a bascule lorsque l'on a une pleine modulation dans la demi-période négative. On obtient ainsi une indication fiable en cas de dépassement tant dans la partie positive que dans la partie négative du signal de sortie de l'amplificateur de puissance.

Tous les comparateurs ont une sortie à collecteur ouvert, de sorte qu'il n'y a pas d'inconvénient à les monter en parallèle sans qu'il ne soit nécessaire de faire appel à des techniques complexes. C'est ainsi que les sorties de IC1c et de IC1d sont interconnectées (moitié de la puissance, positive/négative) et que les sorties de IC1a et de IC1b (pleine puissance positive/négative) le sont aussi. Les 2 sorties communes ainsi réalisées sont reliées chacune à un multivibrateur monostable, IC2a et IC2b respectivement, dont la fonction est d'effectuer un allongement suffisant des impulsions même les plus

courtes fournies par les comparateurs, de façon à en obtenir une bonne visualisation par les LED.
La pseudo-période de chaque multivibrateur monostable (MVM) est fixée à 1 seconde (durée définie respectivement par les paires R11/C3 et R12/C4). Les sorties $\overline{Q}$ de IC2a et IC2b attaquent les 2 LED via un transistor de commutation, T1 et T2 respectivement, et une résistance de limitation de courant, R13 et R14 selon le cas.

Comme la commutation des LED par l'intermédiaire des transistors se fait directement entre la masse et la tension d'alimentation négative de l'étage de puissance, il n'est pas nécessaire que le courant circulant par les LED –quelque 15 mA par LED– soit fourni par la partie de la tension d'alimentation régulée servant principalement à l'alimentation des circuits intégrés.

Notons au passage que les paires R13/T1 et R14/T2 travaillent en source de courant, sachant que les sorties $\overline{Q}$ de chaque multivibrateur monostable commutent toujours impeccablement entre 0 et –15 V et cela indépendamment de l'alimentation de l'amplificateur.

Intéressons-nous brièvement au comportement de commutation. Lors du dépassement du niveau de référence de l'un des comparateurs, la sortie de ce comparateur passe au niveau bas. Le flanc descendant né de ce basculement déclenche l'entrée –T du multivibrateur monostable monté en aval; de ce fait, la sortie $\overline{Q}$ (qui au repos se trouve à «1») est amenée au niveau bas pendant la durée de la pseudo-période de 1 s. Le transistor relié à la dite sortie devient conducteur et la LED correspondante s'illumine. Il existe en outre une connexion entre la sortie $\overline{Q}$ de IC2a et l'entrée de remise à zéro de IC2b. La présence de cette liaison a pour effet une remise à zéro de IC2b et une extinction de la LED –3 dB lors de l'illumination de la LED 0 dB. On garantit ainsi l'illumination, en toute occasion, de l'une seule des 2 LED.

La tension d'alimentation des circuits intégrés est prise sur l'alimentation de l'amplificateur de puissance via une paire de résistances de 1,5 W (R15 et R16); 2 diodes zener, D2 et D4 en effectuent la stabilisation. Il est possible, de par la faible consommation de courant de IC1, de maintenir le courant zener à une valeur très faible, à savoir de 2,5 mA environ. Cette intensité est suffisante pour garantir un bon fonctionnement de l'ensemble et permet aussi de limiter la dissipation des résistances R15 et R16 à une valeur supportable même dans le cas d'étages de sortie à tensions d'alimentation élevées.

Une platine compacte

Les dimensions de la platine dessinée pour ce montage, dont on retrouve en **figure 2** la représentation de la sérigraphie de l'implantation des composants, sont suffisamment compactes pour en permettre la mise en place dans n'importe quel amplificateur. Nous avons opté pour un concept «mono» réalisé en double exemplaire, de sorte que l'on peut également utiliser le montage avec des blocs monophoniques ou avec des étages de sortie stéréophoniques à alimentations séparées –de cette façon, l'indication reste correcte même lorsque la tension d'alimentation de l'un des canaux s'effondre sensiblement lors d'une surmodulation temporaire.

