



**Pupitre
d'éclairages**



**Une
BasicCard
pour GSM**

**Intercepteur de
données pour
Märklin Digital**

**Carte
d'évaluation
pour MC16C/62**

**À la découverte
du PSN**



SPÉCIAL INTERFACES

- Disque « Compact Flash »
- Interface I²C pour RCX (Lego)
- Interface série pour puces 1-Wire



Cours

« Microcontrôleurs »

4^{ème} partie : le compilateur C READS51

Burkhard Kainka

Dès lors que l'on s'intéresse sérieusement aux microcontrôleurs on se trouve inévitablement confronté, un jour ou un autre, à C, un langage de haut niveau. Nous allons, dans le cas présent, utiliser le compilateur C READS51 de la société Rigel.

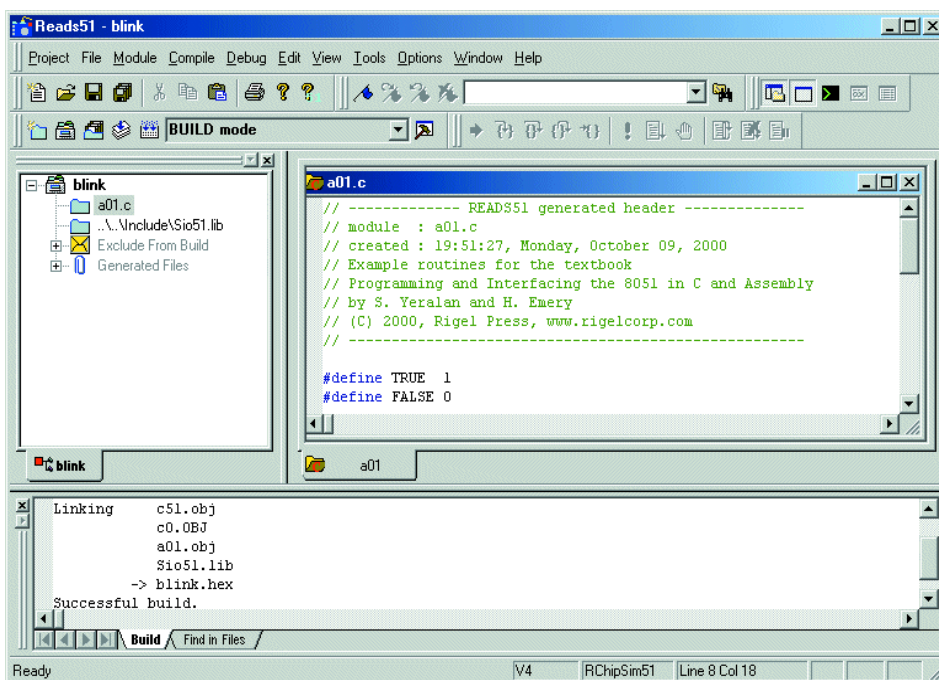


Figure 1. READS51 en action.

Nous avons, jusqu'à présent, utilisé, dans notre cours « Microcontrôleurs », 2 langages de programmation, à savoir l'assembleur et Basic-52. Nous allons maintenant travailler avec C, un compilateur que l'on peut télécharger sur le site Internet de Rigel à l'adresse : www.rigelcorp.com/8051soft.htm

où vous trouverez un lien destiné spécialement aux lecteurs de ce magazine. Les fichiers à télécharger sont [SetupReads51.exe](#) et [Reads51.pdf](#) (ce dernier pour en savoir plus au sujet de ce compilateur). La tâche d'un compilateur C est de

convertir un texte dit code-source en du code machine pur, ce qui n'est pas la fonction d'un interpréteur BASIC qui se contente de créer un code intermédiaire qu'il faudra, en son temps, lors de l'exécution du programme, interpréter et exécuter. Ceci explique que C soit beaucoup plus rapide que le BASIC.

Le langage C n'est pas né d'hier; il existe en outre pour de nombreux systèmes (lire microcontrôleurs et microprocesseurs). L'avantage majeur de C est que les programmes écrits dans ce langage sont relativement indépendants du matériel (hardware) utilisé. Il est possible ainsi de transporter vers d'autres systèmes le résultat de longues heures de labeur relativement facilement. Ce fut en 1988 déjà que l'on définit le langage ANSI-C (ANSI = *American National Standard Institut*) de manière à établir un standard commun. On a vu naître une petite variante, baptisée Small-C, destinée plus spécifiquement aux microcontrôleurs; cette mouture moins évoluée comporte un certain nombre de limitations que ne connaît pas le C, telles que l'absence de variables Real, mais que l'on peut utiliser avec les systèmes les plus minuscules imaginables.

Listage I. Premier programme d'exemple.

```
// ----- READS51 generated header -----
// module : a01.c
// created : 19:51:27, Monday, October 09, 2000
// Example routines for the textbook
// Programming and Interfacing the 8051 in C and
// Assembly
// by S. Yeralan and H. Emery
// (C) 2000, Rigel Press, www.rigelcorp.com
// -----

#define TRUE 1
#define FALSE 0

#include <sfr51.h> // P1_0 is defined here
// prototypes
#include <Sio51.h>

main(){
int n;

// -- initialize serial port (9600 Baud) --
InitSerialPort0(DEF_SIO_MODE);
//DEF_SIO_MODE is defined in
<Sio51.h>
putc('\n');

// endless loop
while(TRUE)
{
P1_0=0; // LED on
putc('+');
for(n=0; n<10000; n++); // waste some cycles
P1_0=1; // LED off
putc('0');
for(n=0; n<10000; n++); // waste some cycles
}
}
```

Il est possible de trouver sur Internet diverses versions de compilateurs pour Small-C très bien faites. Nous avons cependant choisi, pour ce cours, d'utiliser READS51 parce que nous considérons qu'il convient plus particulièrement à ceux qui prendraient le train-C en marche et en raison de son interface utilisateur confortable.

La société Rigel a conçu READS51 spécialement à l'intention de ses propres cartes à microcontrôleurs destinés à l'enseignement sous toutes ses formes. Rigel autorise l'utilisation de ce produit à tout un chacun tant qu'il s'agit d'une utilisation « privée » ou dans le cadre de l'enseignement. Elektor a obtenu l'aimable autorisation – nous n'en

voulons pour preuve la présence sur leur site d'un lien Elektor qui renvoie à ce logiciel– d'utiliser ce compilateur pour notre cours « Microcontrôleurs ». Tout lecteur intéressé pourra télécharger de ce fait READS51 depuis le site Internet de Rigel et installer ce programme. Notons que l'on trouvera à cet endroit d'autres outils, très intéressants eux aussi. Il est évident, il s'agit d'une firme américaine, que le programme et les outils sont en anglais. Il ne faut pas vous attendre à trouver, d'ici peu, une version dans une autre langue vu qu'une traduction dans une autre langue n'a guère d'intérêt.

Tous les exemples de notre cours « Microcontrôleurs » de même que les commentaires sont de ce fait en

langue de Shakespeare. L'universalité de l'anglais facilite les développements et les échanges de programmes.

La meilleure façon de se lancer à l'eau de READS51 est d'utiliser l'un des exemples proposés dans le cadre de cet article. On a vite fait de charger le projet blink par le biais du menu déroulant *Project/OpenProject*. Il suffit de cliquer sur le code-source du module principal a01.c pour le voir apparaître dans la fenêtre de l'éditeur.

Les programmes C comportent toujours une fonction principale main() qui sera exécuté lors du lancement du programme. Au premier abord, l'exemple blink ne remplit que la fonction qu'il évoque. En fait, il utilise d'autres fonctions présentes dans le module Sio51.h et servant à la commande de l'interface série du microcontrôleur. Dans le cas présent, l'interface est paramétrée à 9 600 bauds à une fréquence de quartz de 11,059 MHz, ce qui est très exactement le cas de notre carte à 89S8252 Flash utilisée dans le cadre de ce cours.

Un débutant en C pourra trouver la façon d'écrire étonnante. Voyons-en quelques détails :

```
define TRUE 1
```

Définition d'une constante (à chaque fois que l'on rencontre TRUE, cette constante est remplacée par la valeur « 1 »).

```
include <sfr51.h>
```

Inclusion d'un fichier d'en-tête avec définitions.

```
main()
```

```
{
...
}
```

On a création de la fonction principale main. Tous les paramètres de ces fonctions se trouvent dans un bloc d'instructions délimité par des accolades ({}).

```
int n;
```

Déclaration d'une variable n du type entier (integer) dont la plage des valeurs licites va de -32768 à +32767. Le point-virgule (;) termine cette ligne.

```
InitSerialPort0
```

```
(DEF_SIO_MODE);
```

On appelle une fonction avec paramètre de transfert, il s'agit ici de la fonction d'initialisation de l'interface série tirée du module Sio51.h.

```
// endless loop
```

Un commentaire qui facilite la lisibilité du programme mais qui ne sera pas converti lors

ATMELISP, un nouvel outil de transfert

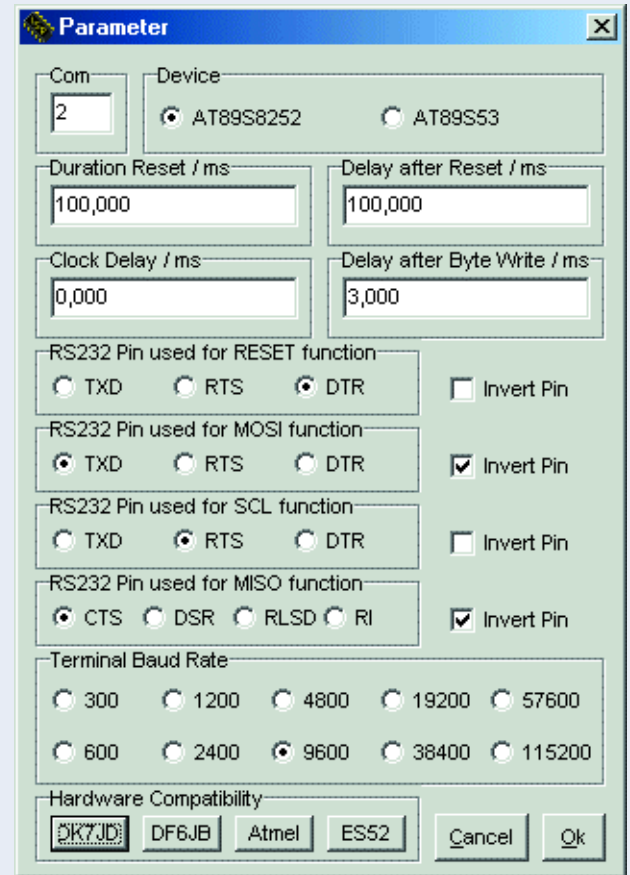
Le programme de téléchargement MicroFlash.exe sans prétentions utilisé pour le transfert (download) de programmes vers la carte 89S8252 Flash ne connaît que 2 ports sériels. COM1 et COM2; il présente en outre l'inconvénient de ne pas signaler une opération de transfert ayant eu lieu correctement, ce qui a posé quelques problèmes à certains utilisateurs. Les lecteurs d'Elektor ne sont pas connus pour leur passivité. Un lecteur allemand, Mr Ulrich Bangert (DF6JB) a développé un nouveau programme notablement plus sophistiqué qu'il a baptisé ATMELISP (pour Atmel ISP = In Site Programming) qui permet une programmation Flash avec nombre de systèmes différents. Ce programme supporte, outre le Starterkit d'Atmel et une mono-carte de l'auteur, également le système Elektor (carte 89S8252 Flash) et le Modul-Bus (ES52-Flash) connu de nos lecteurs d'outre-Rhin. On pourra télécharger ce programme (en anglais) depuis le site Elektor sous la dénomination de **EPS010208-13**.

Après décompression du fichier .zip on se trouve en présence d'un programme exécutable .exe et d'un fichier d'aide (en anglais) de près de 1,6 Moctets. L'écran d'accueil (**figure A**) est extrêmement compact ce qui permet de le caser sans problème à côté d'autres applications ouvertes sur l'écran. Ce n'est que lorsque l'on choisit une fonction du programme que l'on voit apparaître une fenêtre plus encombrante. La première étape consiste bien entendu à définir l'interface série à laquelle est connecté le système et un certain nombre d'autres paramètres importants. Un clic sur le bouton « DK7JD » (code d'appel du radio-amateur qu'est l'auteur B. Kainka) se traduit par un paramétrage automatique des lignes sérielles (RS-232) telles que les utilise notre platine. On constate qu'il est extrêmement simple, à l'aide d'ATMELISP, de programmer une platine que l'on aura développée soi-même et utilisant le même processeur (89S8252). Ce processus se traduit par la sélection de 3 lignes auxquelles on attribue les fonctions requises. Il peut se faire qu'il soit nécessaire, d'adapter les durées de temporisation. Nous avons constaté, avec un PC relativement lent – si si, il y en a encore... – qu'il fallait faire passer le paramétrage de la temporisation Clock Delay de 0 à 0,01 ms. La **figure B** montre comment sélectionner les paramètres. Nous allons illustrer la suite des événements par un exemple concret. Nous allons charger le programme de notre premier exemple, code généré par le compilateur C dans le ROM Flash du microcontrôleur. ATMELISP est capable de lire les fichiers de format binaire et de format Intel-Hex. Nous allons transférer ici le fichier Blink.hex. Il est judicieux de jeter un bref coup d'oeil à l'éditeur de texte intégré (**figure C**)

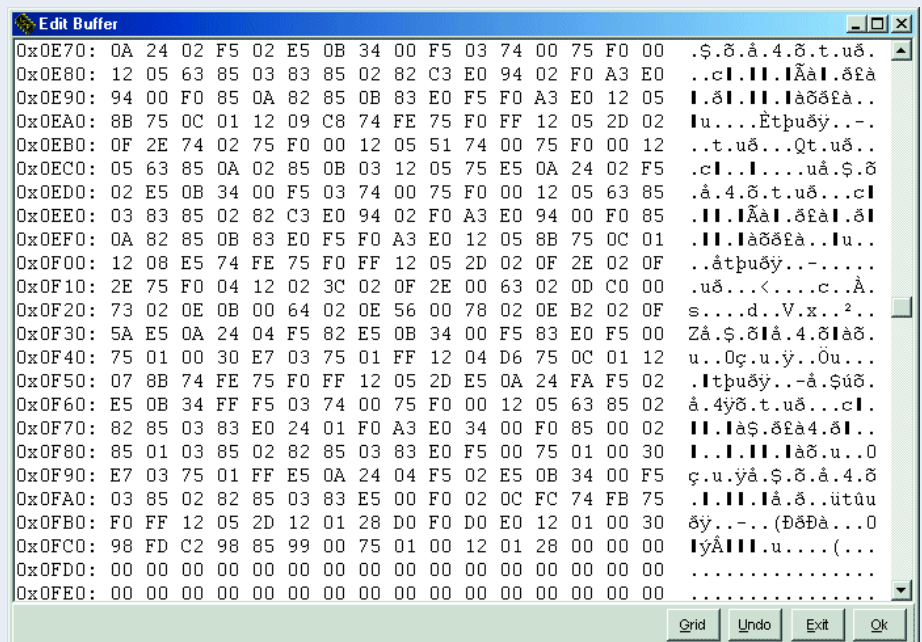
A



B



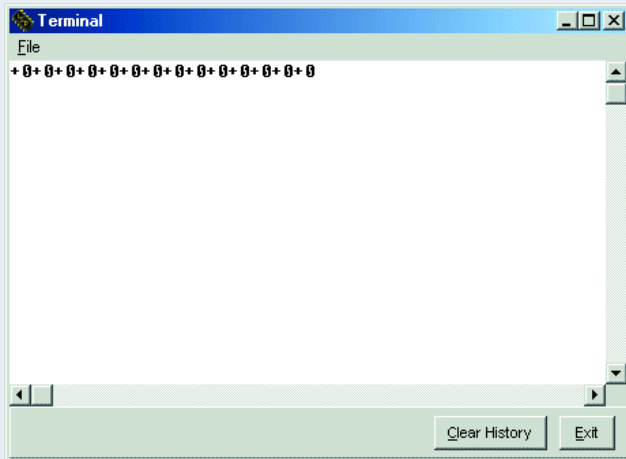
C



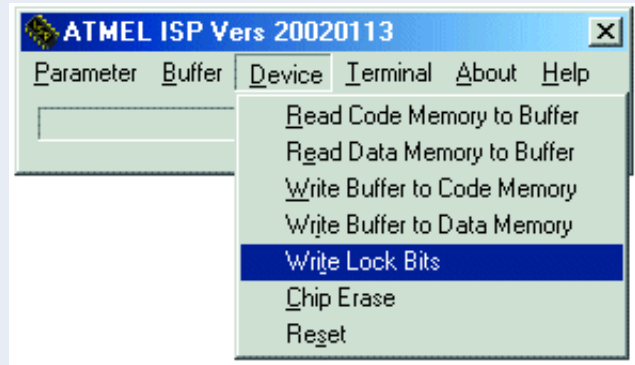
qui donne le contenu et la taille du programme.

On optera, pour la programmation la fonction

« Device/Write Buffer to Code Memory ». Il faut faire attention ici à ne pas confondre « Code Memory » (mémoire de programme) et « Data Memory » (mémoire de données), car il s'agit dans le premier cas du domaine de données de 2 Koctets en EEPROM que possède le microcontrôleur. Il est possible de programmer et de lire ces deux mémoires. Notons en outre qu'il est possible de doter un composant de bits de protection (lock-bit) en vue d'interdire un



D



examen du contenu du programme (**figure D**). Attention à l'utilisation des bits de protection : si l'on a positionné les 3 bits, le composant n'est plus accessible pour une programmation sérielle ultérieure ! Il n'est plus non plus possible d'effacer l'ensemble de la mémoire du composant par el biais de notre programme. La seule façon de se sortir de ce guêpier est de faire appel à un programmeur parallèle.

ATMELISP comporte également un terminal (**figure E**), que l'on pourra mettre à contribution pour visualiser les données disponible en sortie lors de l'exécution de notre premier exemple. Il faudra pour cela déconnecter le câble sériel et enficher son connecteur dans l'autre embase d'interface présente sur notre carte.

E

du processus de compilation.

```
while(TRUE)
```

$$\left\{ \begin{array}{c} \vdots \\ \vdots \end{array} \right\}$$

On a création d'une boucle. On pour-

rait, à la place de TRUE pour une boucle sans fin, trouver une autre condition qui entraînerait l'exécution de la boucle. Toutes les instructions devant être exécutée dans la boucle sont à nouveau délimitées par des accolades.

P1 0=0;

Une attribution. On donne ici la valeur 0 à la variable de bits P1_0.

```
putc( '+' );
```

Émission de texte par le biais de l'interface sérielle. `putc` a été défini dans le fichier *Sio51.h*. On a transmission d'un caractère, que ce soit une variable ou un constante de type `char` (`char = character = caractère`).

```
for(n=0; n<10000; n++);
```

On a création d'une boucle de comptage que l'on écrirait en BASIC de la façon suivante : For n=1 to 10000: Next n. La boucle ne comporte pas, dans le cas présent, d'instructions, situation que rend le point-virgule. On pourrait trouver à cet endroit un bloc d'instructions qui serait pris entre 2 accolades. Peu importe que l'on ait ou non saisi toutes les finesses d'un programme en C, ce qui est vraiment intéressant est de savoir si le dit programme tourne sur notre carte Flash. Il faudra commencer, pour cela, par le compiler, c'est-à-dire, nous le disions plus haut, traduit en langage machine. Cette opération s'effectue par le biais du point de menu *Compile/Build* ou plus simplement encore, par une action sur la touche « *F9* ». Le processus de compilation est relativement complexe vu que l'on commence par la conversion des différents modules-objet qui sont ensuite

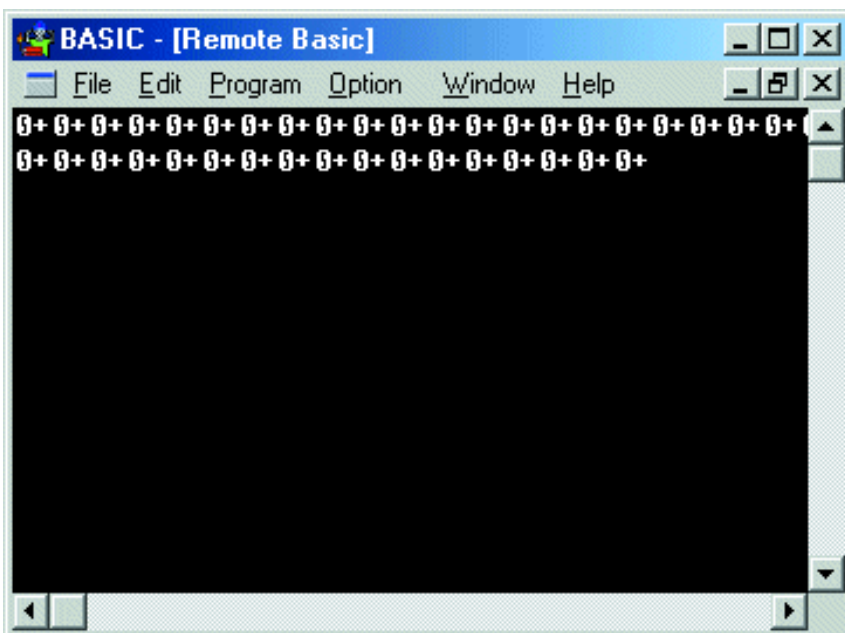


Figure 2. Visualisation de texte dans l'éditeur.

liés pour constituer un programme complet. Le résultat final de cette opération est un fichier, `Blink.Hex` au format Intel-Hex. Il se trouve dans le répertoire de projet `/work/blink`.

Il est très facile maintenant de transférer le fichier Intel-Hex du projet vers la carte à l'aide du programme MicroFlash. On remarquera, lors du transfert vers la mémoire Flash du microcontrôleur que ce programme est, bien que le code-source soit facile à survoler, une fois compilé, relativement long puisqu'il fait près de 4 Koctets. Cela tient au fait qu'il intègre le module `C51.obj`. Ce module comporte toutes les fonctions mise à disposition par `READS51`, y compris celles dont on n'a que faire pour l'application en question.