La platine représentée en figure 2 comporte donc 2 blocs «mono». Si l'on en envisage l'utilisation sur un amplificateur stéréophonique on pourra éventuellement, en fonction de l'espace disponible, ne pas procéder à une séparation physique des 2 platines.

La réalisation de la platine n'est pas délicate; elle demande cependant un certain soin sachant qu'un nombre important de composants est implanté verticalement.

On commencera la mise en place des composants par celle des 3 ponts de câblage présents entre les circuits intégrés IC1 et IC2, ce qui nous évitera de les oublier à un stade ultérieur de la réalisation. On pourra utiliser des supports de bonne qualité pour IC1 et IC2, bien que cela ne soit pas indispensable. Les résistances de puissance R13 à R16 seront montées à une certaine distance du circuit imprimé de manière à permettre une meilleure dissipation de la chaleur qu'elles produisent.

L'interconnexion à l'étage de puissance se fait par l'intermédiaire de 3

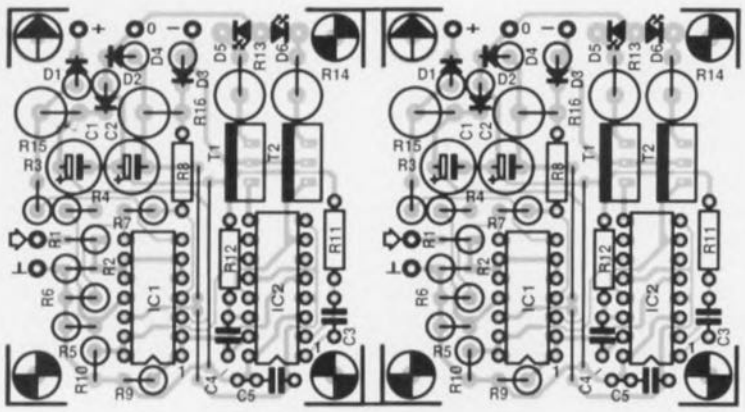


Figure 2. Nous avons gardé à la platine du témoin d'écrêtage, dont on retrouve ici la représentation de la sérigraphie de l'implantation des composants, des dimensions compactes, ce qui ne l'empêche pas de recevoir tous les composants représentés sur le schéma de la figure 1.

Liste des composants

Résistances :

R1 = 6kΩ04 1%
R2 = 1 kΩ 1%
R3,R8 = 60kΩ4 1%
R4,R7 = 2kΩ87 1%
R5,R6 = 7kΩ15 1%
R9,R10 = 150 kΩ
R11,R12 = 10 MΩ
R13,R14 = 1 kΩ 1W
R15,R16 = 4kΩ7 1W5

Condensateurs :

C1,C2 = 47 µF/25 V radial
C3 à C5 = 100 nF

Semi-conducteurs :

D1,D3 = diode zener 2V7/400 mW
D2,D4 = diode zener 15V/1 500 mW
D5 = LED verte
D6 = LED rouge
T1,T2 = BD240
IC1 = LM339
IC2 = 4538

fils reliés à l'alimentation (aux «+», «0» et «−» des condensateurs d'alimentation par exemple, solution bien souvent la plus pratique, vu que ces composants sont en règle générale les plus accessibles. On établit ensuite une liaison entre chaque sortie pour enceinte de l'amplificateur (en aval du relais de sortie) et chaque entrée de la platine du **témoin d'écrêtage**. On utilisera pour cette liaison un morceau de câble blindé. Seule l'extrémité de la gaine de blindage située du côté de la platine du **témoin d'écrêtage** sera reliée au point de masse présent à cet effet à proximité de l'entrée. Attention à ne jamais relier et le zéro et la connexion de masse de la platine du **témoin d'écrêtage** avec les points correspondants de l'amplificateur car cette double connexion ne manquerait pas de créer une boucle de masse.