Il est temps, une fois le téléchargement terminé avec succès, de passer aux premiers essais. On connecte une LED dotée de sa résistance de limitation de courant entre la ligne de port `P1.0` et la ligne d'alimentation positive (`Vcc`). Surprise : la LED clignote ! Un amateur d'électronique traditionnelle ne manquera pas de remarquer qu'il aurait pu obtenir le même résultat à l'aide de 2 transistors seulement. Heureusement que notre programme est capable de plus. Il a en effet profité de l'occasion pour initialiser l'interface série que nous allons utiliser. Nous allons avoir besoin, pour savoir qu'il y a bien transfert de données, d'un programme de terminal.

La solution la plus simple consiste à utiliser le programme Basic-Terminal proposé dans le cadre de ce cours. Il restera à faire en sorte que les paramètres de transmission collent. Le programme en C travaille à 9 600 bauds, `Basic.exe` utilise normalement 19 200 bauds. Il est heureusement facile de modifier le paramétrage du taux de transmission de ce dernier programme. Pour ce faire, on ouvre le fichier `Basic.ini` à l'aide d'un éditeur de texte et on y ajoute la ligne : `Baud=9600`

Listage 2.

Contenu du fichier `.ini` modifié pour le Basic-Terminal.

```
[AHBASIC]
COM=2
Baud=9600
```

Le programme de terminal visualise les données transmises par le programme en C, à savoir une succession de caractères 0 et +, dont l'alternance restitue les états successifs de la ligne de sortie `P1.0` (cf. **figure 2**). C'est la fonction `putc` chargée de l'envoi, l'un

Listage 3. Programme pour des sorties de port rapides.

```
// ----- READS51 generated header -----
// module : C:\Rigel\Reads51\Work\Output\Output.c
// created : 12:33:17, Friday, November 09, 2001
// -----

#define TRUE 1
#define FALSE 0

#include <sfr51.h> // P1 is defined here

main(){
    unsigned char n;
    // endless loop
    while(TRUE)
    {
        for(n=0; n<256; n++)
        {
            P1=n;
        }
    }
}
```

Listage 4. Diviseur de fréquence par 20.

```
// ----- READS51 generated header -----
// module : C:\Rigel\Reads51\Work\Count\count.c
// created : 18:26:23, Monday, November 12, 2001
// -----

#include <sfr51.h>

void pulse(void){
    while(P1_0);
    while(!(P1_0));
}

main(){
    int n;
    n=0;
    while(1)
    {
        while (n<10)
        {
            pulse();
            n=n+1;
        }
        P1_1=1;
        while (n<20)
        {
            pulse();
            n=n+1;
        }
        P1_1=0;
        n=0;
    }
}
```

Tableau 1.

Langage	Mémoire	Durée de boucle	Autostart
Basic-52	8 K ROM, RAM	2500 μ s	avec EEPROM uniquement
READS51 C	>4 K ROM, RAM	125 μ s	oui
Assembler	< 1 K ROM	3 μ s	oui

après l'autre, de caractères, fonction appelée en cours de programme, qui est responsable de cet état de faits.

Sorties de port rapides

Il est temps maintenant, après ce premier exemple, d'écrire notre premier programme en C. Nous allons commencer par générer des sorties de port rapide de manière à pouvoir procéder à une comparaison de vitesse avec les autres langages de programmation utilisés jusqu'à présent.

La première étape consiste à définir un nouveau projet. Il faudra, sous le point de menu « *Project/New Project* », entrer un nom de projet. Nous allons utiliser le nom « *Output* ». READS51 crée alors un nouveau répertoire, « *Work/Output* ». On ouvre ensuite, sous le point de menu « *Module/Create Module* » un nouveau module auquel nous allons donner la dénomination de « *Output* ».

Le compilateur se doit de savoir quel est le projet qu'il lui faut convertir (compiler). Il faut pour cela exécuter la fonction « *Project/Set Project Active* » du menu « *Project* ». Si l'on oublait d'exécuter cette fonction on aura compilation du dernier projet auquel on aura travaillé. On pourra ensuite, par le biais du programme MicroFlash transférer vers la mémoire de programme du processeur le code de programme « *Output.hex* » né de la compilation. Il suffira, pour vérification du fonctionnement du programme, de disposer d'un casque d'écoute ou d'un oscilloscope. On trouvera sur la broche P1.0 le signal présentant la fréquence la plus élevée. Elle est ici de 4 kHz, la durée de période étant, logiquement, de 250 μ s. Il faut au

programme 125 μ s pour chaque nouvelle sortie au niveau du port.

Contrairement à ce qui est le cas en BASIC, C connaît plusieurs types de variables. Si notre premier exemple utilise une variable N de type int, une variable de type entier dont la valeur peut aller de -32768 à +32767, notre second exemple fait appel à une variable de type unsigned char (caractère non-signé), cryptogramme recouvrant un octet. De nombreux essais ont prouvé que cette approche n'a aucune influence sensible sur la vitesse d'exécution du programme.

Il est facile de comparer à l'efficacité des différents langages de programmation (cf. **tableau 1**). Les critères les plus importants sont taille de mémoire nécessaire, vitesse et la capacité de produire un programme autonome (*stand alone*) tournant sur le microcontrôleur Flash.

Les programmes C utilisent bien la RAM du système, l'ensemble du code de programme se trouve lui cependant dans la ROM en Flash. Ceci explique que contrairement à un programme en Basic-52, un programme en C redémarre après coupure de la tension d'alimentation.

Diviseur de fréquence en C

Lorsque l'on cherche une tâche un peu plus compliquée que celles décrites jusqu'à présent, l'idée d'un diviseur de fréquence par 20 paraît une idée intéressante d'autant que ce programme a déjà été écrit en Basic-52. Une comparaison directe entre les 2 versions de ce programme peut aider à reconnaître les différences tant au niveau de la structure que du mode d'écriture. Nous avons baptisé notre programme du nom de projet « *Count* ».

Le listage illustre un avantage majeur du langage de programmation C. Le programmeur est forcé de respecter une approche structurée. Ceci ne peut qu'améliorer la lisibilité du programme. Il a été écrit une fonction « *pulse* » qui interrompt le déroulement du programme tant que l'on attend l'apparition sur la broche P1.0 du flanc montant suivant. Il est souvent fait appel à des fonctions par le biais desquelles on transmet une valeur pour en recevoir une autre en retour. La fonction « *pulse* » utilisée ici ne fournit pas de paramètre en retour (*void = vide*) et elle ne reçoit pas non plus de paramètre. C ne fait pas la distinction entre fonction et procédure que connaissent le Pascal et Delphi.

La variable de bit P1_0 fournit soit un « 1 » soit un « 0 ». Tant que la condition requise par *while* est vraie (= 1), l'exécution de la boucle se poursuit.

La seconde boucle *while* reprend la même condition, avec une négation cette fois (ce qui est exprimé par un point d'exclamation, *!(P1_0)*). On quitte de ce fait la seconde boucle lors du basculement du niveau d'entrée de « 0 » vers « 1 », ce qui correspond à un flanc montant. Le programme principal appelle la fonction *pulse* en 2 endroits, une fois pour *n = 0* à 9 et une autre fois pour *n = 10* à 19.

À nouveau la question cruciale qui se pose est la suivante : quelle est la fréquence la plus élevée applicable à l'entrée avant que n'apparaissent des erreurs de comptage. Nous avons testé le système par la connexion d'un générateur de fonctions à la broche P1.0 et un oscilloscope à la broche P1.1. Les tests effectués nous ont fourni une fréquence limite de 3 kHz. Qu'on se souvienne, en Basic-51 nous avions atteint 50 Hz seulement. On peut espérer atteindre, en assembleur, jusqu'à 100 kHz.

(010208-5)

Littérature :

Sencer Yeralan/Helen Emery
Programming and Interfacing The 8051 Microcontroller in C and Assembly
 Rigel Press 2000

Kris Jamsa
Toute la puissance de C++ en 40 leçons
 Publitrone 1998

Une BasicCard Pour téléphones portables GSM

Découvrez les secrets de votre téléphone GSM

Patrick Gueulle

Une carte à puce du type BasicCard programmée constitue le « nerf » de cet article. Nous allons voir comment utiliser cet « émulateur de carte SIM » pour tester les principales fonctions accessibles par le « menu » normal et le « menu » de maintenance de certains portables GSM.

La carte à puce « SIM » (*Subscriber's Identification Module*) est le véritable « sésame » indispensable à l'utilisation de tout téléphone portable GSM. Contenant tout à la fois les identifiants secrets et les données personnelles de l'utilisateur, elle est émise par un opérateur de téléphonie mobile, qui la confie à son client mais en conserve généralement la propriété.

Une carte à puce à système d'exploitation ouvert, telle que la BasicCard, peut toutefois être transformée en carte SIM « de développement », autorisant une foule d'expérimentations en toute indépendance vis-à-vis des opérateurs.

La BasicCard « Professionnelle »

Depuis le lancement, en 1998, de la BasicCard « compact », ZeitControl a progressivement introduit des versions plus puissantes, dites « enhanced », puis tout récemment « professional ».

Avec la ZC 4.1, c'est une étape majeure qui vient d'être franchie, puisqu'il s'agit de la toute première BasicCard supportant le protocole « T=0 ».

Rappelons, en effet, que toutes les versions précédentes de la BasicCard fonctionnaient exclusivement en protocole « T=1 », particulièrement populaire en Allemagne.

Comme la spécification GSM 11.11 de l'ETSI impose le protocole « T=0 », d'ailleurs de très



loin le plus courant, pour les cartes SIM, il n'était jusqu'à présent pas possible de faire fonctionner une BasicCard dans un téléphone portable.

Il faut savoir que la BasicCard « professional » est équipée de l'une des « puces » les plus performantes d'ATMEL, la AT90SC3232C.

Avec 32 Koctets de mémoire Flash pour le système d'exploitation,

32 Koctets d'EEPROM pour les programmes « utilisateur », 1 Koctet de RAM, et un processeur RISC « AVR » secondé par un coprocesseur cryptographique, ce composant parfaitement adapté à la production de cartes SIM « Phase 2+ » et « SIM Toolkit », n'est guère devancé que par sa variante AT90SC6464C, prévue notamment pour l'UMTS.

C'est dire qu'il s'agit d'une puce de

Tableau 1.
Le « dictionnaire » des
commandes normalisées
GSM 11.11.

20h :	verify CHV
24h :	change CHV
26h :	disable CHV
28h :	enable CHV
C0h :	get response
32h :	increase
04h :	invalidate
B0h :	read binary
B2h :	read record
44h :	rehabilitate
88h :	run GSM algorithm
A2h :	seek
A4h :	select
FAh :	sleep
F2h :	status
2Ch :	unlock CHV
D6h :	update binary
DCh :	update record
10h :	terminal profile
C2h :	envelope
12h :	fetch
14h :	terminal response

haut de gamme, dont l'originalité majeure est d'être entièrement réalisée en technologie « Flash EEPROM ».

Le système d'exploitation embarqué est ainsi téléchargé lors de la personnalisation de la carte, et non plus « masqué » dans une zone de ROM lors de la fabrication de la puce, ce qui est infiniment plus souple.

Dans une carte SIM conventionnelle, c'est généralement un système d'exploitation dédié qui est programmé dans cette zone.

La partie EEPROM accueille alors le système de fichiers normalisé GSM 11.11, et s'il s'agit d'une SIM « Phase 2+ », d'éventuelles applications « SIM Toolkit » (souvent des « applets » Java) implantées par l'opérateur ou par un fournisseur de services (banque, par exemple).

Dans une BasicCard, le système d'exploitation est de type « ouvert », tout comme celui des Javacard, Multos, et autres Windows for smart cards.

Il s'agit, en l'occurrence, d'un interpréteur permettant d'écrire les programmes applicatifs dans ce langage si populaire qu'est le BASIC, que l'on dirait d'ailleurs « taillé sur mesures » pour les cartes à puce.

Une fois mises au point à l'aide d'un outil de développement performant mais néanmoins gratuit

(<http://www.basiscard.com>), les applications sont téléchargées dans la zone EEPROM.

Dans le présent projet, un programme BASIC de près de 450 lignes occupe environ 17% de la zone EEPROM, le solde étant partagé entre un système de fichiers compatible GSM 11.11 et un gros fichier « log », dans lequel la carte pourra enregistrer, sur demande, l'historique des commandes qu'elle recevra de l'extérieur.

Une carte SIM de développement

La vocation d'une carte SIM de développement n'est évidemment pas de téléphoner (encore qu'en principe, elle puisse tout de même permettre d'appeler le 112), mais plutôt de rendre possibles toutes les manipulations normalement irréalisables avec une carte émise par un opérateur.

Dans une « vraie » carte SIM, en effet, les fichiers les plus « sensibles » sont protégés, en écriture et parfois même en lecture, pas des codes « administrateur » connus seulement de l'opérateur qui l'a émise.

Dans le cadre de ce projet, tous les codes confidentiels (y compris les « PIN » de l'utilisateur) sont désactivés une bonne fois pour toutes, ce qui libère à 100% l'accès à l'ensemble des fichiers existants.

On se gardera bien d'en déduire qu'il suffirait de recopier les identifiants d'une carte SIM valide, pour en créer un « clone » utilisable.

Deux « garde-fous » infranchissables évitent en effet que notre projet ne puisse être détourné vers des applications frauduleuses :

– Tout d'abord, l'algorithme cryptographique (A3/A8) assurant l'authentification des cartes SIM au travers des réseaux GSM est ici purement factice, et il ne saurait être question de le modifier sans avoir accès au code source.

– De plus, cloner une carte supposerait que l'on recopie, à l'identique, la clef secrète qui réside dans une zone de l'original rigoureusement verrouillée en lecture.

En revanche, l'écriture de valeurs bien particulières dans certains fichiers peut permettre, par exemple, d'utiliser pour nos manipulations des

téléphones verrouillés (« Simlockés ») par l'opérateur qui en a sans doute « subventionné » la fourniture, et qui ne peuvent normalement fonctionner qu'avec leur carte SIM d'origine.

De même, notre carte donnera souvent accès aux menus « cachés » des téléphones (notamment de marque Motorola), réservés en principe aux seuls techniciens habilités.

Il n'en faut pas davantage, par exemple, pour obtenir l'affichage d'un code de sécurité malencontreusement oublié, ou pour activer des fonctions de repérage des relais des différents réseaux disponibles en un lieu donné.

Mais il y a mieux : la carte contient en effet suffisamment de mémoire EEPROM pour pouvoir enregistrer, dans un fichier interne, toutes les commandes que lui envoie le téléphone, ou n'importe quel autre « terminal » dans lequel on voudra bien l'introduire (par exemple un PC exécutant une application de gestion de cartes SIM).

Téléchargé ensuite par un utilitaire approprié, ce fichier « LOG » pourra être analysé à loisir, à des fins didactiques ou de diagnostic. Enfin, notre carte est compatible « Proactive SIM », ce qui signifie que, contrairement à une simple carte SIM de phase 1 ou 2, elle ne se contente pas d'exécuter les ordres que lui adresse le téléphone : elle peut, de sa propre initiative, prendre le contrôle de celui-ci, et notamment de son afficheur ou de son écouteur.

Une petite application indépendante a ainsi pu être implantée, qui ne s'exécute que si le téléphone est compatible « phase 2+ ». Moyennant quoi, l'écran affiche, au bout d'une minute environ, un mot hexadécimal (le « *Terminal profile* ») dont le décodage fournit un diagnostic détaillé des possibilités « SIM Toolkit » du mobile.

Par la suite, la carte s'immiscera toutes les minutes dans le fonctionnement du téléphone (sans pour autant le perturber le moins du monde) afin de lui faire afficher un court instant « BasicCard » en diffusant une petite mélodie.

Entrons dans le détail

Notre carte SIM de développement reconnaît, et c'est la moindre des choses, la totalité des commandes normalisées (voir **tableau 1**) qu'un mobile GSM est susceptible de lui envoyer dans l'état actuel de la normalisation (phases 1, 2, et 2+).

Reconnaître ne signifie pas nécessairement exécuter, en particulier lorsqu'il s'agit de commandes liées aux codes confidentiels.

Même si la carte ignore purement et simplement telle ou telle commande, elle y répondra toujours par un compte-rendu permettant la

poursuite de la « session GSM » en cours.

Si, par exemple, on demande l'activation du code PIN, la carte répondra « bonne exécution » (9000h) mais ne réclamera pas, pour autant, ce code par la suite.

De même, si on tente de modifier le PIN, la carte répondra que l'opération est exécutée, mais continuera imperturbablement à considérer n'importe quel code comme étant le bon. Ce stratagème permet, et c'était précisément le but recherché, de jouer librement avec tous les fichiers de la carte.

Cela étant, il est indispensable de préciser ici comment est structurée une commande GSM. Non pas tant pour en construire de toutes pièces (ce sera plutôt le rôle de logiciels spécialisés), mais pour pouvoir interpréter, le moment venu, le contenu de ce fameux fichier « log ».

Les cartes SIM fonctionnant en protocole « T=0 », toute commande destinée à la carte commence par cinq octets d'en-tête (*header*) : CLA INS P1 P2 LEN.

CLA est la « classe ISO » de la commande, toujours égale à A0h pour une carte SIM.

INS est un code opératoire appartenant à la liste du tableau 1.

P1 et P2 sont deux paramètres dont la signification varie d'un code opératoire à l'autre, et qui restent très souvent, par défaut, à 00h. LEN, enfin, indique la longueur du bloc de donnée qui suit, à moins qu'il n'y en ait pas (LEN = 00h).

Il faut distinguer les « commandes entrantes »

Tableau 2.

Les fichiers disponibles dans notre carte.

Répertoire racine :

2FE2 (ICCID)

Répertoire Télécom :

6F3A (ADN, Abbreviated Dialling Numbers)
6F3B (FDN, Fixed Dialling Numbers)
6F3C (SMS, Short Messages)
6F40 (MSISDN, Own Numbers)
6F42 (SMSP, Short Messages Service Parameters)
6F43 (SMSS, SMS Status)
6F44 (LND, Last Number Dialed)

Répertoire GSM ou DCS :

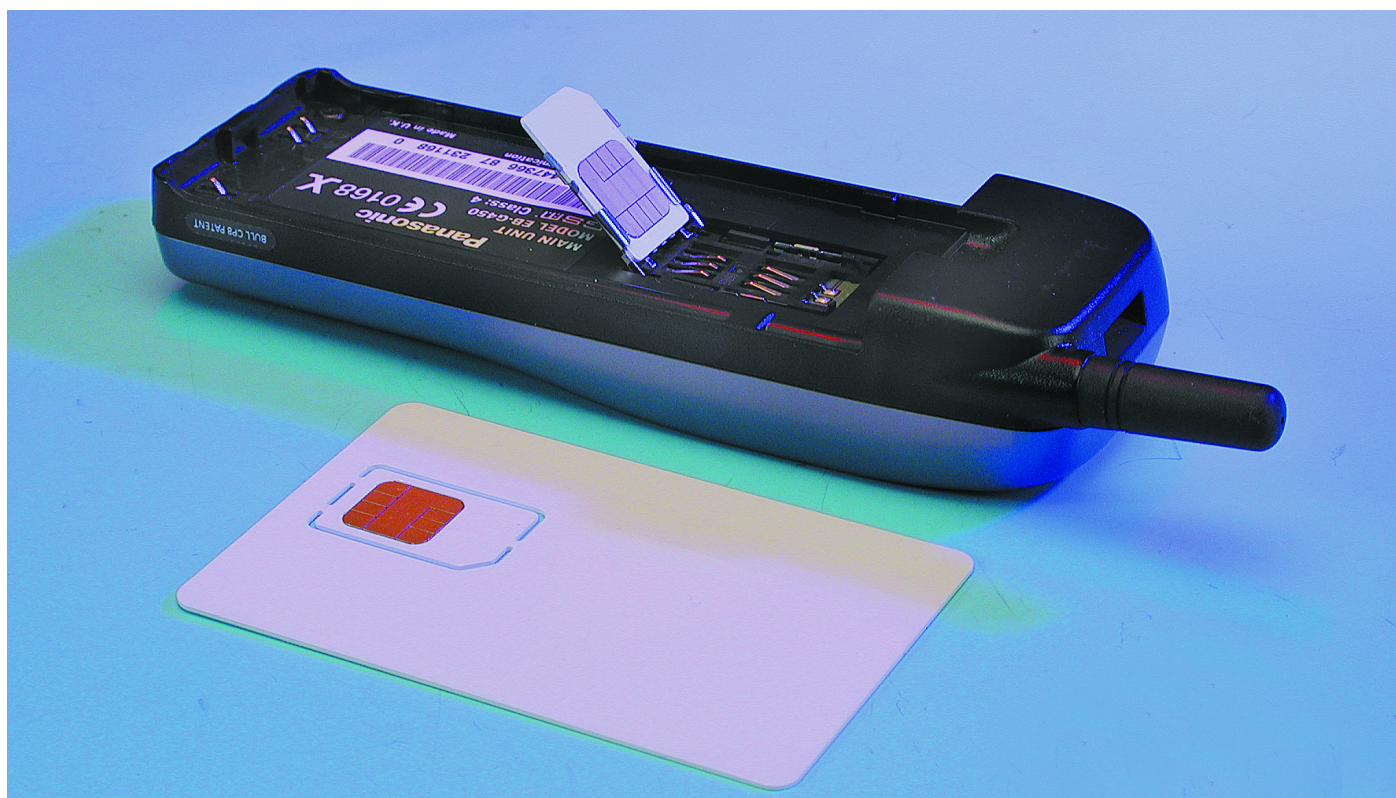
6F05 (LP, Language preference)
6F07 (IMSI, International Mobile Subscriber Identity)
6F20 (Kc, Ciphering Key)
6F30 (PLMN, Preferred PLMNs)
6F31 (HPLMN search period)
6F38 (SST, SIM Service Table)
6F3E (GID1, Group Identifier level 1)
6F3F (GID2, Group Identifier level 2)
6F74 (BCCH, Broadcast Control Channels)
6F78 (ACC, Access Control Class)
6F7B (FPLMN, Forbidden PLMNs)
6F7E (LOCI, Location Information)
6FAD (AD, Administrative Data)
6FAE (Phase)

(mobile vers carte), dont le bloc de données est émis à la suite de LEN, des « commandes sortantes » (carte vers mobile) où la carte est priée de renvoyer LEN octets de données.

Dans quasiment tous les cas, la carte

retourne au minimum deux octets de « compte-rendu » (SW1 et SW2) qui, lorsque tout s'est bien passé, sont respectivement 90h et 00h.

Lorsque la carte a des données à renvoyer en réponse à une com-



mande entrante, elle répond par SW1=9Fh, SW2 précisant alors le nombre d'octets qu'il faut venir chercher avec une commande « GET RESPONSE » (A0 C0 00 00 SW2).

C'est là une particularité importante du protocole « T=0 », de ne pas supporter les commandes à la fois entrantes et sortantes, possibles en protocole « T=1 » et naturellement

moins lourdes à manier.

La commande la plus utilisée est sans aucun doute « SELECT », qui sert à se déplacer dans l'arborescence des répertoires et fichiers, un peu comme la commande « CD » (*Change Directory*) de MS-DOS.

Dans le répertoire racine (repéré 3F00h), notre carte dispose des sous-répertoires suivants :

7F10h (Télécom)

7F20h (GSM)

7F21h (DCS, simple « fantôme » du répertoire GSM, utilisé par les mobiles 1800 MHz)

Pour sélectionner le répertoire « Télécom » à partir du répertoire racine, on utilisera ainsi une commande de la forme

A0 A4 00 00 02 7F 10.

Le **tableau 2** fournit la liste limitative des fichiers que contiennent les différents répertoires de notre carte.

Cette sélection, très étudiée, se compose des quelques fichiers dont l'existence est indispensable pour que la carte puisse exécuter une « session GSM » (autrement dit pour qu'elle soit reconnue comme une SIM), et de ceux permettant de se livrer aux manipulations les plus intéressantes.

Après sélection d'un fichier donné, des commandes de lecture (*read binary*, *read record...*) ou d'écriture (*update binary*, *update record...*) permettront de manipuler à volonté les données qu'il contient.

Pour une utilisation fructueuse de toutes les possibilités de la carte, il est évidemment nécessaire de connaître le rôle de ces différents fichiers.

Bien que la seule référence officielle et exhaustive soit la spécification GSM (www.etsi.org), et en particulier sa section GSM 11.11, voici l'essentiel de ce qu'il faut savoir.

Intéressons nous donc en priorité au répertoire GSM (ou DCS), qui offre le plus d'opportunités d'expérimentations instructives.

Les fichiers Phase et SST (*SIM Service Table*), tout d'abord, jouent un rôle important, dans la mesure où ils renseignent le mobile sur les possibilités de la carte (le symétrique, en somme, du « Terminal profile » que peuvent émettre certains mobiles).

Le fichier Phase contient un seul octet, égal à 02h en phase 2 ou 03h en phase 2+, et carrément absent dans la plupart des SIM de phase 1.

Le fichier SST, pour sa part, contient un nombre d'octets d'autant plus important que la carte SIM est perfectionnée. Chaque « service » susceptible d'être supporté par la carte est en effet matérialisé par deux bits indiquant respectivement si le service est disponible ou non, et s'il est activé ou non, selon l'affectation du **tableau 3**.

D'habitude, seul l'opérateur a le pouvoir de modifier la SST (par exemple lorsque l'on souscrit un service supplémentaire).

Dans notre cas, la SST n'est pas protégée en écriture, mais il ne faut pas y écrire n'importe quoi pour autant !

Son contenu d'origine (DF 30 C3 F3 00 00 00 03) reflétant fidèlement les possibilités

Tableau 3.

Les services de la SIM Service Table.

Octet	Actif	Existe	SERVICES	GSM
1	b2	b1	Service 1: CHV (PIN)1 disable function	
	b4	b3	Service 2: Abbreviated Dialling Numbers (ADN)	
	b6	b5	Service 3: Fixed Dialling Numbers (FDN)	
	b8	b7	Service 4: Short Message Storage (SMS)	
2	b2	b1	Service 5: Advice of Charge (AoC)	
	b4	b3	Service 6: Capability Config. Parameters (CCP)	
	b6	b5	Service 7: PLMN Selector	
	b8	b7	Service 8: Party Subaddress	
3	b2	b1	Service 9: MSISDN	
	b4	b3	Service 10: Extension 1	
	b6	b5	Service 11: Extension 2	
	b8	b7	Service 12: SMS Parameters	
4	b2	b1	Service 13: Last Number Dialed (LND)	
	b4	b3	Service 14: Cell Broadcast Message Identifier	
	b6	b5	Service 15: Group Identifier Level 1	
	b8	b7	Service 16: Group Identifier Level 2	
5	b2	b1	Service 17: Service Provider Name	
	b4	b3	Service 18: Service Dialling Numbers (SDN)	
	b6	b7	Service 19: Extension 3	
	b8	b7	Service 20: RFU	
6	b2	b1	Service 21: VGCS Group Identifier List (EF VGCS and EF VGCS)	
	b4	b3	Service 22: VBS Group Identifier List (EF VBS and EF VBSS)	
	b6	b5	Service 23: Enhanced Multi-Level Precedence & Pre-emption Service	
	b8	b7	Service 24: Automatic Answer for eMLPP	
7	b2	b1	Service 25: Data download via SMS-CB	
	b4	b3	Service 26: Data download via SMS-PP	
	b6	b5	Service 27: Menu selection	
	b8	b7	Service 28: Call control	
8	b2	b1	Service 29: Proactive SIM	
	b4	b3	Service 30: Cell Broadcast Message Identifier Ranges	
	b6	b5	Service 31: Barred Dialling Numbers (BDN)	
	b8	b7	Service 32: Extension 4	
9	b2	b1	Service 33: De-personalization Control Keys	
	b4	b3	Service 34: Co-operative Network List	

RFU = Réserve pour de Futurs Usages

offertes par notre sélection de fichiers, il serait inopportun de tenter d'activer des services que la carte ne supporte pas.

On pourra, par contre, toujours désactiver occasionnellement tel ou tel service en mettant à zéro un ou plusieurs bits originellement à 1.

De même, on pourra faire passer temporairement la phase de 03h à 02h, et étudier les réactions qui en résultent (inhibition des fonctions « Proactive SIM », par exemple).

Si ICCID contient le numéro de série de la carte, nullement confidentiel, IMSI identifie l'utilisateur auprès de son opérateur. Son « numéro de compte », en somme...

Le code de l'opérateur est d'ailleurs incorporé dans IMSI, et c'est ce qui permet au « roaming » de fonctionner : lorsqu'un réseau reconnaît un IMSI étranger, il demande à l'opérateur émetteur de la SIM s'il autorise son client à utiliser ses services, et dans l'affirmative, son nom s'affiche sur l'écran.

Dans notre cas, le code réseau d'origine est 001-01, valeur correspondant à un réseau fictif et réservée aux SIM de test.

Dans le même ordre d'idées, nous avons également programmé, par défaut, le fichier AD avec une valeur propre aux cartes de test. Cela présente l'avantage que la carte ne sera pas refusée par un mobile « sim-locké » par un opérateur ou un fournisseur de services, et pourra donc être utilisée sans restriction sur celui-ci.

Les fichiers GID1 et GID2 jouent d'ailleurs un rôle comparable, certains mobiles n'acceptant que les SIM contenant une valeur bien précise dans l'un ou l'autre de ces fichiers normalement inaccessibles en écriture.

Le contenu de LOCI se trouvera, pour sa part,

mis à jour par chaque réseau sur lequel on tentera peut-être d'inscrire manuellement le mobile, tentative qui se soldera bien évidemment par un échec puisque les identifiants de la SIM sont factices.

On y retrouvera alors le code du dernier opérateur sollicité, ainsi que quelques détails concernant le lieu où la tentative a été effectuée et la raison de l'échec de l'inscription.

Simultanément, le code de chaque réseau ayant rejeté la SIM s'inscrit dans le fichier FPLMN, à concurrence d'un maximum de quatre.

Bien entendu, cette liste peut être vidée à volonté, en remettant tous les octets de FPLMN à FFh.

La plupart des fichiers du répertoire « Télécom » sont affectés aux données personnelles de l'utilisateur : répertoires de numéros de téléphone et mini-messages « SMS ».

Différentes options du menu de n'importe quel téléphone portable permettent d'intervenir sur le contenu de ces fichiers, mais il existe aussi des applications pour PC qui apportent un considérable surcroît de confort.

Les logiciels auxiliaires

N'importe quel lecteur de cartes à puce asynchrones permet, en principe, d'intervenir sur tous les fichiers de notre carte à partir du clavier d'un PC, cela par l'intermédiaire de

commandes ISO 7816 de « bas niveau ».

Pour écrire 02h dans l'octet « Phase », par exemple, on pourrait enchaîner les commandes suivantes :

```
A0 A4 00 00 02 3F 00
A0 A4 00 00 01 7F 20
A0 A4 00 00 02 6F AE
A0 D6 00 00 01 02
```

Il faut pourtant bien reconnaître que les versions « Pro » des logiciels de gestion de cartes SIM facilitent infiniment les choses, grâce à leur « éditeur » incorporé.

L'un des meilleurs exemples en la matière est SIMSurf Profi, livré dans certains kits bâtis autour des lecteurs « ChipDrive » de Towitoko (www.towitoko.de).

Capables de fonctionner en mode PC/SC (moyennant l'installation des drivers ad-hoc), ces lecteurs très populaires en Europe sont également compatibles avec l'utilitaire UTILPCSC.EXE nécessaire pour exploiter le fichier « LOG » que notre carte est capable de mettre en oeuvre.

Ses versions UTIL1.EXE et UTIL2.EXE, pour leur part, sont destinées aux lecteurs « CyberMouse » (ACR20S ou ACR30S de chez ACS) fournis dans les kits BasicCard (www.basiscard.com).

Ces différents fichiers utilitaires évoqués dans l'encadré ainsi que le fichier TPIMG à transférer sur la carte-SIM du type ZC4.1 sont disponibles au téléchargement sur le site d'Elektor (www.elektor.fr) et pour ceux qui n'auraient pas accès à la Toile, sur une disquette (EPS010138-11) disponible auprès des adresses habituelles.

Un driver spécial (disponible sur le site <http://www.acs.com.hk>) permet éventuellement l'utilisation de UTILPCSC.EXE, mais il est au moins aussi commode de rester en mode série « natif » !

Développé, tout comme la carte SIM, en ZCBasic version 4, ce logiciel prend en charge les quatre opérations de base relatives au fichier « LOG » : activation (Activate), désactivation (Deactivate), téléchargement (Download), vidage (Clear).

En général, on activera le fichier



Réalisation de la carte

Bien que ce projet ait été entièrement développé au moyen d'un kit BasicCard version 4.12, il n'est aucunement nécessaire de disposer de l'intégralité de celui-ci pour réaliser et utiliser l'émulateur de carte SIM.

En pratique, il est nécessaire et suffisant de réunir les éléments suivants :

- l'utilitaire BCLOAD.EXE, inclus dans le kit de développement téléchargeable gratuitement sur www.basicc card.com ;
- un lecteur de cartes à puce compatible, qui pourra aussi bien être le « CyberMouse » fourni dans les kits BasicCard du commerce, que n'importe quel lecteur PC/SC correctement installé ;
- une BasicCard vierge compatible avec le fichier de programmation TP.IMG (en l'état actuel des choses, une ZC4.1 RSA [2001.09.28]).

Le lecteur de cartes à puce étant installé sur le PC selon les instructions de son fabricant, on ouvrira une fenêtre MS-DOS sur un répertoire de travail contenant le fichier TP.IMG et l'utilitaire BCLOAD.EXE.

On tapera alors simplement l'une ou l'autre des commandes suivantes :

- BCLOAD -D -P1 TP.IMG si on utilise un « CyberMouse » branché sur COM1 ;
- BCLOAD -D -P2 TP.IMG si on utilise un « CyberMouse » branché sur COM2 ;
- BCLOAD -D -P101 TP.IMG si on utilise un lecteur compatible PC/SC, quel que soit le port sur lequel il se trouve connecté.

Si le lecteur est correctement reconnu, il doit s'afficher une invitation à insérer la carte vierge (**écran A**).

Sauf incompatibilité entre la carte et le fichier, qui serait alors signalée par BCLOAD (**écran B**), le processus de programmation doit se traduire par le défilement, sur l'écran, d'une liste d'adresses dont on attendra évidemment la fin pour retirer la carte (**écran C**).

Il faut maintenant initialiser la carte, en exécutant simplement la fonction « Clear LOG file » de notre logiciel. Attention, cela peut prendre plusieurs secondes la première fois (**écran D**) !

Trois versions distinctes du logiciel sont fournies :

- UTIL1.EXE, destinée à un « CyberMouse » branché sur COM1 ;
- UTIL2.EXE, destinée à un « CyberMouse » branché sur COM2 ;
- UTILPCSC.EXE, destinée à un quelconque lecteur PC/SC.

Il ne restera plus qu'à découper la carte au format « SIM Micro » lorsque sera venu le moment de l'insérer dans un téléphone portable GSM.

La **figure 1** fournit les cotes permettant d'exécuter ce travail avec toute la précision voulue, si tant est que la BasicCard utilisée ne soit pas déjà prédécoupée (cela dépend des séries !). Si l'on sort alors la puce avec un minimum de soins on pourra toujours la remettre en place en dotant l'arrière de la carte, au niveau de la découpe, d'un morceau de scotch à effet temporaire. La carte pourra alors être réinsérée en toute sécurité dans le lecteur pour une nouvelle lecture des informations glanées dans le mobile.

Notons que l'on trouve facilement, chez les revendeurs de téléphones portables, des adaptateurs permettant de ramener la carte à son format primitif à chaque fois que l'on devra la réintroduire dans le lecteur du PC.

Notons que les programmes présents dans le répertoire BasicCardPro créé lors de l'installation de l'environnement de développement offrent nombre d'autres possibilités. Il est possible ainsi, d'obtenir des informations sur les cartes que l'on enfiche dans le lecteur (cf. **écran E**). Nous vous laissons le plaisir d'expérimenter avec cet outil à la palette de potentialités très riche.

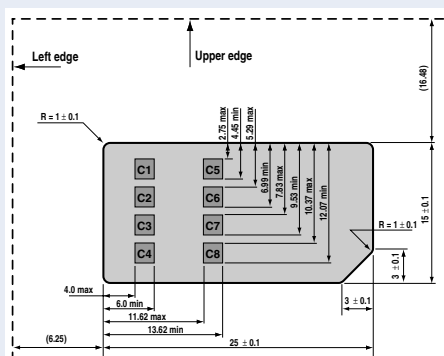


Figure 1. Cotes d'une carte SIM à mettre dans un mobile.

```
BCLOAD
Microsoft(R) Windows 98
(C)Copyright Microsoft Corp 1981-1998.
C:\WINDOWS>CD\
C:\>CD BASICC1
C:\BasicCardPro>BCLOAD -D -P1 TP.IMG
BCLOAD BasicCard Loader Utility Version 4.20 (c) ZeitControl 2001
Insert card in reader, or press any key to abort:
```

```
MS-DOS prompt
Microsoft(R) Windows 98
(C)Copyright Microsoft Corp 1981-1998.
C:\WINDOWS>CD\
C:\>CD BASICC1
C:\BasicCardPro>BCLOAD -D -P1 TP.IMG
BCLOAD BasicCard Loader Utility Version 4.20 (c) ZeitControl 2001
Insert card in reader, or press any key to abort:
SET STATE LOAD
Error: Image file TP.IMG was created for a BasicCard of type ZC4.1 RSA (2001.09.28), but this card is an Enhanced BasicCard ZC3.9
C:\BasicCardPro>
```

```
MS-DOS prompt
WRITE EEPROM 8H2E00
WRITE EEPROM 8H2E40
WRITE EEPROM 8H2E80
WRITE EEPROM 8H2EC0
WRITE EEPROM 8H2F00
WRITE EEPROM 8H2F40
WRITE EEPROM 8H2F80
WRITE EEPROM 8H2FC0
WRITE EEPROM 8H3000
WRITE EEPROM 8H3040
WRITE EEPROM 8H3080
WRITE EEPROM 8H30C0
WRITE EEPROM 8H3100
WRITE EEPROM 8H3140
WRITE EEPROM 8H3180
WRITE EEPROM 8H31C0
WRITE EEPROM 8H3200
WRITE EEPROM 8H3240
WRITE EEPROM 8H3280
WRITE EEPROM 8H32C0
WRITE EEPROM 8H3300
WRITE EEPROM 8H3340
WRITE EEPROM 8H3380
EEPROM CRC
SET STATE TEST
C:\BasicCardPro>
```

```
UTIL
SIM Emulator utilities Copyright (c)2001 Patrick GUEULLE
Insert card in reader, or press any key to abort:

1 -> Download LOG file
2 -> Activate LOG file
3 -> Deactivate LOG file
4 -> Clear LOG file
0 -> Quit
Your choice, then ENTER
```

```
cardinfo
Insert card in reader, or press any key to abort:
ATR historical bytes: BasicCard ZC1.1
Type: Compact BasicCard ZC1.1
State: LOAD (no Basic program active)
EEPROM start 8H8020, length 992 bytes

Remove card from reader, or press any key to abort:
Insert card in reader, or press any key to abort:
ATR historical bytes: BasicCard ZC3.9
Type: Enhanced BasicCard ZC3.9
State: LOAD (no Basic program active)
EEPROM start 8H8020, length 8096 bytes

Remove card from reader, or press any key to abort:
Insert card in reader, or press any key to abort:
ATR historical bytes: 0
Type: BasicCard "ZC4.1 RSA [2001.09.28]"
State: TEST (Basic program active, can be overwritten)
Application ID not configured

Remove card from reader, or press any key to abort:
```


Tableau 4.
Éléments d'interprétation
du « Terminal profile ».

Octet 1 :

- Bit 1 : Profile download
- Bit 2 : SMS-PP data download
- Bit 3 : Cell Broadcast data download
- Bit 4 : Menu selection

Octet 2 :

- Bit 1 : Command result
- Bit 2 : Call Control by SIM

Octet 3 (Proactive SIM) :

- Bit 1 : Display Text
- Bit 2 : Get Inkey
- Bit 3 : Get Input
- Bit 4 : More Time
- Bit 5 : Play Tone
- Bit 6 : Poll Interval
- Bit 7 : Polling Off
- Bit 8 : Refresh

Octet 4 (Proactive SIM) :

- Bit 1 : Select Item
- Bit 2 : Send Short message
- Bit 3 : Send SS
- Bit 4 : Send USSD
- Bit 5 : Setup Call
- Bit 6 : Setup Menu
- Bit 7 : Provide Local Information

« LOG » juste avant de placer la SIM dans un téléphone GSM, et on le désactivera aussitôt après l'en avoir retirée, afin d'écartier tout risque de saturation ultérieure.

On téléchargera alors le fichier (qui se retrouvera sur le disque dur du PC, sous le nom *CARD.LOG*), puis on le videra afin de libérer de

la place pour de nouvelles manipulations.

CARD.LOG est un fichier ASCII, lisible à l'aide de n'importe quel éditeur de texte, et contenant des valeurs hexadécimales.

Une ligne distincte est affectée à chaque commande reçue par la carte, tandis que toute ligne vide indique un RESET de la carte à cet instant précis (arrêt et remise en route du téléphone, par exemple).

Voici un court extrait enregistré lors des premiers instants de fonctionnement d'un mobile compatible Phase 2+, à la fin duquel on ne manquera pas de remarquer la commande « Terminal profile » par laquelle le téléphone informe la carte SIM de l'étendue de ses propres possibilités

```
A0 A4 00 00 02 7F 20
A0 C0 00 00 16
A0 A4 00 00 02 6F AE
A0 C0 00 00 0F
A0 B0 00 00 01
A0 A4 00 00 02 6F 05
A0 C0 00 00 0F
A0 B0 00 00 03
A0 A4 00 00 02 6F 05
A0 C0 00 00 0F
A0 A4 00 00 02 7F 20
A0 C0 00 00 16
A0 A4 00 00 02 6F AE
A0 C0 00 00 0F
A0 B0 00 00 01
A0 10 00 00 04 0F 03 FF F7
```

C'est d'ailleurs ce mot hexadécimal 0F 03 FF F7 qui s'affichera sur l'écran du mobile lorsqu'il aura terminé sa procédure d'initialisation (ces quelques longs instants pendant lesquels on obtient souvent le message « Essayez plus tard » si l'on tente d'appeler une fonction un tant soit peu gourmande en ressources machine).

Son décodage (tableau 4) d'après la spécification GSM 11.14 indique que ce mobile supporte la totalité des fonctionnalités SIM Toolkit actuelles, à l'exception de la fonction « Send USSD ».

Cela permet de dépister, par exemple, l'incompatibilité de cet appareil avec diverses fonctions avancées (envoi de SMS, d'E-mails, etc.) d'une étonnante carte SIM prépayée internationale, la « GSM card easyRoam » de Swisscom (<http://www.easy-roam.com>), ce que ne laisserait aucunement soupçonner l'étude de la notice du constructeur !

Mais nous arrivons là au stade du cas particulier : chaque modèle de mobile donnera des résultats différents, d'ailleurs largement variables en fonction des changements que l'on aura jugé utile d'apporter au contenu par défaut des fichiers de la carte SIM.

Bien entendu, l'auteur ne peut prétendre avoir « fait le tour » de tous les types de téléphones portables en circulation, bien qu'il ait expérimenté sa carte SIM avec des modèles aussi bien récents qu'anciens de Motorola, Nokia, Alcatel, Panasonic, Sagem, etc.

Il n'a pas davantage épuisé toutes les découvertes que permet potentiellement un tel outil, et invite cordialement ses lecteurs à pousser toujours plus loin l'aventure !

(010138)



NdlR : le seul petit problème que nous avons rencontré pour le moment avec ce projet est le coût relativement élevé de ces cartes, près de 10 \$US par carte (ce qui se justifie par sa complexité) avec un minimum, pour l'instant, de 10 cartes (l'impact des frais de port se justifie difficilement pour un nombre plus faible).

Intercepteur de données pour Märklin Digital

Des paquets d'impulsions sur l'écran du PC

Nils Körber DHØHAN

Peu importe que l'on tourne sous DELTA-Control, Control Unit, Digital Railrunner, EDiTS Pro ou tout autre système : sur les rails, les signaux de pilotage numérique de locomotives au « format Motorola » présentent une très grande similitude. Grâce à l'intercepteur de données décrit dans le présent article il devient possible d'observer ces signaux en « *live* » sur l'écran de son PC.