La platine dotée de ses LED pourra être placée directement derrière la face avant, de manière à faire passer les 2 LED par 2 orifices percés à cette intention. On prévoira l'utilisation de picots pour la connexion des LED sachant qu'il n'est prévu que 3 points : 2 × cathode et 1 × anode commune. Si l'espace disponible ne permet pas cette solution, on pourra placer le circuit imprimé à un autre endroit et relier les 2 LED montées dans la face avant à la platine à l'aide de 3 morceaux de fil de câblage souple. La longueur de cette liaison et la technique utilisée ne sont pas critiques. La photographie de la **figure 3** donne un exemple de mise en oeuvre de la première technique.

Ceci termine la réalisation du montage. Comme nous le disions dans l'introduction, il n'y a pas de réglage à effectuer. Vous aurez vite fait de constater à l'usage combien faible est la différence entre une pleine modulation et un niveau situé 3 dB plus bas. Cette LED −3 dB s'avérera alors un auxiliaire précieux vous signifiant qu'il vous reste une certaine marge de puissance vers le haut.

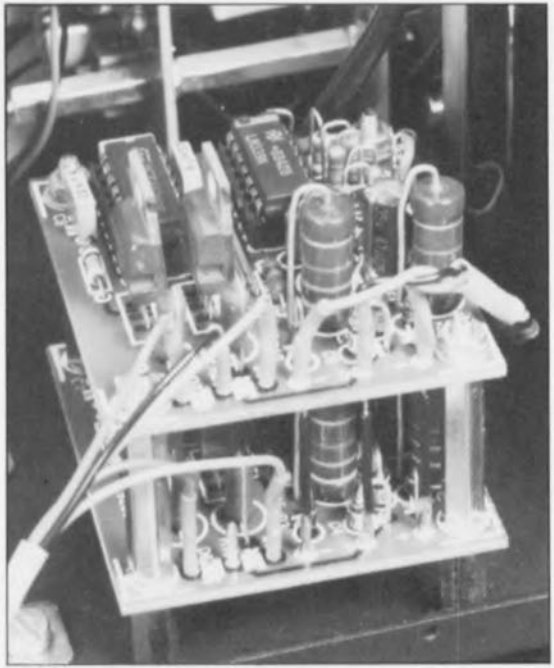
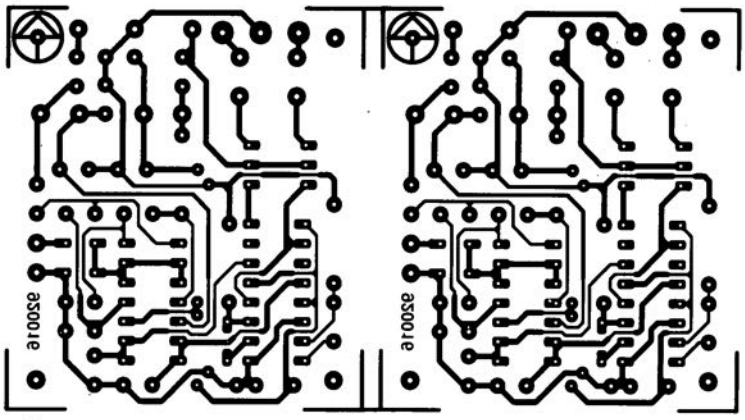


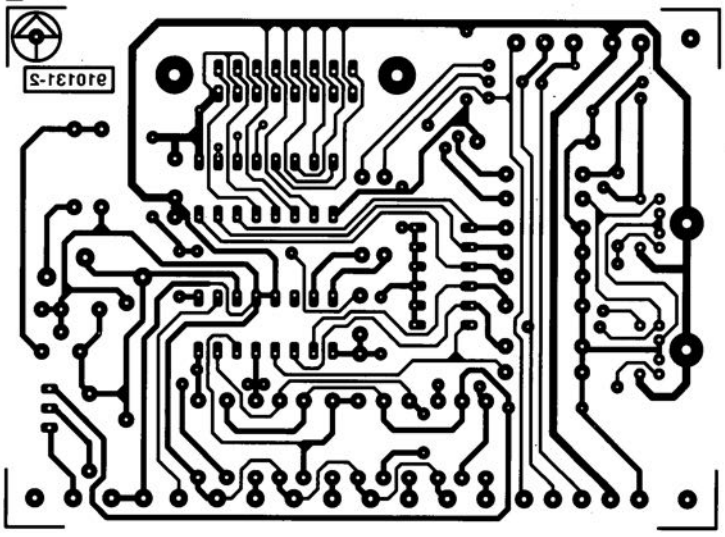
Figure 3. Exemple de montage du témoin d'écrêtage sur un amplificateur de puissance. Rien n'interdit cependant de monter les LED dans la face avant et de mettre la platine ailleurs dans le coffret de l'ampli.