Nous avons eu maintes occasions de décrire le protocole du système de commande Märklin Digital [1] (cf. le numéro 251, mai 1999). Pour cette raison nous n'entrons pas, dans le cadre de cet article, dans le détail. En théorie, la transmission des données se fait sous la forme de trains d'impulsions constitués soit de 18 impulsions individuelles soit de 9 paires d'impulsions. Les niveaux de tension relevés sur les rails atteignent alors de l'ordre de -15 et $+15$ V, les valeurs exactes de ces niveaux de tension variant légèrement en fonction de l'appareil servant au pilotage et de la charge que constituent les différents modules connectés au système.

En fonction des circonstances, une impulsion unique sera interprétée comme un bit, une paire d'impulsions constituée de 2 impulsions distinctes l'étant comme un trit (bit ternaire). La valeur logique correspondante se cache dans le rapport cyclique des signaux voire dans leur largeur d'impulsion à cet instant. Dans le flux de données numériques véhiculé par les rails les bits et trits

prennent l'aspect suivant :

Bit « 0 »

$26 \mu\text{s}$ à $+15$ V, $182 \mu\text{s}$ à -15 V

Bit « 1 »

$182 \mu\text{s}$ à $+15$ V, $26 \mu\text{s}$ à -15 V

Trit « 0 »

$26 \mu\text{s}$ à $+15$ V, $182 \mu\text{s}$ à -15 V

$26 \mu\text{s}$ à $+15$ V, $182 \mu\text{s}$ à -15 V

Trit « 1 »

$182 \mu\text{s}$ à $+15$ V, $26 \mu\text{s}$ à -15 V

$182 \mu\text{s}$ à $+15$ V, $26 \mu\text{s}$ à -15 V

Trit « ouvert »

$182 \mu\text{s}$ à $+15$ V, $26 \mu\text{s}$ à -15 V

$26 \mu\text{s}$ à $+15$ V, $182 \mu\text{s}$ à -15 V

Dans l'ancien format Motorola un paquet d'impulsions complet se compose de ce fait de 4 trits pour l'adresse de locomotive, de 1 trit

pour la fonction « *function* » et de 4 trits pour la vitesse y compris l'information de changement de sens de circulation. Dans le format Motorola « nouveau » les 4 derniers trits évoqués sont remplacés par 8 bits servant à donner la vitesse, l'information de sens de circulation et les fonctions f1 à f4. La longueur des trains d'impulsions est de ce fait restée la même dans les 2 formats, à savoir de l'ordre de 3,75 ms.

Pour notre intercepteur de données, le signal de sortie du booster disponible sur les rails est appliqué, après une adaptation de niveau, à l'interface série du PC. Le programme de l'intercepteur de données échantillonne, au cours d'une série de mesures effectuées pendant quelques secondes, à des intervalles très rapprochés les niveaux présents (*sampling*) et stocke les résultats de ces opérations. En fin de mesure, le

programme traite les valeurs de mesure avant de les visualiser sur l'écran du PC.

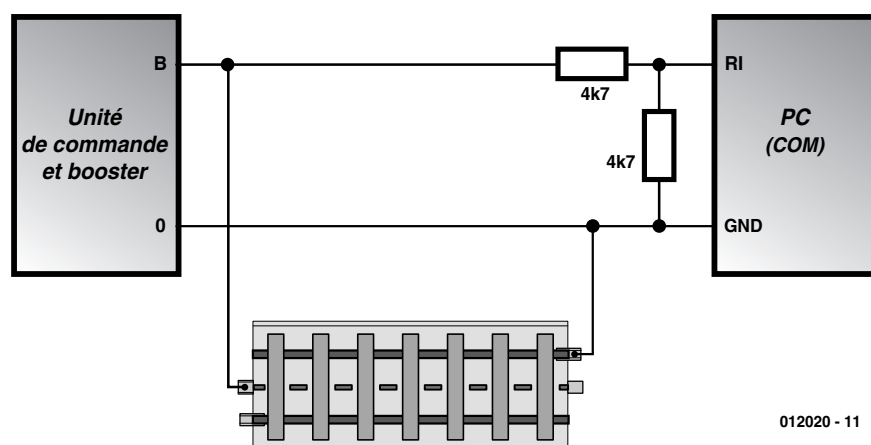
La qualité de la mesure dépend intrinsèquement de la vitesse et de la constance du taux d'échantillonnage. Il faut partant tenir compte d'un certain nombre de facteurs (nous entrerons dans le détail dans le paragraphe « Le logiciel »). Il faudrait, à taux d'échantillonnage constant, que 2 échantillonnages aient lieu dans la largeur d'impulsion courte de 26 μ s si l'on veut obtenir un résultat de mesure fiable. Ceci implique un taux d'échantillonnage (*sampling rate*) minimum de quelque 83,5 kHz. Un PC moderne tournant sous Windows ne devrait pas avoir de problème à ce niveau. Sur son PC équipé d'un Celeron d'Intel tournant à 466 MHz l'auteur a relevé un taux d'échantillonnage de l'ordre de 350 kHz.

L'aspect « matériel » : un diviseur de tension

Comme nous le disions plus haut, le signal présent sur les rails est appliqué à l'interface série du PC. Étant donné le risque de rencontrer des tensions qui pourraient, occasionnellement, dépasser 15 V, un diviseur de tension (**figure 1**) constitué d'une paire de résistances de 4,7 k Ω , fait en sorte que les niveaux arrivant au PC se situent dans les limites définies par la norme RS232C/V.28.

On utilise, au niveau de l'interface série, les lignes RI (*Ring Indicator*) et GND (*Ground*, masse). Si l'on a affaire à une embase à 25 broches, la ligne RI se trouve sur la broche 22, la masse sur la broche 7; dans le cas d'une embase à 9 contacts RI se trouve en broche 9 et GND en broche 5. Il n'y a pas d'inconvénient à ce que la liaison bifilaire vers le PC ait une certaine longueur, mais il faudra impérativement respecter la polarité correcte. Le point de connexion B va, sur le schéma de Märklin, à la ligne centrale du point de contact.

Il faudra en outre être conscient d'une réalité : en raison de la liaison vers le PC, la boucle de courant constituée par la voie du réseau ferroviaire ne présente plus un potentiel indépendant; en effet, elle se trouve reliée, par le biais du point de



012020 - 11

Figure 1. Voici comment interconnecter le booster/système de pilotage au PC.

connexion de masse, à la terre. Si cette particularité pose des problèmes avec d'autres extensions dont vous auriez doté votre réseau, il faudra déconnecter celles-ci de la boucle de courant constituée par votre réseau.

Le logiciel

Le programme de l'intercepteur de données a été écrit en Visual BASIC 5 (Microsoft). L'accès à l'interface série fait appel à une DLL (*Dynamic Link Library*), la fameuse PORT.DLL, évoquée dans le cours « L'électronique sur PC » publié l'an

dernier ([2]), sachant cependant que cette DLL peut également être téléchargée du site Internet d'Elektor.

L'interface utilisateur

L'interface utilisateur que l'on découvre en **figure 2** comporte 5 domaines :

– Settings

On peut dans cette fenêtre choisir l'interface série à utiliser (COM 1 ou COM 2) et définir le nombre d'échantillons (*samples*) à prendre pour la mesure.

– Measurement

Le bouton Start, comme on peut se l'imaginer aisément, démarre une mesure. Le bouton

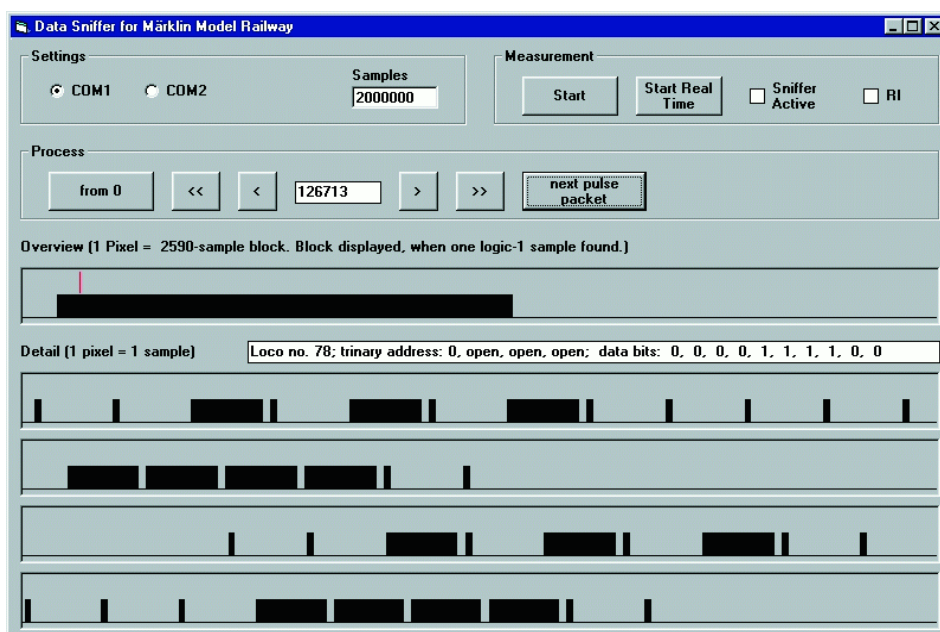


Figure 2. L'interface utilisateur du programme de l'intercepteur de données.

Start Real Time lance lui aussi la mesure, mais, par désactivation des interruptions en cours de mesure, évite toute intervention d'autres processus tournant sous Windows au cours de la mesure, ce qui se traduit par une meilleure qualité des mesures et une augmentation du taux d'échantillonnage.

En cours de mesure, le repère de fonction « Sniffer Active » est doté d'un onglet. Entre les mesures, le repère RI visualise, par échantillonnage cyclique, le niveau de la ligne RI.

– Process

Les différents boutons de cette fenêtre-barre d'outils servent à la commande du domaine d'affichage détaillé. On y voit en outre apparaître le numéro d'ordre du premier échantillon affiché. Le bouton « *next pulse packet* » constitue une spécificité de notre intercepteur : ce bouton prépositionne l'affichage détaillé au début (plus exactement juste un peu avant le début) du train d'impulsions suivant et force le traitement du contenu de ce train d'impulsions. Le résultat de ce traitement est alors affiché sous la forme d'un texte dans la fenêtre de visualisation détaillée.

– Fenêtre de visualisation (Overview)

On voit apparaître dans cette fenêtre l'ensemble de la mesure, les échantillons étant compactés sous la forme de blocs. Si l'un des blocs comporte un « 1 » (ce qui correspond à un niveau de 15 V, le bloc est marqué comme étant actif. On voit en outre apparaître un trait rouge à l'endroit où commence l'affichage détaillé.

– Visualisation détaillée (Detail)

L'affichage du détail visualise dans 4 fenêtres horizontales s'étalant sur toute la largeur de la fenêtre les valeurs des différents échantillons. Le premier échantillon affiché pourra être commandé par le biais des boutons présents dans la fenêtre « *Process* ». On a visualisation ainsi, à tout moment, d'un total de 3 088 échantillons.

La recopie d'écran montre le paquet d'impulsions, au format « Motorola ancien », corres-

pondant à la locomotive numéro 78. Les bits de données visualisés 0000111100 éventuellement combinés en bits de données ternaires 00110 indiquent que la fonction est désactivée et que la vitesse est fixée à 6. L'affichage commence à l'échantillon 130297 d'un total de 2 000 000. L'ensemble de la mesure a nécessité de l'ordre de 6 s.

Spécificités et système minimum

Il faudra, pour disposer de résultats de mesure fiables, opter, en règle générale, pour le bouton « Start Real Time ». Comme nous le disions plus haut, ce bouton inhibe (désactive), en cours de mesure, toutes les interruptions pouvant être désactivées et limite de ce fait les perturbations que pourraient entraîner d'autres programmes Windows tournant en arrière-plan. Il faudra en outre éviter, en cours de mesure, de procéder à quelque opération que ce soit au niveau du PC, action sur le clavier ou mouvement de souris par exemple.

Le programme de l'intercepteur de données est disponible sous la forme d'un Setup VB5. Il n'est par tant pas nécessaire de disposer sur son PC d'un Visual BASIC quel qu'il soit. Le programme est disponible gratuitement sur le site Internet d'Elektor (www.elektor.presse.fr) ou sous la forme d'une disquette (EPS0102020-11) disponible auprès des adresses habituelles. On y trouvera tant le code-source que le programme compilé (.exe) destiné à tourner sur le PC. Notons que le programme a été compilé en code système (native code) plutôt qu'en code P classique, ce choix se traduisant par un accroissement sensible du taux d'échantillonnage.

L'intercepteur de données requiert un PC disposant d'une interface série libre, COM1 ou COM2 et tournant sous Windows, qu'il s'agisse de WIn98, 98, NT4.0 ou 2000 peu importe. L'exécution du programme de devrait pas poser de problème sur tout PC dont la vitesse de processeur est de 200 MHz ou plus. Il faudra veiller à ce que soit la boucle de courant que constitue l'ensemble du réseau ferroviaire soit la sortie du module de commande/booster soit mise hors-potential par rapport à la terre ou être reliée à la terre par le biais de l'élément conducteur que constitue le rail (cette mise à la terre ne devant cependant pas se faire par le biais du rail central !). En effet, de par la liaison au PC, la boucle de courant formée par le réseau n'est plus hors-potential, mais reliée à la terre par le biais de la ligne GND de l'interface du PC.

(012020)

Merci Mr Dezair pour la relecture !

Littérature et adresses Internet

- [1] Numérisation & réseaux ferroviaires miniature, Elektor n° 251, mai 1999, page 62 et suivantes
- [2] « L'électronique sur PC », Elektor n° 267 à 273, septembre 2000 à mars 2001

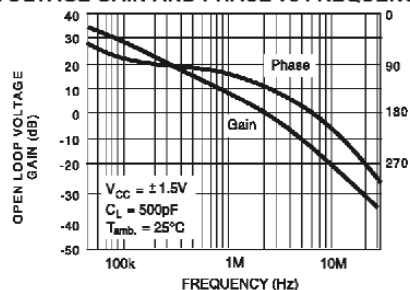
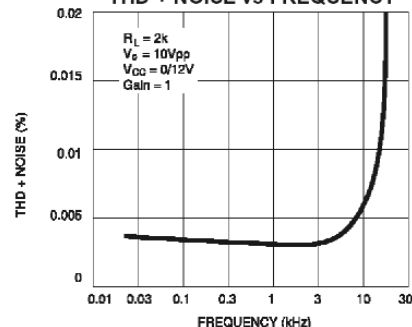
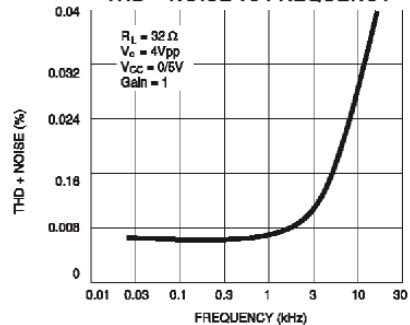
Page d'accueil de l'auteur où l'on pourra trouver d'autres projets (non électroniques aussi) ayant trait au modélisme ferroviaire : www.koerber-home.de.

TS925

Quadruple Opamp à masse virtuelle

ELEKTOR
INFOCARTE 4/2002**Valeurs-limite**

Symbole	Paramètre	Valeur	Unité
V_{CC}	Tension d'alimentation	14	V
V_{id}	Tension d'entrée différentielle	± 1	V
V_i	Tension d'entrée (ne doit jamais dépasser $V_{CC} + 0,3$ V)	-0,3 à +14	V
T_{oper}	Température de service	-40 à +125	°C

VOLTAGE GAIN AND PHASE vs FREQUENCY**THD + NOISE vs FREQUENCY****THD + NOISE vs FREQUENCY****TS925**

Quadruple Opamp à masse virtuelle

ELEKTOR
INFOCARTE 4/2002**TS925**

Quadruple Opamp à masse virtuelle

Fabricant :

STMicroelectronics

Fiche de caractéristiques

<http://eu.st.com/stonline/books/pdf/docs/4949.pdf>**Caractéristiques :**

- Entrées et sorties rail à rail
- Faible niveau de bruit de 9 nV/√Hz
- Taux de distorsion faible
- Courant de sortie élevé : 80 mA
- Capable d'attaquer directement des charges de 32 Ω
- Rapide : 4 MHz, 1,3 V/μs
- Tension d'alimentation : 2,7 à 12 V
- Tension d'offset d'entrée faible : 900 μV
- Masse fantôme réglable
- Mode de standby

Description :

Le TS925 est un quadruple amplificateur opérationnel BiCMOS à entrées et sorties rail à rail, spécifié pour une alimentation en 3 et 5 V. Le courant de sortie important lui permet d'attaquer des impédances de charge faibles (jusqu'à 500 pF) sans le moindre problème. Une masse virtuelle interne à impédance

faible élimine tout besoin de tension de référence externe ou la prise de quelque mesure additionnelle pour la définition du point de fonctionnement. De par ses valeurs de bruit intrinsèque, de distorsion faibles et son courant de sortie élevé, le TS925 est prédestiné aux applications d'audio haut de gamme et aux systèmes de télécommunication requérant des tensions d'alimentation faibles. En mode de veille (stand-by) la consommation de courant totale est de 6 μA (à $V_{CC} = 3$ V).

Applications typiques :

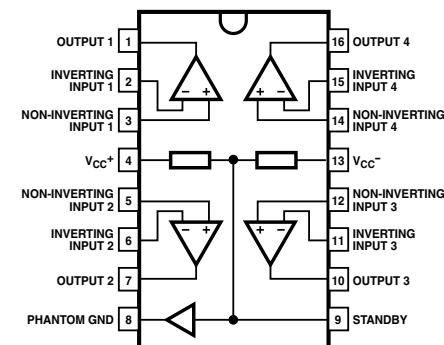
- Amplificateur pour casque d'écoute
- Amplificateur pour cartes-son, hauts-parleurs piézo, carte MPEG et systèmes multimédia
- Téléphones de poing et équipements de communication portables, circuit de commande (driver) de puissance, tampon

Application :

Filtre-bouchon ajustable, Elektor n° 286, avril 2002

Types de boîtier

TS925(A)N	DIP16
TS925(A)D	SO16
TS925(A)P	TSSOP16
Plage de température : -40 à +125 °C	



010120 - 14

Brochage

TS925

Quadruple Opamp à masse virtuelle

ELEKTOR
INFOCARTE 4/2002

Valeurs électriques caractéristiques ($V_{CC}^+ = 5\text{ V}$, $V_{CC}^- = 0\text{ V}$, $T_{amb} = 25^\circ\text{C}$ sauf mention contraire)

Symbole	Paramètre	Conditions	Min.	Typ.	Max.	Unité
V _{io}	Tension d'offset en entrée	T _{amb} TS925			3	mV
		T _{amb} TS925A			0,9	mV
		T _{min} ≤ T _{amb} ≤ T _{max} TS925			5	mV
		T _{min} ≤ T _{amb} ≤ T _{max} TS925A			1,8	mV
DV _{io}	Dérive de la tension d'offset en entrée			2		μV/K
I _{io}	Courant d'offset en entrée	V _{out} = 2,5 V		1	30	nA
I _{ib}	Courant de repos en entrée	V _{out} = 2,5 V		15	100	nA
V _{OH}	Tension de sortie High (R _L à V _{CC/2})	R _L = 10 kΩ	4,90			V
		R _L = 600 Ω	4,85			V
		R _L = 32 Ω		4,4		V
V _{OL}	Tension de sortie Low (R _L à V _{CC/2})	R _L = 10 kΩ			50	mV
		R _L = 600 Ω			120	mV
		R _L = 32 Ω		300		mV
A _{vd}	Gain grand signal (V _{out} = 4 V _{SS})	R _L = 10 kΩ		200		mV/V
		R _L = 600 Ω		40		mV/V
		R _L = 32 Ω		17		mV/V
GBP	Produit Gain x Bande passante	R _L = 600 Ω		4		MHz
CMR	Rapport de la réjection en mode commun		60	80		dB
SVR	Rapport de la réjection de la tension d'alimentation	V _{CC} = 3 à 5 V	60	85		dB
I _O	Courant de court-circuit de sortie		50	80		mA
SR	Taux de montée (<i>slew rate</i>)		0,7	1,3		V/μs
P _m	Réserve de phase à gain unitaire	R _L = 600 Ω, C _L = 100 pF		68		degré
GM	Réserve de gain	R _L = 600 Ω, C _L = 100 pF		12		dB
e _n	Tension de bruit d'entrée équivalente	f = 1 kHz		9		nV/√Hz
THD	Distorsion harmonique totale	V _{out} = 3 V _{SS} , f = 1 kHz, A _v = 1, R _L = 600 Ω		0,01		%
C _S	Diaphonie			120		dB
I _{CC}	Consommation de courant totale	hors-charge, V _{out} = V _{CC/2}		6	8	mA
I _{stby}	Consommation de courant en veille	Broche 9 à V _{CC}		10		μA
V _{enstby}	Tension d'activation du mode Standby			0,3		V
		T _{min} ≤ T _{amb} ≤ T _{max}		0,4		V
V _{disbty}	Tension de désactivation du mode Standby		1,1			V
		T _{min} ≤ T _{amb} ≤ T _{max}	1			V
Masse fantôme						
V _{pg}	Tension de sortie	sans courant de sortie	V _{CC/2} − 5 %	V _{CC/2}	V _{CC/2} + 5 %	V
I _{pgsc}	Courant de court-circuit de sortie (source)		12	18		mA
I _{pgsk}	Courant de court-circuit de sortie (drain)		12	18		mA
Z _{pg}	Impédance	DC à 20 kHz		3		Ω
E _{npg}	Tension de bruit en sortie (f = 1 kHz) (C _{dec} : condensateur de découplage en broche 9)	C _{dec} = 100 pF		200		nV/√Hz
		C _{dec} = 1 nF		40		nV/√Hz
		C _{dec} = 10 nF		17		nV/√Hz

TS925

Quadruple Opamp à masse virtuelle

ELEKTOR
INFOCARTE 4/2002

Valeurs électriques caractéristiques ($V_{CC}^+ = 3\text{ V}$, $V_{CC}^- = 0\text{ V}$, $T_{amb} = 25^\circ\text{C}$ sauf mention contraire)

Symbole	Paramètre	Conditions	Min.	Typ.	Max.	Unité
V _{io}	Tension d'offset en entrée	T _{amb} TS925			3	mV
		T _{amb} TS925A			0,9	mV
		T _{min} ≤ T _{amb} ≤ T _{max} TS925			5	mV
		T _{min} ≤ T _{amb} ≤ T _{max} TS925A			1,8	mV
DV _{io}	Dérive de la tension d'offset en entrée			2		μV/K
I _{io}	Courant d'offset en entrée	V _{out} = 1,5 V		1	30	nA
I _{ib}	Courant de repos en entrée	V _{out} = 1,5 V		15	100	nA
V _{OH}	Tension de sortie High (R _L an V _{CC/2})	R _L = 10 kΩ	2,90			V
		R _L = 600 Ω	2,87			V
		R _L = 32 Ω		2,63		V
V _{OL}	Tension de sortie Low (R _L an V _{CC/2})	R _L = 10 kΩ			50	mV
		R _L = 600 Ω			100	mV
		R _L = 32 Ω		180		mV
A _{vd}	Gain grand signal (V _{out} = 2 V _{SS})	R _L = 10 kΩ		200		mV/V
		R _L = 600 Ω		35		mV/V
		R _L = 32 Ω		16		mV/V
GBP	Produit Gain x Bande passante	R _L = 600 Ω		4		MHz
CMR	Rapport de la réjection en mode commun		60	80		dB
SVR	Rapport de la réjection de la tension d'alimentation	V _{CC} = 2,7 à 3,3 V	60	85		dB
I _O	Courant de court-circuit de sortie		50	80		mA
SR	Taux de montée (slew rate)		0,7	1,3		V/μs
P _m	Réserve de phase à gain unitaire	R _L = 600 Ω, C _L = 100 pF		68		degré
GM	Réserve de gain	R _L = 600 Ω, C _L = 100 pF		12		dB
e _n	Tension de bruit d'entrée équivalente	f = 1 kHz		9		nV/√Hz
THD	Distorsion harmonique totale	V _{out} = 2 V _{SS} , f = 1 kHz, A _V = 1, R _L = 600 Ω		0,01		%
C _S	Diaphonie			120		dB
I _{CC}	Consommation de courant totale	keine Last, V _{out} = V _{CC/2}		5	7	mA
I _{stby}	Consommation de courant en veille	Broche 9 à V _{CC}		6		μA
V _{ensbty}	Tension d'activation du mode Standby			0,3		V
		T _{min} ≤ T _{amb} ≤ T _{max}		0,4		V
V _{distby}	Tension de désactivation du mode Standby		1,1			V
		T _{min} ≤ T _{amb} ≤ T _{max}	1			V
Masse fantôme						
V _{pg}	Tension de sortie	sans courant de sortie	V _{CC/2} - 5 %	V _{CC/2}	V _{CC/2} + 5 %	V
I _{pgsc}	Courant de court-circuit de sortie (source)		12	18		mA
I _{pgsk}	Courant de court-circuit de sortie (drain)		12	18		mA
Z _{pg}	Impédance	DC à 20 kHz		3		Ω
E _{npg}	Tension de bruit en sortie (f = 1 kHz) (C _{dec} : condensateur de découplage en broche 9)	C _{dec} = 100 pF		200		nV/√Hz
		C _{dec} = 1 nF		40		nV/√Hz
		C _{dec} = 10 nF		17		nV/√Hz

À la découverte du PSN

Expériences pratiques à base de processeurs de signal

Gunther May, DL3ABQ

Le traitement numérique de signaux effraie encore de nombreux utilisateurs potentiels pour la simple raison qu'elle fait appel, principalement, à des algorithmes mathématiques. Il serait bon d'arriver à faire fi de ces préjugés car il est possible, sans connaissances techniques approfondies, de procéder à moult expériences intéressantes au cours desquelles on fait inévitablement connaissance avec un minimum de théorie.

Il y a quelque temps déjà, Elektor avait proposé un cours baptisé « Espresso » [6], qui constituait une bonne introduction au traitement numérique de signaux. De nombreux thèmes y furent abordés à partir d'un minimum de théorie de base et de nombreux

essais effectués à l'aide d'une carte-son pour PC.

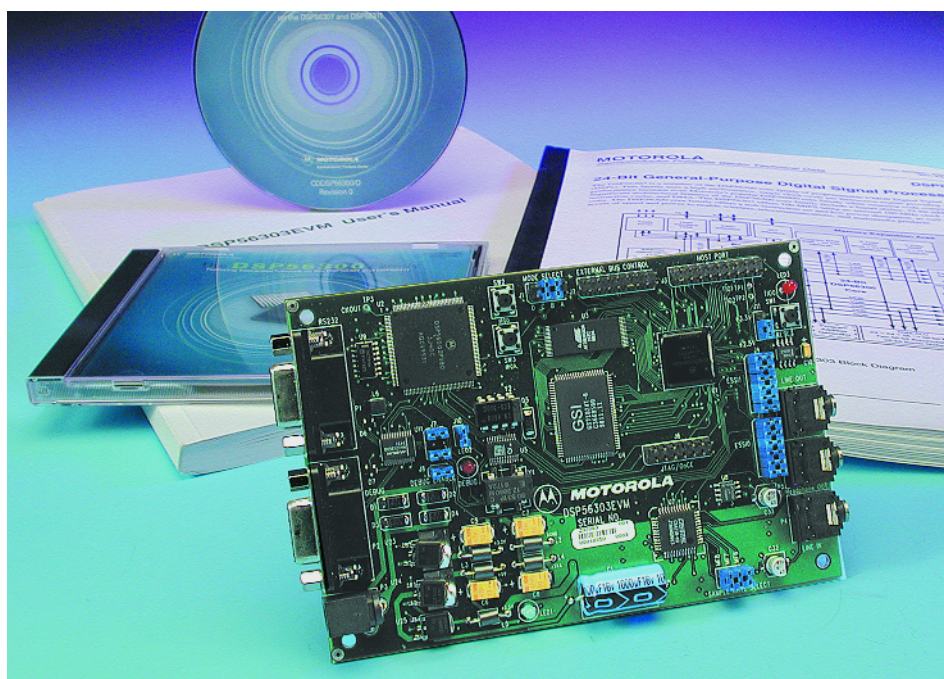
Nous avons, cette fois, décidé d'aborder ce thème d'une autre façon, à savoir sous son aspect pratique. Les algorithmes de traitement

numérique du signal sont, en règle générale, pris en compte par un processeur numérique de signal (PSN aka (*also known as* = alias) DSP = *Digital Signal Processor*). Il s'agit en fait d'un processeur de type particulier capable d'exécuter des séries d'instructions typiquement destinées au traitement de signaux, et ce à très grande vitesse. Nous allons commencer par nous intéresser à des systèmes comportant ce type de processeur.

Systèmes à PSN

La structure générique des systèmes destinés au traitement de signaux à l'aide d'un PSN ressemble, en règle générale, au synoptique de la **figure 1**.

On commence par préparer le signal d'entrée, il s'agit le plus souvent d'un signal analogique, de la musique par exemple, avant de le convertir en un signal numérique. De manière à éviter, lors de la conversion analogique/numérique (A/N), ce que l'on appelle des phénomènes de repliement (*aliasing*), nous ne pou-



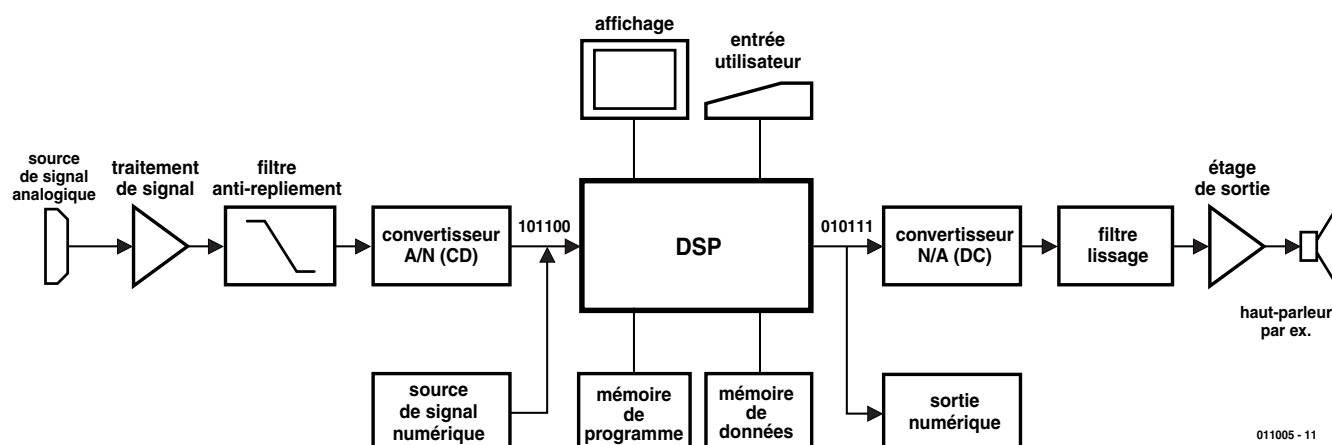


Figure 1. Modèle synoptique d'un système à PSN.

vons pas entrer dans le détail dans le cadre de cet article, on intercale le plus souvent, en amont du convertisseur analogique/numérique, un filtre passe-bas (analogique !) à flancs raides, filtre dont la fréquence de coupure ne doit pas dépasser la moitié de la valeur du taux d'échantillonnage, filtre que l'on a baptisé, qui s'en étonnerait, filtre anti-repliement (*anti-aliasing*). Ce filtre a pour fonction d'éliminer les composantes de fréquence les plus élevées.

Une fois la conversion effectuée on dispose, en numérique, d'une série de valeurs discrètes. On peut bien évidemment également utiliser directement un signal d'entrée numérique. En tout état de cause, le système à PSN peut se mettre maintenant au traitement du signal, en effectuer la visualisation et procéder à son stockage.

En fin de traitement le PSN fournit à nouveau un signal numérique qui pourra, par exemple, être traité par de nouveaux PSN montés en aval. Une autre approche consiste à reconverter les valeurs numériques discrètes en un signal analogique continu.

Cette opération est confiée à un convertisseur numérique/analogique (N/A) suivi d'un filtre de lissage qui remplit par interprétation les intervalles séparant les échantillons. Le signal ainsi obtenu pourra ensuite subir un nouveau traitement, en analogique cette fois, une amplification par exemple avant d'être appliqué à un haut-parleur destiné à sa reproduction.

Il existe sur le marché des circuits intégrés de conversion A/N-N/A dédiés qui intègrent en outre le filtre anti-repliement et le filtre de lissage. On appelle ce type de composant un CoDec (*Coder-Decoder*).

Kit d'évaluation PSN

Les fabricants de PSN proposent, pour la plupart, des kits d'évaluation (starter kit) de leurs produits. Il s'agit dans la plupart des cas de sys-

tèmes minimum au prix très abordable comportant un PSN et les composants connexes requis, le tout monté sur une platine. Il est facile, dès lors que l'on dispose d'une telle carte et du logiciel qui l'accompagne, de développer l'un ou l'autre programme de son cru.

Au nombre des composants connexes présents sur la carte on compte le plus souvent un Codec pour les conversions A/N et N/A, de la mémoire externe et de circuits offrant l'un ou l'autre type d'interface. Ce type de systèmes à PSN est hors de portée de l'ama-

Tableau 1 Revue des kits d'évaluation (Starter Kit)

	TI C3X Starter Kit	TI 67II Starter Kit	AD 21061 EZLITE	Motorola DSP56303EVM
Fabricant	Texas Instruments (www.ti.com)	Texas Instruments (www.ti.com)	Analog Devices (www.analog.com)	Motorola (www.motorola.com)
DSP	TMS320C31	TMS320C6711	ADSP-21061	DSP56303
MIPS/MFLOPS	25/50	1200/600	40/150	100/FLOPs à émulation logicielle seule
Précision de calcul	Virgule flottante 32 bits	Virgule flottante 32 bits	Virgule flottante 32 bits	Virgule fixe 24 bits
RAM	64 KB de RAM interne	16 MB de SDRAM externe	1 MB de SRAM externe	1,5 MB de SRAM externe
ROM	16 M de mémoire de programme/données	128 Koctets de Flash externe	1 MB d'EPROM externe	0,5 MB de Flash externe
Codec	Mono, 20 kHz, 14 bits	Stéréo, 48 kHz, 16 bits	téréo, 48 kHz, 16 bits	Stéréo, 48 kHz, 16 bits
Interfaces	Analogique, Parallèle	Analogique, Parallèle, JTAG, Carte-fille	Analogique, Sérielle, connecteur d'extension	Analogique, Sérielle, JTAG
Logiciels fournis	Assembleur, Débogueur	Assembleur, Débogueur, Compilateur-C (limités à 256 Koctets de code)	Assembleur, Débogueur, Compilateur-C	Assembleur, Débogueur, Compilateur-C
Autres accessoires fournis		Alimentation, câble de données	Alimentation (n'existe pas en version européenne), câble de données	
Prix (en \$US, ces prix pouvant varier quelque peu)	\$99	\$295	\$179	\$249

Informations non contractuelles; sauf erreur d'impression

teur de réalisations personnelles pour la simple et bonne raison qu'il est pratiquement impossible de souder manuellement les PSN puissants en raison du type de boîtier dans lequel ils sont proposés (TOFP par exemple). D'autres PSN ont un boîtier du type BGA qu'il n'est bien évidemment pas question de souder à la main. Vu d'autre part, que le prix de ce type de cartes est, de façon plus ou moins sensible, « subventionné » par le fabricant du type de PSN utilisé, ceci en vue de favoriser l'utilisation de leurs produits, le prix d'achat du PSN, si tant est que l'on puisse l'obtenir en petites quantités (10 ou moins) enlève tout intérêt à une réalisation personnelle. Il n'en reste pas moins que l'on peut être amené à mettre soi-même la main à la pâte pour réaliser le « montage de ses rêves ». Analog Devices, Motorola et Texas Instruments, sont, dans l'ordre alphabétique, les sources les plus connues de ce type de systèmes PSN d'évaluation, les fameux DSP-Starter Kits. Nous vous proposons dans le **tableau 1** un panorama de quelques-uns des kits PSN d'évaluation de ces fabricants.

Critères de sélection

On est bien évidemment en droit de se poser la question de savoir quelle carte convient le mieux à une application donnée. Voici quelques aspects dont il faudra tenir compte : Il est bon, dès le début, d'opter pour un PSN à virgule flottante car cette caractéristique évite à l'utilisateur d'avoir à se soucier de points tels que changement d'échelle, erreur d'arrondi et problèmes de dépassements dans un sens ou dans l'autre.

Il faudra, lors d'un examen critique au niveau du prix, que certaines cartes requièrent, pour le développement, l'acquisition de l'un ou l'autre logiciel additionnel.

Il faudra, lors de l'interprétation de données ayant trait à la vitesse telles que MIPS ou MFLOPS (la signification de ces abréviations est donnée un peu plus loin dans un tableau), se rappeler que les fabricants déterminent ces facteurs à partir de modèles différents, ce qui s'explique logiquement par les différences au niveau de l'architecture des instructions. Il est partant très difficile d'effectuer une comparaison correcte des valeurs données par les fabricants. D'autres facteurs dont nous ne pouvons pas donner le détail ici, entrent en ligne de compte pour l'établissement de la vitesse réelle de traitement.

Dans le tableau 1 les tailles de mémoire sont données en bits (KB et MB); il faudra, dans ce cadre, tenir compte de la largeur des mots d'instruction (ainsi, Texas annonce, pour le TMS320C31-50 utilisé ici en référence, 16 M (Words) de mémoire de programme/données

et 2 K (Words) de RAM (et d'autres facteurs encore).

Mise en oeuvre pratique

L'un des thèmes du traitement numérique de signaux que l'on retrouve, sous une forme ou une autre, dans presque chaque application est le filtrage de signaux. Les kits d'évaluation énumérés plus haut comportent tous au minimum un programme d'exemple de réalisation de filtre.

Il nous faut, avant de nous intéresser au fonctionnement pratique d'un filtre de ce genre, faire un rien de théorie à son sujet.

Filtre numérique

Nombre d'applications requièrent des filtres chargés d'extraire du mic-mac de fréquences dont est constitué tout signal, un signal utile donné, voire d'éliminer un signal dont on ne veut pas. Il existe, pour

cette fonction, différents types de filtres, les filtres passe-bas par exemple, chargés d'atténuer très fortement les composantes de fréquence élevée et de ne laisser passer que les fréquences les plus faibles ou encore les filtres passe-bande qui laissent passer quasiment sans la moindre atténuation les signaux se trouvant à l'intérieur d'une certaine plage de fréquences, les fréquences sortant, d'un bord ou de l'autre, du domaine défini étant elles très fortement atténuées.

La **figure 2** montre ce que l'on appelle une courbe d'amplitude d'un filtre. L'amplitude est représentée en ordonnée. Lorsque l'amplitude est de 0 dB, le signal passe sans encombre; la présence d'une atténuation se traduit par des amplitudes plus faibles. Nous avons ici affaire à un filtre passe-bas caractérisé par une fréquence de coupure de quelque 9 kHz.

On a besoin, par exemple, lorsque

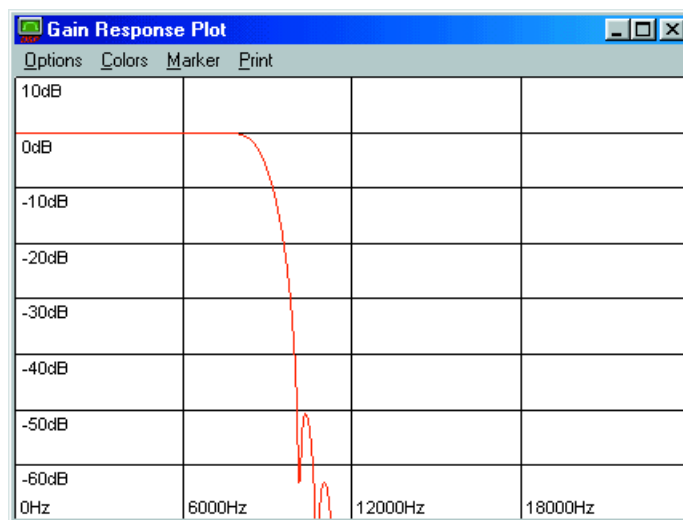


Figure 2. Courbe d'amplitude d'un filtre FIR numérique.

Abréviations

Codec	Coder/Decoder, convertisseur A/N-N/A pour la conversion de signaux analogiques
DSP	Digital Signal Processor processeur de signal numérique
FIR	Finite Impulse Response = filtre (non récursif)
IIR	Infinite Impulse Response = filtre (récursif)
MFLOPS	Mega Floating Point Operations per Second (Million d'opérations en virgule flottante par seconde)
MIPS	Mega Instructions per Second (Million d'instructions par seconde)

l'on veut réaliser un récepteur radio, en règle générale, de filtres passe-bande étroits de manière à pouvoir extraire le signal utile du mélange de fréquences capté ou encore en vue d'éliminer certaines fréquences.

Outre les filtres analogiques réalisés à l'aide d'inductances, de condensateurs et/ou de quartz et de résonateurs céramique, il est également possible, à l'aide d'un PSN, d'implémenter un filtre sous la forme d'un programme.

Il est en fait relativement facile de réaliser un filtre numérique de ce type : ils calculent le signal de sortie sous la forme d'une somme pondérée dérivée du signal d'entrée et des valeurs calculées auparavant. Ces filtres sont subdivisés en 2 catégories principales :

- les FIR (*Finite Impulse Response*) utilisent, pour les calculs, uniquement des valeurs d'entrée. Les avantages de ce type de filtres sont, d'une part, une garantie de stabilité (aspect que nous ne pouvons pas détailler plus ici) et la possibilité, d'autre part, de construire des filtres de phase absolument linéaires, filtres d'une importance capitale pour la transmission de données.

- les IIRs (*Infinite Impulse Response*) utilisent quant à eux, en outre des valeurs déjà calculées. Cette approche présente l'avantage de ne requérir, pour la réalisation d'une fonction de filtre, le calcul que d'un nombre beaucoup plus faible de valeur, ce qui diminue très fortement la quantité de calculs et la taille de mémoire nécessaires. Il est un aspect de ce type de filtre auquel il est important de veiller, leur stabilité. Il est en outre très difficile de réaliser des IIR qui aient une phase tant soit peu linéaires.

Implémentation sur PSN

On peut subdiviser le développement d'un filtre de ce genre en plusieurs étapes :

- Calcul du filtre à partir d'éléments propres que l'on entre dans un programme de conception de filtre
- Entrée des coefficient du filtre en code de programme de filtre PSN
- Conversion du code de programme en code machine
- Transfert du programme vers le système-cible.

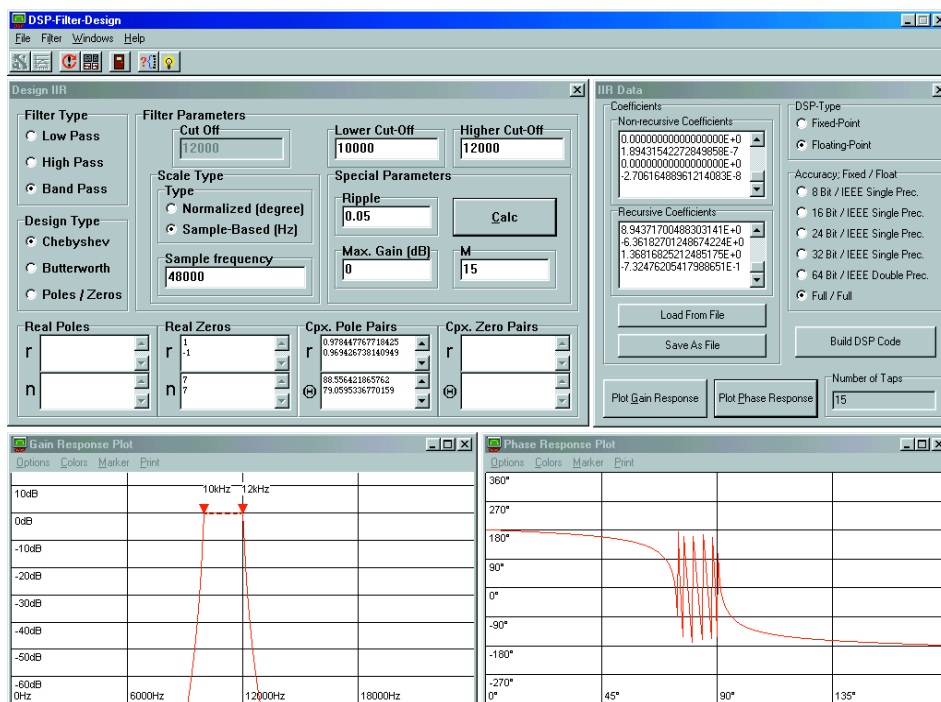


Figure 3. Interface utilisateur du logiciel « DSP-Filter-Design ».