1 témoin d'écrêtage
2 convertisseur A/N-N/A pour I²C
3 convertisseur RS-232-A/N

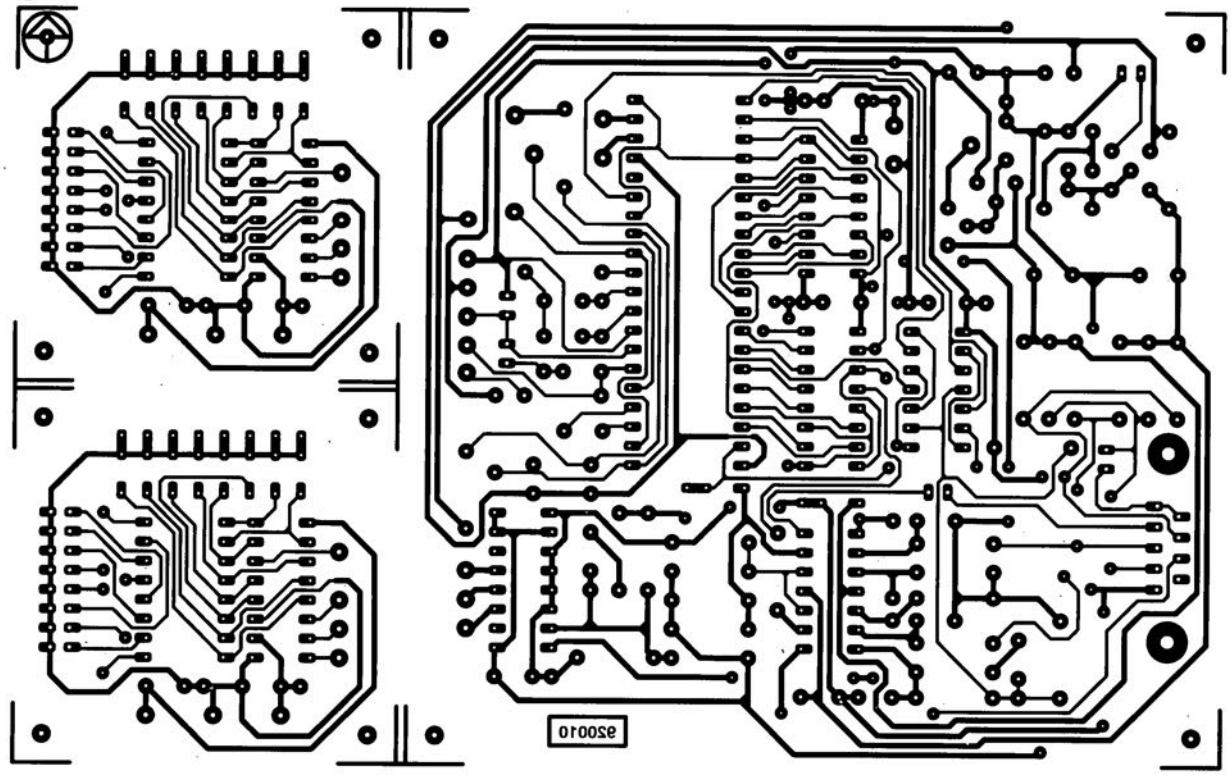
1



2

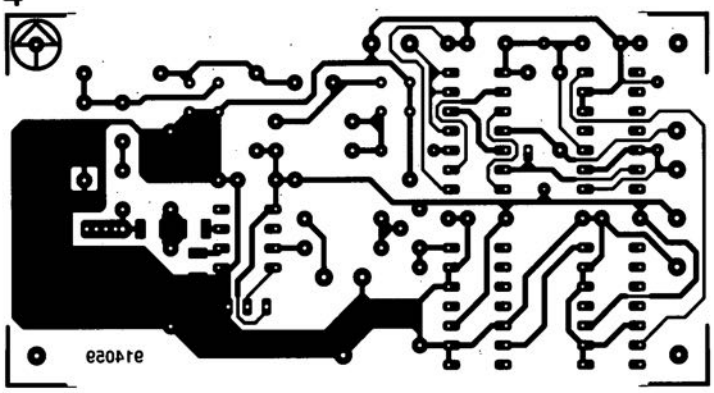


3