Les 2 premières étapes pourraient être confiées, par exemple, au programme « *DSP Filter Design* » (cf. **figure 3**), programme écrit par l'auteur dans le cadre d'un concours « Jeunes & Inventions ».

Pour calculer un filtre il suffit à l'utilisateur d'entrer les caractéristiques de coupure du filtre à réaliser (dans la fenêtre supérieure gauche de la figure 3). À partir de là on a calcul des coefficients du filtre (en haut à droite); on aura ensuite visualisation de la courbe d'amplitude (la courbe des fréquences passantes) dans la fenêtre inférieure gauche et la réponse en phase (le déphasage

entre les signaux d'entrée et de sortie).

Le programme est en outre capable pour certains des PSN les plus courants, de générer directement, à partir des coefficients du filtre, le code de programme de sorte que l'utilisateur n'a pas à posséder de connaissances particulières en ce qui concerne la programmation. Les programmes accompagnant le kit d'évaluation permettront de réaliser les 2 dernières étapes, à savoir la compilation en vue d'obtenir le code machine et le téléchargement (*download*) vers le système-cible.

Le programme « *DSP Filter Design* » est disponible en version de démonstration à l'adresse Internet : www.qsl.net/dl3abq et sur le site d'Elektor : www.elektor.fr.

(011005-1)

Littérature :

- [1] Madiset, Williams (1998) : The Digital Signal Processing Handbook; CRC Press
- [2] Taylor, The Athena Group Inc., Mellot : Hands-on Digital Signal Processing; McGraw-Hill
- [3] McClellan, Schafer, Yoder (1998) : DSP First; Prentice Hall
- [4] El-Sharkawy (1994) : Signal Processing, Image Processing and Graphics Applications with Motorola's DSP96002 Processor; Prentice Hall
- [5] « Traitement de signaux par PSN » : Elektor n°233, novembre 1997, page 54 et suivantes
- [6] « Espresso » : Elektor n°235, janvier 1998 et suivants
- [7] Pages Internet des fabricants de PSN : Texas Instruments, Analog Devices et Motorola

Interface I²C pour RCX (Lego)

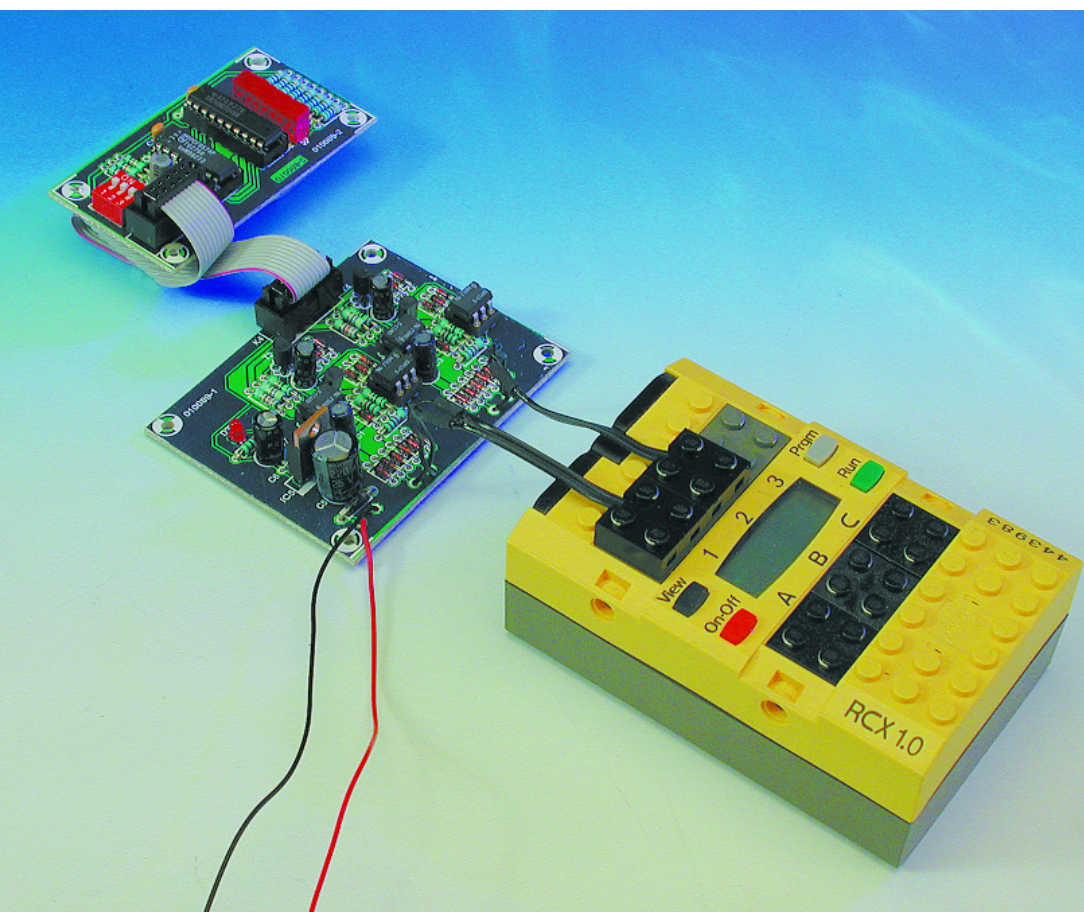
Des possibilités d'extension inédites !

Projet : Wim Huiskamp

Texte : David Daamen

Tout le monde, quasiment, sait aujourd'hui que le module RCX de Lego se prête étonnamment bien à toutes sortes d'expériences, celles ayant trait à la robotique en particulier. Cependant, dès que le projet concerné sort du cadre restreint d'expériences limitées, on constate inévitablement que le nombre d'entrées et de sorties que comporte ce module

est, peut-être, sur le bord juste. Ceci explique qu'Elektor vienne sur la scène avec une interface I²C pour ce bloc Lego. Elle ne manquera pas de vous ouvrir un monde tout neuf : en effet, il est possible, en principe, d'y connecter jusqu'à 128 composants I²C au maximum !



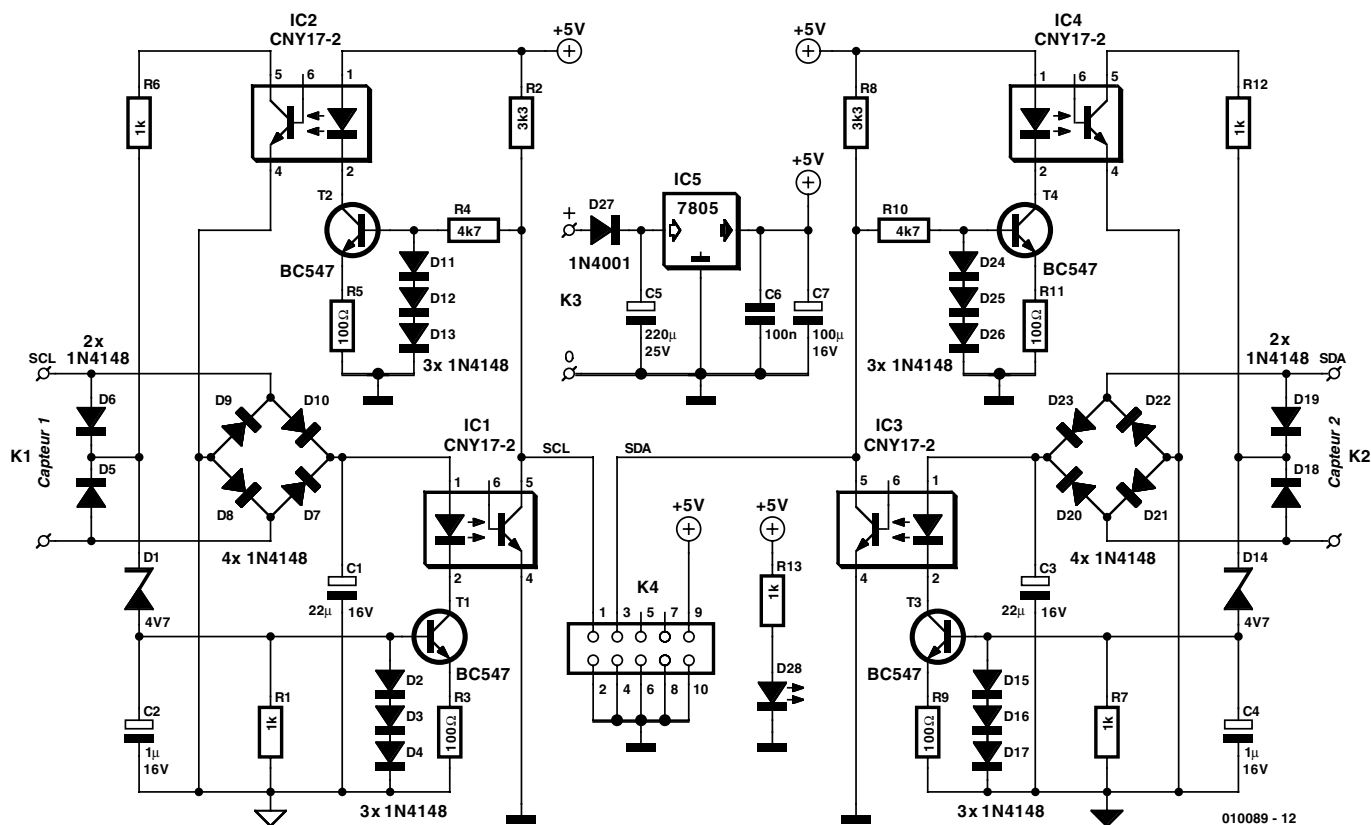


Figure 1. L'interface I²C pour RCX de Lego comporte 2 sous-ensembles identiques pour les signaux SCL et SDA respectivement.

Le RCX, le composant « intelligent » du système *Robotics Invention* de Lego qui, lui, fait partie du programme *Mindstorms* de Lego, convient parfaitement pour servir de base pour nombre d'applications automatisées en tous genres. Il devient ainsi possible, si l'on fait abstraction des efforts de programmation qui restent indispensables, de créer, à l'aide de composants Lego, un petit robot. Cependant, quelles que soient la puissance et la modularité du concept du module RCX, on se heurte inévitablement à une barrière infranchissable dès lors que l'on a besoin, pour une régulation ou un processus de mesure quelconque, de plus d'entrées et/ou de sorties que les 3 de chaque sorte dont dispose ce type de module...

Ce n'est d'ailleurs pas la première fois que nous signalons cette –permettez-nous de la qualifier ainsi– insuffisance du concept. En avril 2000 nous avons démarré une série de 5 articles [1] dans laquelle nous vous présentions le *Robotics Invention System* de Lego. À l'époque déjà, nous avions tenté de connecter

plusieurs capteurs à une même entrée. Dans le numéro double de la même année [2], nous vous proposons une astuce permettant la connexion de plusieurs interrupteurs sur une seule entrée.

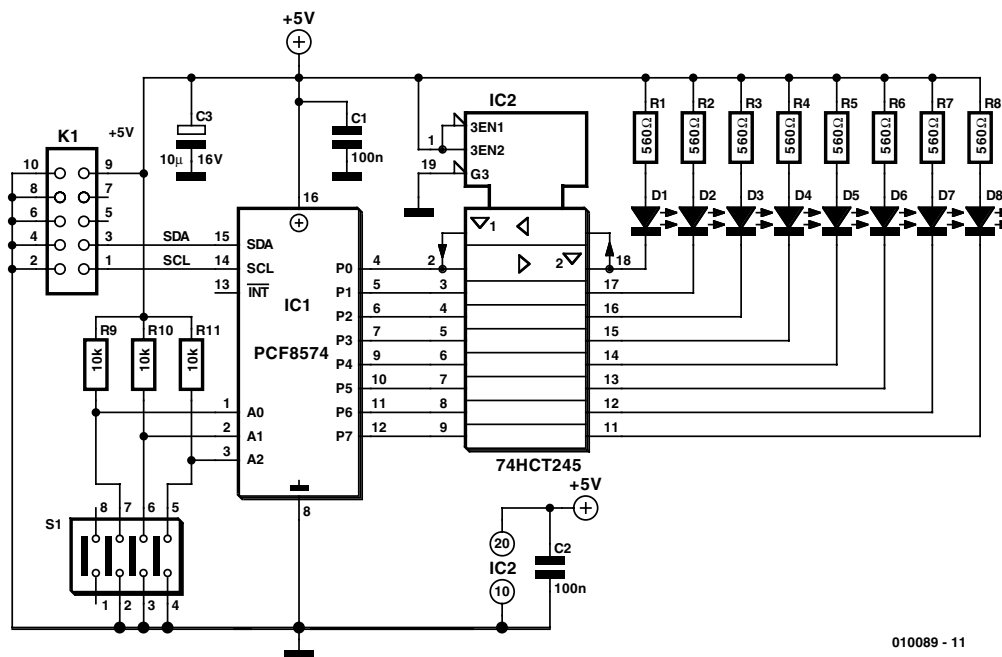
Il va sans dire que nous ne sommes pas les seuls, à la rédaction d'Elektor, à avoir remarqué que les possibilités d'entrées/sorties (I/O) du module RCX étaient relativement limitées. Nombre d'amateurs enthousiastes ont imaginé des réalisations permettant, par le biais d'un multiplexage, de permettre la connexion d'un nombre plus important de périphériques, terme générique désignant tout dispositif connecté à un système d'E/S. Nous y avons découvert nombre de solutions astucieuses qui peuvent constituer une solution à l'une ou l'autre situation donnée. Le montage que nous vous présentons ici nous paraît être plus charmant encore. Grâce à lui en effet, le module RCX est en mesure de communiquer, par le biais d'un bus I²C, avec pas moins, théoriquement du moins, de 128 composants au maximum !

Fonctionnement de l'interface

Le bus Inter-IC (I²C) est un bus dit bifilaire (à 2 fils) requérant une ligne de données (SDA = *Serial Data*) bidirectionnelle et un signal d'horloge (SCL = *Serial Clock*). Ce bus respecte un certain nombre de règles. La transmission des données se fait entre un système dit « maître » (*master*) et un ou plusieurs systèmes « esclaves » (*slave*). Le maître est le composant qui démarre la communication et c'est à lui de fournir le signal d'horloge.

Il est en outre interdit aux composants de mettre d'activer au niveau haut (« 1 ») les lignes du bus. L'obtention d'un niveau haut se fait par la mise à haute impédance de la ligne concernée, une résistance de 3kΩ3 reliée à la ligne positive de l'alimentation assurant alors la présence d'un « 1 » logique sur le bus. À l'inverse, l'activation à un niveau bas pour rendre un « 0 » est parfaitement autorisée. La résistance de 3kΩ3 se trouve alors forcée à la masse.

Le module RCX doit être en mesure de mettre tant des « 1 » que des « 0 » sur le bus, mais doit également pouvoir lire ces 2 états. Une astuce connue dans le « monde » du RCX il est possible d'y arriver par le biais d'une unique entrée de capteur. Il est en effet pos-



010089 - 11

Figure 2. Circuiterie de test avec circuit intégré d'expansion de port I²C et circuit de commande pour 8 LED.

sible, par logiciel, de configurer une entrée de capteur en mode actif ou passif. En mode passif la valeur de mesure (= la tension présente à l'entrée) est déterminée par une résistance interne et la résistance du capteur connecté (à l'entrée en question). La tension présente au niveau de ce diviseur de tension peut varier entre 0 et 5 V. Au contraire, si l'on met l'entrée de capteur en mode actif, on a, entre les différentes mesures, mise de la ligne du capteur en question à la tension de la pile (entre 7 et 9 V). Périodiquement, toutes les 3 ms environ, ce niveau de tension plus élevé est découplé de la ligne et la valeur de tension présente sur l'entrée déterminée normalement. Il faut de l'ordre de 0,1 ms pour effectuer une mesure.

Il est partant possible, de cette façon, par une commutation logicielle entre des capteurs actifs et passifs, d'obtenir un signal allant du module RCX vers le bus I²C. Il est de plus possible, par le biais de la même entrée de capteur, de lire (prendre en compte) le signal en provenance du bus.

L'électronique

L'interface nécessaire est identique pour les 2 signaux SDA et SCL. On reconnaît de ce fait, sur le schéma du montage représenté en **figure 1**, 2 fois la même électronique. Nous nous limiterons pour cette raison à la description du fonctionnement pour l'un des 2 signaux seulement.

Des opto-coupleurs subdivisent le circuit en 2 parties. Cette précaution permet d'éviter

tout risque d'endommagement tant du module RCX que des systèmes (devices) qui y sont connectés s'il devait naître, à un endroit ou un autre, des tensions de niveau trop important (au cas, par exemple, où l'on n'aurait connecté qu'une seule des lignes d'alimentation). Le sous-ensemble côté « I²C » est en effet alimenté par une source d'énergie propre. C'est là la tâche de IC5. La diode D27 sert de protection contre une inversion malheureuse de polarité de la tension d'alimentation, évitant ainsi tout dégât dans le cas d'une erreur de ce genre. La LED D28 fait office de dispositif de signalisation de la présence (ou non si elle est éteinte !) de la tension d'alimentation.

Le sous-ensemble du RCX est alimenté par le biais des diodes D7 à D10. Les diodes constituent un pont de redressement, ce qui permet de ne pas avoir à se poser de question de polarité lors de connexions aux modules RCX. C1 tamponne la tension d'alimentation aux instants où se font les mesures. Comme nous le disions plus haut, dans le mode *actif*, la pile est en effet déconnectée à intervalles plus ou moins réguliers. La tension en provenance de la connexion du capteur est, pour sa part, fournie au montage au travers des diodes D5 et D6 –cette paire de

diodes supprime tout problème de polarité. La combinaison constituée par la diode zener D1 et les diodes « standard » D2 à D4 permet de faire la différence entre les modes *passif* et *actif*. Lorsque l'on se trouve en mode passif, la diode zener n'est pas passante. Cependant, dès application de la tension de pile, le condensateur C2 se charge ce qui se traduira par la présence d'une tension de 1,8 V maximum aux bornes de la combinaison de diodes D2 à D4.

C2 remplit une seconde fonction, il assure le pont lors des coupures de la tension ayant lieu pendant les mesures périodiques. En présence de ces 1,8 V, le transistor T1 est passant de sorte qu'il circule un courant au travers de la LED de l'opto-coupleur. R1 permet une décharge suffisamment rapide du condensateur lors du retour en mode passif. Cette approche reposant sur une séparation non pas de l'Église et de l'État mais de l'alimentation et du circuit de commutation permet la mise sur le bus de « 1 » ou de « 0 » définis avec une excellente précision.

La circuiterie présente en aval de IC1 est élémentaire et se résume en fait à la résistance R2. Cette résistance a pour fonction d'assurer un niveau haut sur le bus lorsque la ligne concernée se trouve à haute impédance. Lorsque la LED est allu-

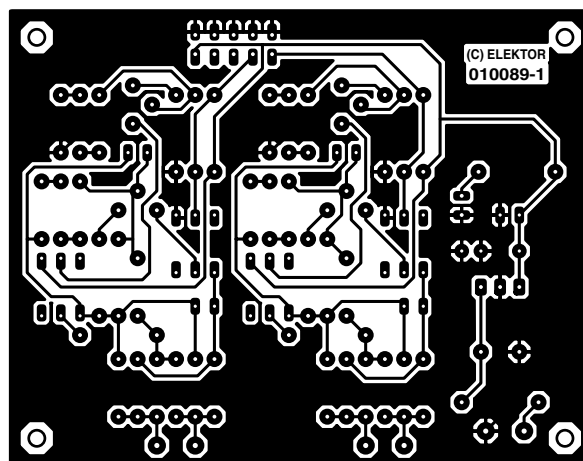
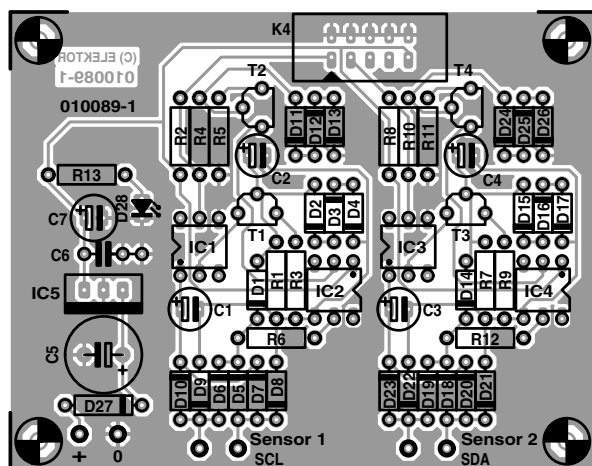


Figure 3. Dessin des pistes et sérigraphie de l'implantation des composants de l'interface I²C.

Liste des composants de l'interface

Résistances :

R1, R6, R7, R12, R13 = 1 kΩ
R2, R8 = 3 kΩ
R3, R5, R9, R11 = 100 Ω
R4, R10 = 4 kΩ

Condensateurs :

C1, C3 = 22 μF/16 V radial
C2, C4 = 1 μF/16 V radial
C5 = 220 μF/25 V radial
C6 = 100 nF

C7 = 1 000 μF/16 V radial

Semi-conducteurs :

D1, D14 = diode zener 4V7/400 mW
D2 à D13, D15 à D26 = 1N4148
D27 = LED faible courant
T1 à T4 = BC547
IC1 à IC4 = CNY17-2
IC5 = 7805

Divers :

K1 à K3 = 6 picots
K4 = embase mâle à 2 rangées de 5 contacts (HE-10)

mée la partie « transistor » de l'opto-coupleur est en conduction et le bus présente un niveau bas. On constate que le circuit travaille en inverseur, situation dont le logiciel doit tenir compte.

La prise en compte du signal en provenance du bus par le module RCX est l'affaire de toute une série de composants, de la résistance R4, les diodes D11 à D13, la paire transistor T2 + résistance R5, ainsi que IC2 et R6. Le niveau logique présent sur la ligne SCL ou SDA arrive, par le biais de R4, sur la base de T2 et la triplette de diodes, D11 à D13. Ces diodes limitent la tension appliquée à la base de T1 à 1,8 V. Lorsque le niveau présent sur le bus est un niveau haut, le transistor T2 devient conducteur et la LED intégrée dans l'opto-coupleur IC2 s'allume. La LED dérive son courant d'alimentation de l'alimentation séparée prévue pour la partie I²C.

Côté « RCX » de IC2, le transistor intégré dans l'opto-coupleur fera en sorte, par le biais de la résistance R6, que la tension présente sur l'entrée du capteur corresponde à un niveau faible lorsque la LED est allumée. De par la présence des diodes D5 et D6, il n'y a pas d'inquiétude à se faire, une nouvelle fois, quant à la polarité lors du branchement.

Liste des composants de la platine de test

Résistances :

R1 à R8 = 560 Ω
R9 à R11 = 10 kΩ

Condensateurs :

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

Semi-conducteurs :

D1 à D8 = LED rectangulaire
IC1 = PCF8574
IC2 = 74HCT245

Divers :

K1 = embase mâle à 2 rangées de 5 contacts (HE-10)
S1 = interrupteur DIP triple (un quadruple fait également l'affaire)

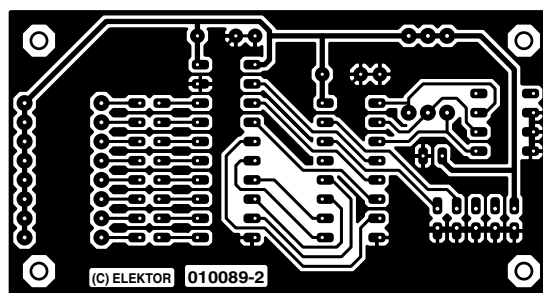
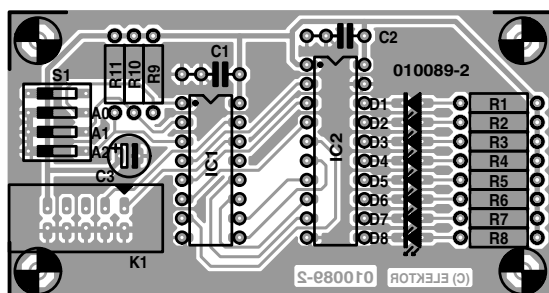


Figure 4. Dessin des pistes et sérigraphie de l'implantation des composants de la platine de test.

Le premier appareil

Il faut bien évidemment disposer, pour pouvoir tester l'interface, d'un appareil à connecter au bus. Nous allons, à cet effet, utiliser le circuit représenté en **figure 2**. IC1, un composant connu sous la dénomination de « *I²C-port-expander* » (expanseur de port I²C) constitue le coeur de cette électronique. Cet « appareil » reçoit une valeur de programmation de 8 bits transmise par le bus, valeur représentée ensuite sous forme binaire aux sorties. Supposons que vous envoyiez le chiffre « 1 » vers le circuit intégré en question; cela se traduira par le passage au niveau haut de la broche P0. En cas d'émission du nombre 17 ce sera le tour aux lignes P4 et P0 de présenter un niveau haut ($10001_B = 17_D$).

Les sorties de IC1 sont reliées, au travers de IC2, un circuit de commande (driver) 8 ports, à 8 LED. Dans le cas d'une autre application ces organes à commander pourront être n'importe quels autres indicateurs ou actionneurs, tels que résonateurs (*zoomer*) ou relais (ces derniers requérant la présence d'un étage-tampon). L'expanseur de port est en outre capable de prendre en compte une valeur à 8 bits placées sur les lignes P0 à P7. Il faudra, pour cela, que le maître commence par mettre les 8 lignes au niveau haut, les lignes pouvant ensuite être activées, l'une indépendamment de l'autre, au niveau bas.

Lors d'une opération de lecture la valeur prise en compte est ensuite fournie. Notons au passage qu'il est possible de connecter un maximum de 8 expanseurs de port au bus. L'adresse de ce circuit intégré est en effet du type 0100xxx0, nombre dans lequel les états (« 0 » ou « 1 ») de xxx pourront être définis par le biais des contacts du triple interrupteur DIP S1.

La réalisation

L'étape de construction de ces 2 platines ne devrait pas poser de problème, surtout si vous avez gravé vos propres circuits imprimés à partir des dessins des pistes représentés en **figures 3 et 4**. Signalons quelques aspects auxquels ils faudra, en cours de réalisation, porter attention : respect de la polarité des composants polarisés tels que diodes, condensateurs, transistors et autres circuits intégrés. On pourra envisager de placer les opto-coupleurs dans un support pour circuit intégré. Un coup d'oeil à la liste des composants permet de constater que nous avons opté, en ce qui concerne D28 de la platine d'interface, pour une LED à faible courant. Cette caractéristique est importante sachant que cette LED devra être alimentée directement par le bloc RCX. Toutes les mesures ser-

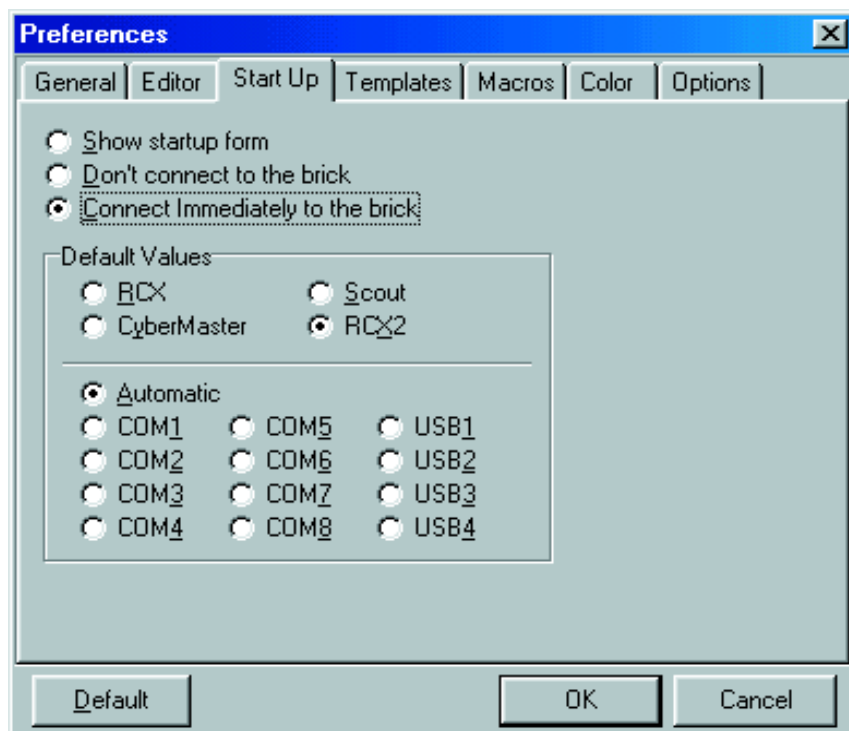


Figure 5. Il est important, après lancement de BricxCC, de sélectionner RCX2.

vant à économiser les piles sont bien entendu les bienvenues.

La liaison vers le monde extérieur se fait par le biais de picots et d'embases HE-10. On pourra souder aux bornes K1 et K2 des mini-câbles Lego standard. K3 recevra l'adaptateur secteur 9 V externe.

L'embase HE-10 K4 sert à « sortir » le bus I²C vers l'extérieur. On trouvera sur ce connecteur non seulement les lignes SCL et SDA (broches 1 et 3 respectivement) mais également la tension d'alimentation (broche 9) et bien entendu la masse (broches 2, 4, 6, 8 et 1). Le fait d'avoir prévu également la tension d'alimentation sur cette embase permet, par le biais d'un morceau de câble en nappe (si tant est que le courant drainé par la charge concernée reste dans des limites acceptables), la connexion de la platine de test, elle aussi dotée d'une embase à 10 contacts du même type, sans qu'il ne soit partant nécessaire de prévoir d'alimentation propre pour la platine de test; la charge que constituent les 8 LED ne pose pas le moindre problème.

Le logiciel

Il va sans dire que le programme associé à cette réalisation constitue

une partie essentielle de l'interface. Elle comporte un certain nombre de fonctions qui permettent la transmission de données à un niveau plus haut avec des composants I²C. Ces fonctions ont été écrites en NQD 2.2 (NQD = *Not Quite C*). Pour pouvoir utiliser le matériel nous avons travaillé avec le progiciel (*firmware*) en version 2.0. (le système ne fonctionne pas avec des versions de progiciel antérieures !). Si votre module RCX n'est pas « équipé » de cette version et que vous tenez à mettre en oeuvre les fonctions I²C décrites ici, il vous faudra commencer par réactualiser (upgrade) le progiciel. Voici comment vous y prendre : il faudra commencer par télécharger le progiciel V2.0 (le « logiciel système ») pour le module RCX (cf. le lien à utiliser en fin d'article).

On transfère ensuite la nouvelle version de progiciel dans le RCX. On pourra utiliser à cet effet le programme BricxCC. Une fois dans le programme BricxCC, il faudra paramétrer RCX2 et non pas RCX. Lors du premier démarrage du programme on verra se poser cette question. On arrivera ensuite aux paramètres par le biais du menu *Edit -> Preferences* situé en regard du point de menu *Start Up*.

BricxCC est d'ailleurs une programme qui assure la communication entre le PC et le module RCX, non seulement pour le chargement d'une nouvelle version de progiciel, mais également pour la transmission de programmes NQC. NQC est un compilateur capable de traduire des programmes de type C à l'intention du module RCX.

Rien de tel qu'une paire d'exemples de programmes pour illustrer les fonctions chargées d'assurer la communication avec le bus.

Nous allons commencer par nous intéresser à **i2c_test_wr.nqc**, programme que l'on retrouve, ainsi que les autres exemples, sur la disquette associée à cet article (**EPS010098-11**) disponible auprès des adresses habituelles de même d'ailleurs que dans la zone de téléchargement correspondante du site d'Elektor (www.elektor.presse.fr). Le programme procède à l'incrémentation d'un compteur jusqu'à ce que ce dernier ait atteint la valeur 255. Chaque nouvelle valeur est envoyée à la platine de test par le biais du bus. Nous rencontrons, successivement, les lignes suivantes :

```
i2c_init();
i2c_start();
i2c_data = PCF8574_0;
i2c_send();
i2c_data = test;
i2c_send();
i2c_stop();
```

L'instruction `i2c_start()` démarre une mission I²C. On envoie ensuite l'adresse du composant avec lequel on veut entrer en communication. Il s'agit en l'occurrence ici de l'expandeur de port. On envoie ensuite les données proprement dites par le biais de la variable `i2c_data`. Il ne reste plus ensuite qu'à clore la communication

Dans notre programme d'exemple on procède ensuite à une incrémentation de la valeur de la variable « test » et on exécute à nouveau la boucle. Cette approche fonctionne correctement mais il n'est pas nécessaire, en principe, d'interrompre la communication tant que l'on s'adresse au même composant. De ce fait, l'initialisation, le démarrage, l'émission de l'adresse et l'arrêt peuvent dans ce cas-là s'effectuer en-

dehors de la boucle. En tout état de cause, cette solution permet un gain de vitesse d'exécution.

Dans le second exemple, **i2c_test_rd.nqc**, on procède à la lecture du circuit intégré de l'expandeur de port. Avant de pouvoir effectuer cette opération il faudra commencer par établir la communication de façon classique :

```
i2c_init();
i2c_start();
i2c_data = PCF8574_0;
i2c_send();
```

Il faut, pour pouvoir les utiliser en entrée, que toutes les lignes soient forcées au niveau logique haut. Pour ce faire, on envoie la valeur 0xFF (255) vers le composant concerné :

```
i2c_data = 0xFF;
i2c_send();
```

Ceci fait on peut passer à l'étape de lecture du port proprement dite :

```
i2c_recv();
test= i2c_data;
```

Il reste à clore la communication :

```
i2c_stop();
```

Le programme d'exemple présente une structure légèrement différente pour la bonne et simple raison qu'il comporte, lui aussi, une boucle. Notons que les 2 programmes ne s'ennuient pas avec la vérification ou le positionnement du bit d'acquiescement (*acknowledge*) **i2c_ackn**. Ce bit envoyé par un esclave peut être reçu par le système-maître après que ce dernier ait envoyé des données vers l'esclave. Inversement, il est également possible d'envoyer un bit d'acquiescement vers un dispositif esclave lorsque le maître a pris en compte les données envoyées par celui-ci. Cette précaution peut être de mise pour certaines applications mais n'est pas nécessaire dans le cas présent.

Il apparaît qu'avec ce logiciel, le bloc RCX et l'interface peuvent atteindre une vitesse de 2 messages I²C complets par seconde environ. Rien de très impressionnant, caractéristique dont il faudra peut-être tenir compte pour certaines applications critiques.

L'implémentation proprement dite des fonctions utilisateur se trouve dans le fichier `i2c_master`. Lorsqu'un programme fait appel à ces fonctions il faudra que ce fichier soit ajouté à l'aide de la directive de compilateur **#INCLUDE**. C'est à dessein, pour ne pas trop compliquer le sujet, que nous n'entrons pas dans le détail des fonctions de bas niveau. Ceux d'entre nos lecteurs que le sujet intéresse pourront procéder à l'examen du code-source. La seule remarque que nous tenons éventuellement à ajouter à ce sujet est que la compensation de l'inversion introduite par l'interface se fait à ce niveau. Il est également possible, dans le dit fichier, de déterminer quelle entrée de capteur représente, en finalité, un signal donné présent sur le bus. En standard, on a choisi SCL pour le capteur 1 et SDA pour le capteur 2. De plus, les différentes adresses possibles pour le PCF8574 (le circuit intégré expandeur de port de notre circuit de test) sont déjà paramétrées sous la forme PCF8574_0 à PCF8574_7 (0x40h à 0x4Eh). Signalons cependant que cela ne signifie en aucun cas que l'interface ou le logiciel soient limités à ces seuls types et domaines. Vous pourrez, dès à présent, mettre en oeuvre d'autres composants I²C tels que convertisseurs A/N ou N/A (le PCF8591 par exemple), circuits de commande de relais et de LED (le SAA1064 par exemple) et même un circuit d'horloge en temps réel tel que le PCF8583 pourra être utilisé avec le module RCX de Lego !

(010089)

Littérature :

- [1] **Lego Robotic System**,
Elektor n° 262, avril 2000, page 38
- [2] **Entrée analogique pour le module RCX de Lego**,
Elektor n° 265/266, juillet/août 2000, page 76

Pour de plus amples informations :

Bricx Command Center 3.3

<http://hometown.aol.com/johnbinder/briccc.htm>

Not Quite C

<http://www.cs.uu.nl/people/markov/lego>

Officiele NQC (download)site

<http://www.enteract.com/~dbaum/nqc/doc>

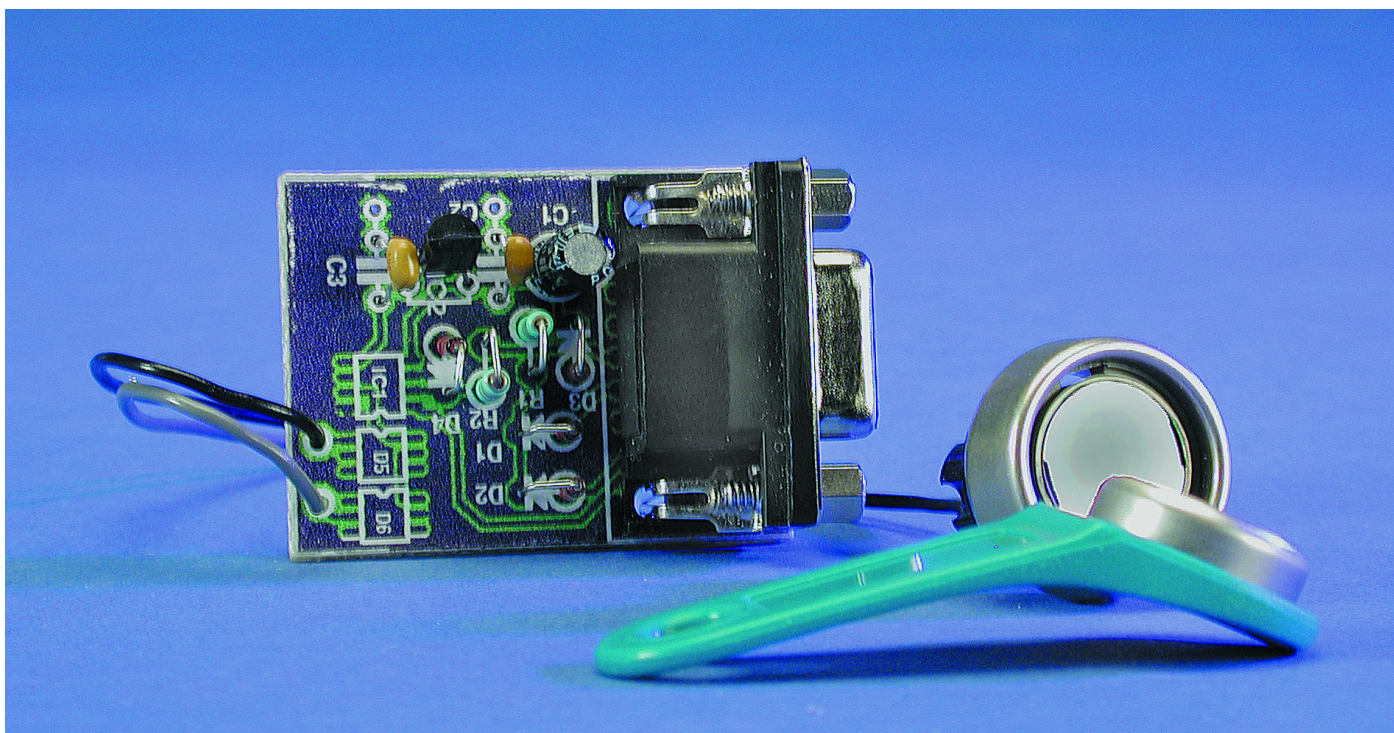
RCX firmware 2.0

<http://mindstorms.lego.com/sdk2/>

Interface série pour puces 1-Wire

Accès aisé sous Windows

Les puces pour le bus à 2 fils (1-Wire) de Dallas Semiconductor sont capables d'entrer en relation entre elles, avec un autre circuit ou un ordinateur, à travers une liaison série très simple. Avec l'interface RS-232 décrite ici, rien de plus facile que de causer avec un PC, d'autant que Dallas fournit gracieusement le logiciel.



Dallas Semiconductor a développé toute une série de circuits pour liaison à fil unique (1-wire, à condition d'oublier de compter la masse !) et nous n'avons pas manqué d'y faire

écho dans nos colonnes, pensons par exemple aux projets **e-Key** [1] en novembre 2000 et **OneWireSpy**, **moniteur pour puces 1-Wire** [2] de

juin 2000. Dans ces réalisations, la communication avec le PC était opérée, sur un bus à deux fils, par un microcontrôleur programmé à cet

effet. Dallas propose même des circuits intégrés spéciaux qui prennent en charge totalement la mise en liaison de microcontrôleurs et d'ordinateurs. Les pilotes, le logiciel d'évaluation et de conception sont disponibles gratuitement sur le site Web de Dallas, ce qui facilite encore la mise en œuvre de ces puces. Le « 1-Wire SDK », le kit de développement de logiciel pour bus à deux fils, s'accorde avec différentes versions de Windows : Me, 2000, 98, 95, NT 4.00 et NT 3.51. Il comporte des exemples de programmation en C, C++, Pascal (Borland Delphi), Microsoft Access et Microsoft Visual Basic. Il dispose également de quelques outils de conception pour les applications à 16 bits (Windows 3.1 et DOS).

Nouveaux composants

L'offre en composants pour bus à deux fils est vaste. Nous vous avons déjà présenté dans nos publications les IButtons, ces composants qui ressemblent à des piles boutons. Mais la série compte aussi des potentiomètres numériques, capteurs de température, montres, commutateurs, convertisseurs A/N, identités électroniques et mémoires, disponibles en boîtier TO-92 ou version CMS. Chacun d'eux possède son propre code de famille (selon le type) et un numéro de série unique, de manière à les identifier et les adresser sur le bus. Le **tableau 1** reprend la liste des composants pour ce bus actuellement disponibles.

L'atout principal du bus à deux fils, c'est la simplicité avec laquelle on peut se bâtir de petits réseaux, pour connaître la température dans différentes pièces d'une habitation, enclencher ou déclencher des circuits, actionner des gradateurs, etc. Le matériel est simple. Deux fils assurent l'alimentation et la communication. À l'aide de l'ensemble des logiciels de développement que Dallas a mis à disposition sur l'Internet, tout un chacun peut se lancer dans l'aventure. Il faudra bien sûr le connecter à un PC, mais avec les interfaces intégrées conçues pour ce bus, cela ne pose plus problème. Cet article va décrire une interface série RS-232, mais très bientôt, nous vous présenterons une interface pour l'USB.

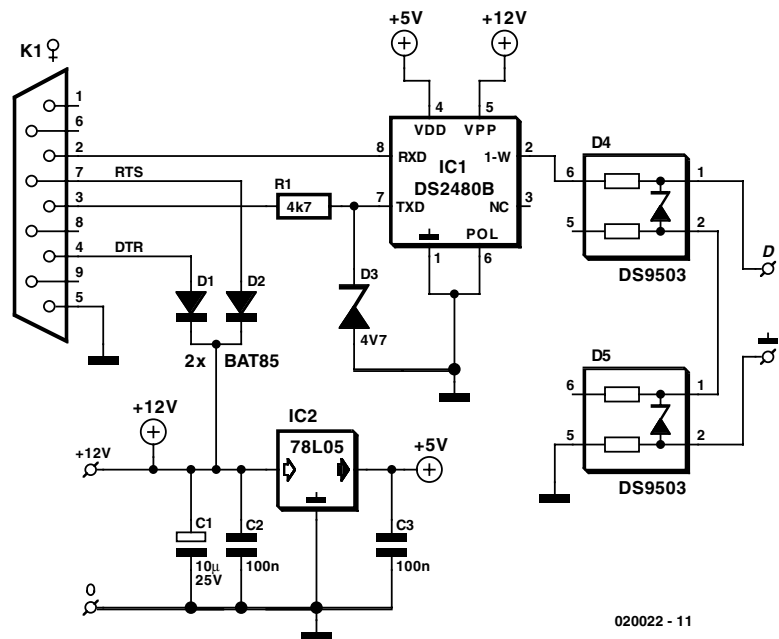


Figure 1. Le montage d'interface se base, en gros, sur deux circuits intégrés de Dallas.