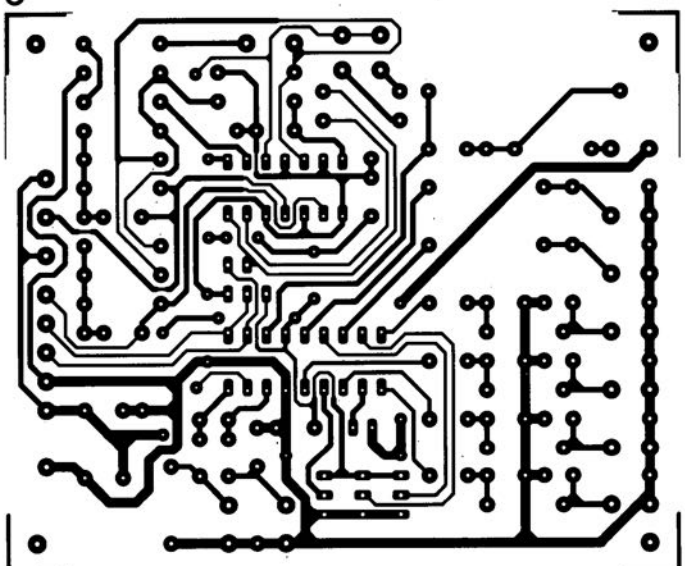


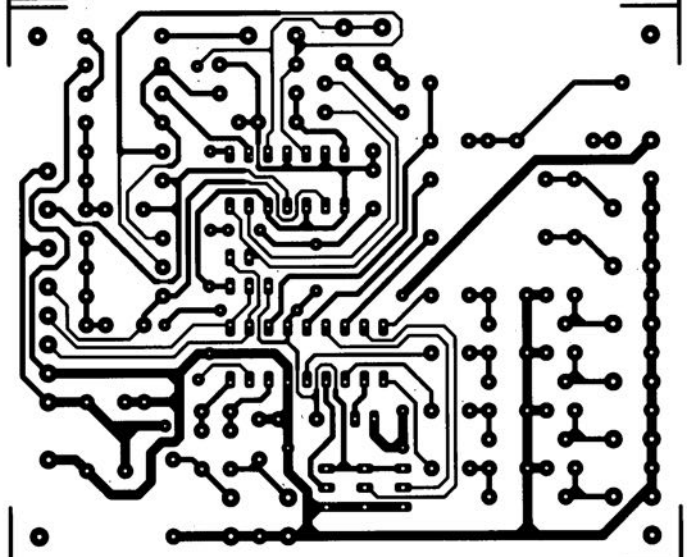
4 prédiviseur 1,3 GHz
5 commutateur audio/vidéo électronique
6 émulateur de 8751 : côté composants
7 émulateur de 8751 : côté pistes

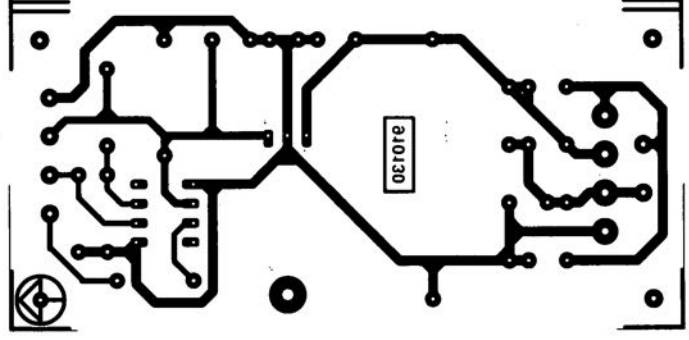
4



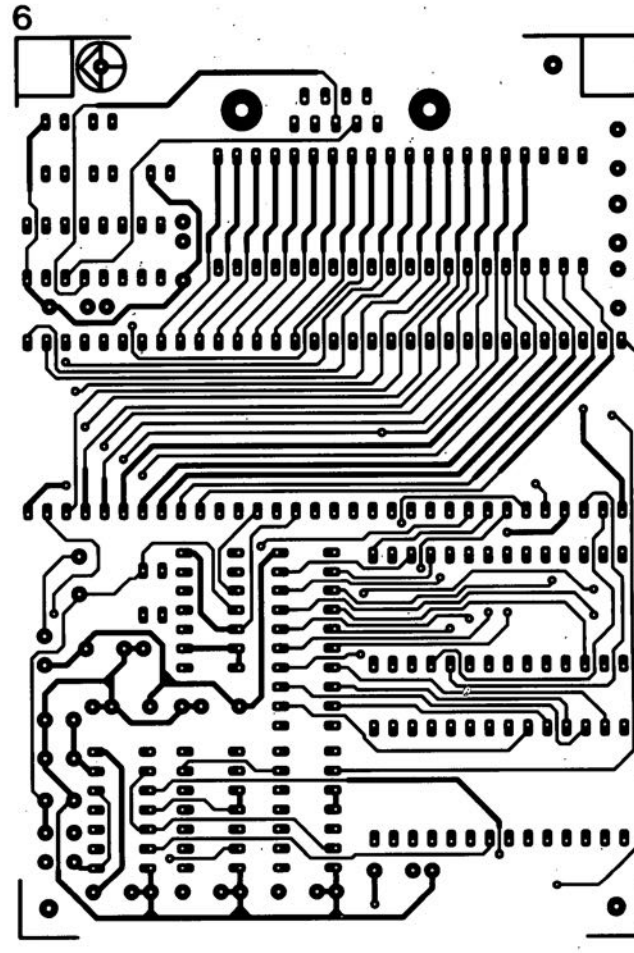
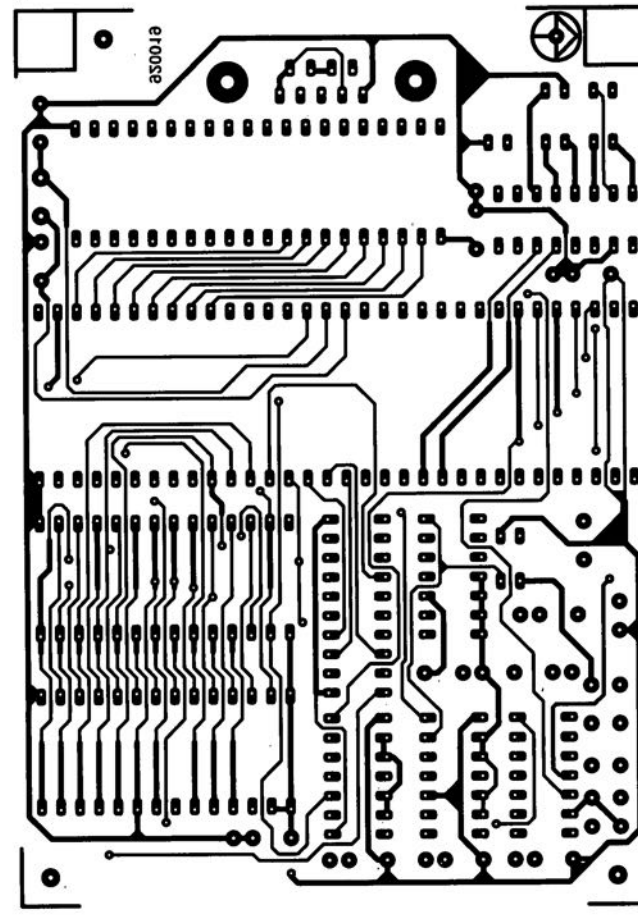
5







6



7