Au cœur de l'interface : le DS2480B

Petite par ses dimensions (le boîtier SO-8), cette puce est grande par ses possibilités. C'est en fait un composant assez complexe, mais comme le logiciel de Dallas est là, nous n'aurons pas à nous tracasser énormément à propos de sa cuisine intérieure. Le lecteur qui, malgré tout, veut savoir ce à quoi joue réellement le DS2480B peut se plonger à loisir dans les 30 pages du feuillet de caractéristiques. Pour notre part, nous préférons, au fil de ces colonnes, le considérer comme une boîte noire et nous concentrer sur les composants présents sur le bus à deux fils, la manière dont ils communiquent avec le PC nous intéresse beaucoup moins.

Matériel

Pour une interface série entre le PC et le bus à deux fils, il nous faut peu de choses, le DS2480B se coltine tout le boulot, de connivence avec le pilote présent dans le logiciel. La **figure 1** nous dévoile le schéma complet, et encore, il s'agit là de la version la plus étendue qui se puisse imaginer. Dans la majorité des applications plus élémentaires, on peut encore élaguer à gauche et à droite.

taires, on peut encore élaguer à gauche et à droite.

Interface série

La puce d'interface communique avec le PC par les bornes RxD et TxD du port RS-232. Comme chacun sait, la norme prévoit qu'un « 1 » logique correspond à -12 V, un zéro logique à +12 V, mais que les protocoles sériels pour microcontrôleurs utilisent à la place +5 V et 0 V, respectivement. Avec la broche 6 (POLarity) de la puce à la masse, les signaux sériels sont inversés dans le DS2480B pour satisfaire à la norme RS-232, avec le secours d'une fixation de niveau, par R1 et D3, pour le signal TxD sur la broche 7. La résistance et la diode zener font en sorte que la tension sur cette broche soit limitée entre 0 et 4,7 V.

La plupart (mais pas tous !) des ports RS-232 modernes se contentent d'un 0 V pour le « 1 » logique et de +5 V pour un « 0 », ce qui permet de faire l'impasse sur le matériel supplémentaire pour se conformer exactement au standard (-12 V et +12 V).

Le bus à deux fils

Comme le nom anglais (1-Wire) le dit, un seul « fil » porteur de signal sort du DS2480 B pour assurer toute la communication. La plupart des puces de la série « 1-Wire » peuvent recevoir leur alimentation par l'intermédiaire de ce bus. À côté d'elles, il y a les mémoires programmables (EPROM, ROM programmables électriquement, à ne pas confondre

donc avec les composants à lucarne !) ont besoin d'impulsions de programmation de 12 V pour l'enregistrement des données. Cette tension est disponible sur l'interface, sur la broche 5 du DS2480B et mise sur le bus automatiquement, si une opération de programmation est en cours.

Dallas Semiconductor prévient avec insistance, une seule EPROM peut être présente sur le bus, raccordée par un câble de faible longueur, au moment de la programmation. En outre, aucun autre composant sans EPROM ne peut y être raccordé lors de l'exécution d'une commande de programmation, sous peine de l'endommager, lui ou le DS2480B !

D4 et D5 assurent la protection de l'interface en cas de décharge électrique disruptive (ESD), elle ne craint donc pas les charges électrostatiques éventuellement amenées par le bus. Le DS9503 possède une diode zener intégrée dont la tension d'avalanche fait envi-

ron 7,5 V. Les résistances intégrées de 5 Ω n'ont qu'un effet négligeable pendant une communication normale, mais présentent une grande impédance pour la diode zener, si une décharge se produit par le bus. Aucune tension dangereuse ne pourra donc atteindre la broche 2 du CI d'interface. Si l'interface ne doit pas servir à programmer d'EPROM, il n'y aura pas d'impulsion de 12 V, la tension de service sur le bus ne sera pas supérieure à 5 V et D5 n'aura plus d'utilité. La broche 5 de D4 est alors reliée au plan de masse de la platine, la prise de terre de K2 à l'anode (broche 2 de D4).

Alimentation

L'alimentation peut se faire à partir du port sériel RS-232, mais ce ne

sera pas suffisant, sur le bus à deux fils, avec des mémoires qui réclament une tension de programmation de 12 V. Dans ce cas, on branche sur K3 une alimentation stabilisée externe de 12 V. Pour toute sécurité, souvenez-vous des recommandations énoncées dans le paragraphe sur le bus.

Dans les autres cas, les lignes DTR et RTS de l'interface peuvent fournir l'alimentation. Si l'une d'elle est au niveau haut (+12 V), la marge est suffisante pour en faire un +5 V convenable.

Construction et test

La figure 2 décrit la platine conçue pour cette interface.

Les DS2480B et DS9503 sont des

Tableau I. Panoramique sur les composants « I-Wire » actuellement disponibles.

Composant	Fonction	Mémoire
DS1820	Thermomètre Numérique	16 bits d'EEPROM
DS18B20	Thermomètre à résolution programmable	16 bits d'EEPROM
DS18S20	Thermomètre de précision	16 bits d'EEPROM
DS1821	Thermostat autonome	2 octets de NV
DS1822	Thermomètre à résolution programmable	pas de NV
DS2401	Numéro de série	Pas de mémoire additionnelle
DS2404	EconoRAM Time Chip	4 096 bits de RAM
DS2404S-C01	Dual-Port Memory Plus Time	4 096 bits de RAM
DS2405	Commutateur adressable	Pas de mémoire additionnelle
DS2406	Double commutateur adressable	1 024 bits d'EPROM
DS2409	Coupleur MicroLAN	Pas de mémoire additionnelle
DS2417	Puce chrono (time-chip) avec Interruption	Compteur temps réel à 32 bits
DS2423	I-Wire RAM avec compteur	4 096 bits de RAM
DS2430A	EEPROM I-Wire	256 + 64 bits d'EEPROM
DS2433	EEPROM I-Wire	4 096 bits d'EEPROM
DS2434	Thermomètre	32 octets d'EEPROM, 32 octets de SRAM
DS2435	Thermomètre /Histogramme température-temps	32 octets d'EEPROM, 32 octets de SRAM
DS2436	Thermomètre, Tension A/	32 octets d'EEPROM, 8 octets de SRAM
DS2437	Tension A/N, « Full state », temps réel, température	40 octets d'EEPROM
DS2438	Tension A/N, « Full state », temps écoulé, température	40 octets d'EEPROM
DS2450	I-Wire pour convertisseur A/N	Mémoire d'état/de commande uniquement
DS2480B	Circuit de commande de puissance I-Wire	Mémoire d'état/de commande uniquement
DS2490	Pont USB vers I-Wire	Commande de mode et FIFO d'E/S
DS2502	Mémoire additionnelle seule	1 024 bits d'EPROM
DS2502-UNW	UniqueWare	1 024 bits d'EPROM
DS2502-E64	Puce d'adresse IEEE EUI à 64 noeuds	256 bits préprogrammés, 768 bits programmables par l'utilisateur
DS2505	Mémoire additionnelle seule	16 384 bits d'EPROM
DS2505-UNW	UniqueWare	16 384 bits d'EPROM
DS2506	Mémoire additionnelle seule	65 536 bits d'EPROM
DS2506-UNW	UniqueWare	65 536 bits d'EPROM
DS2890	Potentiomètre numérique I-Wire	Mémoire pour caractéristiques et commande
DS9502	Diode de protection CES	-
DS9503	Diode de protection CES avec résistances	-

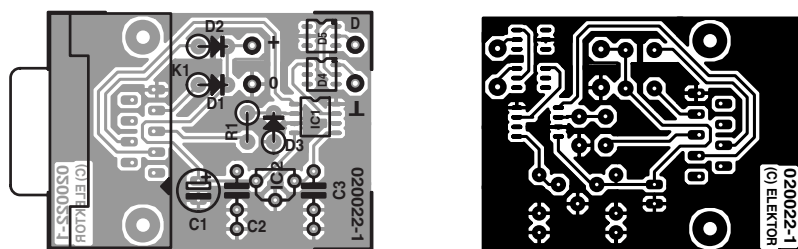


Figure 2. Avec des boîtiers CMS, la platine est minuscule, mais demande de l'attention lors du soudage !

Liste des composants

Résistances :

R1 = 4kΩ7

Condensateurs :

C1 = 10 μF/25 V radial

C2, C3 = 100 nF

Semi-conducteurs :

D1, D2 = BAT85

D3 = diode zener 4V7/400 mW

D4, D5 = DS9503 (Dallas Semiconductor 6 broches SOC)

IC1 = DS2480BS (Dallas Semiconductor 8 broches SOIC)

IC2 = 78L05

Divers :

K1 = embase Sub-D à 9 contacts femelle encartable en équerre

CMS et nous conseillons de les souder en priorité sur la face cuivrée. Il faut bien identifier le repère de la broche 1, une encoche. On étame la pastille correspondant à un coin du CI, avant d'y souder la patte, puis on fait de même dans le coin opposé en

diagonale. On vérifie que toutes les autres pattes s'appliquent exactement sur le cuivre et c'est alors seulement qu'on les soude. La soudure excédentaire peut être évacuée à l'aide de tresse à dessouder. C'est ensuite au tour des résistances

et des diodes montées verticalement côté composants, pour finir par le régulateur de tension IC2, les condensateurs et les connecteurs.

Il s'agit alors d'inspecter à fond le travail, en particulier pour déceler d'éventuels ponts de soudure entre les broches des CMS.

Pour le test du matériel, nous téléchargeons d'abord iButton Viewer avec les pilotes TMEX à l'adresse www.ibutton.com/software/tmex/index.html, la version la plus récente actuellement porte le numéro 3.20. Le fichier TM320_32.EXE installe les pilotes et le logiciel, puis lance aussitôt la visionneuse iButton, laquelle va se mettre à la recherche de l'interface « 1-Wire ».

On raccorde le montage à l'aide d'un cordon sériel normal à 9 broches (PAS un câble *null modem*, donc !) à un port COM libre du PC. L'alimentation externe sur K3 n'est pas nécessaire, pas plus que de composant sur le bus à deux fils. Si tout va bien, notre interface sera détectée après une pression sur le bouton Auto Detect. L'installation est alors terminée et il faut relancer la visionneuse iButton. Un message d'erreur apparaît immédiatement si le programme n'a pas réussi à communiquer avec l'interface. À ce stade, c'est sûrement le matériel qu'il faut incriminer : les soudures, l'orientation des puces, d'un condensateur électrolytique ou d'une diode.

Si tout fonctionne comme prévu, le logiciel nous renseigne un adaptateur DS9097U sur le port COM, c'est l'interface à bord du DS2480. La communication entre PC et interface est établie et nous pouvons enfin travailler avec le bus à deux fils, ce qui était bien notre objectif initial.

Composants sur bus à deux fils

Puisque nous avons accès au nouveau bus, c'est le moment d'y brancher l'un ou l'autre intervenant. Mais au préalable, mieux vaut lire le feuillet de caractéristiques, parce que certains d'entre eux ne se contentent pas des deux fils, il leur faut encore quelques composants externes, comme un quartz pour l'horloge DS2417, ou leur propre alimentation stabilisée.

Si vous n'avez pas d'emblée l'humeur à programmer, vous pouvez utiliser la visionneuse iButton, qui dispose de ce qu'il faut pour détecter et annoncer quels composants sont présents sur le bus à deux fils. Vous pouvez voir sur la **figure 3** la fenêtre d'accueil de la visionneuse. Dans la colonne de gauche, il y a les numéros d'identification individuels des cinq composants trouvés sur le bus. À droite, une courte description du composant sélectionné, dans ce cas-ci, un DS2450, un convertisseur A/N à quatre canaux.

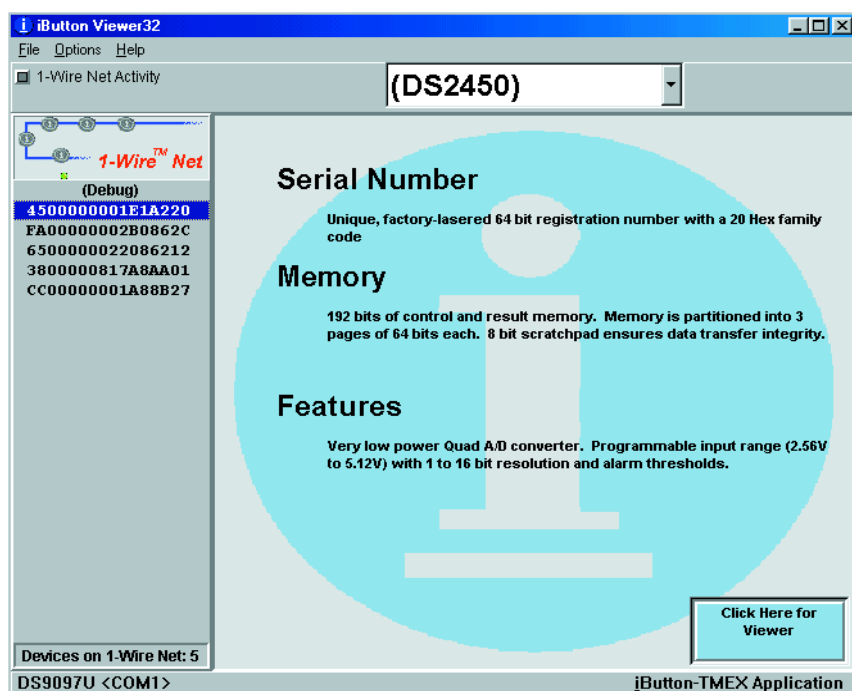


Figure 3. La fenêtre principale de la visionneuse iButton.

Littérature :

- [1] **e-Key**,
Elektor n°269,
novembre 2000, page 26 et suivantes
- [2] **OneWireSpy**,
moniteur pour puces 1-Wire,
Elektor n°264, juin 2000,
page 22 et suivantes

Un clic sur le bouton « Viewer » ouvre une fenêtre dans laquelle vous trouverez quatre valeurs mesurées et divers réglages pour cette puce (voyez la **figure 4**). Vous pouvez librement modifier ces réglages : l'intervalle entre mesures, par exemple, débrancher l'un ou l'autre canal, changer le domaine de mesure d'un canal et ainsi de suite.

On peut aussi ouvrir plusieurs fenêtres en même temps, pour observer à l'écran plusieurs composants simultanément, se faire une idée générale du réseau et expérimenter, entre autres, sur la manière de faire travailler ensemble divers composants du bus à deux fils. C'est ainsi que vous pourrez par exemple connecter un potentiomètre « 1-Wire » sur un canal d'un convertisseur A/N et voir évoluer sa résistance.

Il est bien intéressant de bricoler avec ce système et sans doute de réaliser des applications simples, mais avec la visionneuse iButton, pas question de coupler les composants au niveau logiciel, sur le bus. Si par exemple vous souhaitez commander automatiquement un interrupteur par le bus lorsque la valeur mesurée par le canal A du convertisseur A/N dépasse une certaine limite, il vous faudra programmer vous-même. C'est le but du logiciel suivant que l'on trouve sur le site de Dallas : le « *1-Wire Software Developers' Kit* ».

1-Wire SDK

Vous trouverez la commande de téléchargement de cet ensemble sur la même page

Liens Internet :

1-Wire SDK:

<http://www.ibutton.com/software/tmex/>

1-Wire application notes:

http://dbserv.maxim-ic.com/an_prodline2.cfm?prodline=21

Dallas Semiconductor :

www.dalsemi.com

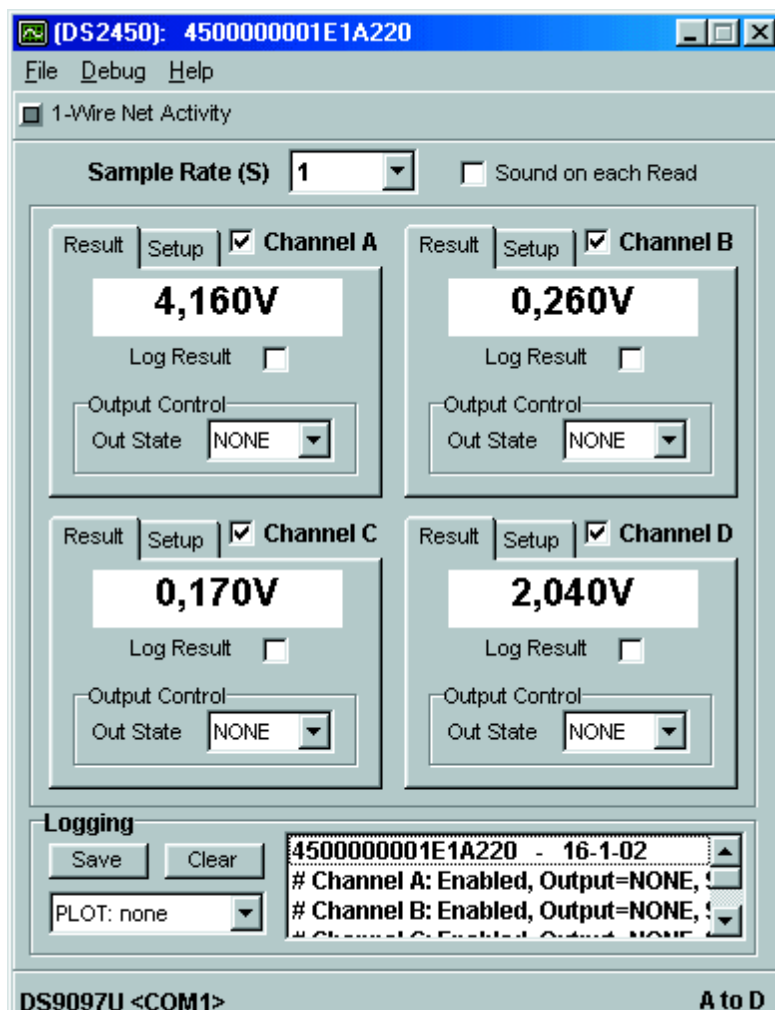


Figure 4. Cette fenêtre-ci montre les quatre résultats de mesure et les divers réglages du convertisseur A/N à quatre canaux DS2450.

Internet que la visionneuse iButton. La version actuelle est SDK V4.0 ALPHA. Elle contient entre autres la documentation de TMEX API (*application program interface*), la description des fonctions que l'on peut appeler à partir de ses propres applications pour communiquer avec le bus à deux fils et les composants qui y sont branchés. Elle propose également les pilotes pour interfaces, série et USB, d'un PC et des exemples d'applications en Pascal, Delphi, C et Visual Basic. Plus qu'il n'en faut pour se lancer soi-même dans la programmation.

Le bus à 2 fils à l'usage

Il n'est malheureusement pas possible de relier, sans autre forme de procès, un nombre illimité de composants « 1-Wire » à l'interface

décrite ici. Il existe, en pratique, des limites, tant physiques que topologiques, à l'extension du réseau. Aussi longtemps que l'interface et sa cour d'intervenants restent sur la table du laboratoire, la longueur du bus n'atteindra guère plus d'un mètre, pas de souci à se faire. Mais avant d'entreprendre, tête baissée, le forage des murs et l'installation de câbles d'un bout à l'autre de la maison, il y a peut-être intérêt à parcourir la note d'application AN148 « *Guidelines for Reliable 1-Wire Networks* ». Elle explique clairement, en long et en large, les mesures à prendre pour garantir une communication fiable en réseau. Cette notice contient encore d'autres informations sur l'emploi du DS2480B. Vous la trouverez parmi les autres notes d'application « 1-Wire ».

(020022)

Compilateurs pour 8051

Versions gratuites, shareware et de démonstration

Le type de microcontrôleur le plus utilisé dans le monde industriel est et reste le 8051 (et tous ses successeurs qui lui sont compatibles). L'un des avantages majeurs de l'utilisation de ce type de contrôleur tient au fait qu'il existe aujourd'hui une quantité impressionnante d'informations tant au niveau matériel qu'au niveau logiciel. Internet constitue dans ce cadre, pour les utilisateurs de composants dérivés du 8051, une source quasi-inépuisable. Nous nous limiterons ici spécifiquement aux compilateurs 8051.

Les bons vieux 8051 et 8052 doivent sans doute une bonne partie de leur popularité au fait qu'ils se laissent aisément programmer à l'aide d'un interpréteur BASIC, devenu fort populaire lui aussi, le BASIC-52.

De nos jours, plusieurs fabricants de semi-conducteurs proposent des microcontrôleurs au noyau basé sur le 8051, mais fabriqué selon une technologie plus moderne (et partant plus rapide) et qui possèdent en outre de nombreuses adjonctions par rapport à l'original telles que, par exemple, capacité de parasites mémoire plus importante, chien de garde, interface LCD, convertisseurs N/A et A/N, et toutes sortes d'autres interfaces telles que le bus CAN.

Même le 89S8252 utilisé sur notre carte 89S8252 Flash d'expérimentation sur laquelle repose notre cours « Microcontrôleurs » est lui aussi un « descendant » du 8051.

Pour nombre d'applications, BASIC-52 est trop lent de sorte qu'il est préférable d'utiliser l'assembleur. Ce n'est sans doute pas l'approche la plus facile. Dans ces conditions, on peut fort bien imaginer une solution intermédiaire sous la forme d'un compilateur; celui-ci convertit un certain nombre d'instructions « compréhensibles » en un programme en langage machine que l'on pourra ensuite transférer dans le contrôleur.

Il existe un nombre important de compilateurs 8051, mais l'utilisateur devra, pour la majorité d'entre eux, payer plus de 100 €, si



ce n'est encore bien plus. Cela représente une somme importante pour tout un chacun qui ne veut programmer que très rarement ou qui désireait prendre un compilateur en main. Il existe heureusement différents compilateurs gratuits et des versions de démonstration limitant la taille du code de programme générable. Ces

derniers sont bien souvent plus qu'adéquats pour la création de programmes de petite taille. Combien de fois n'avons nous pas été surpris de découvrir ce dont était capable un programme en langage machine de 1 ou 2 Koctets.

SDCC [1] est l'un des compilateurs C gratuits pour la famille 8051 les plus

connus. SDCC est l'abréviation de *Small Device C Compiler*, encore que certaines sources sur Internet traduisent ce quarteron de caractères par *Sandeep Dutta's C Compiler* (l'auteur ayant développé le compilateur). SDCC est un compilateur C ANSI optimisé pour les processeurs de type 8051 et Z80. On travaille pour le moment au développement de nouvelles versions pour d'autres types de processeurs populaires. SDCC existe en versions tournant sous Windows et sous Linux. Un certain nombre de programmeurs continuent d'assurer le développement de ce programme.

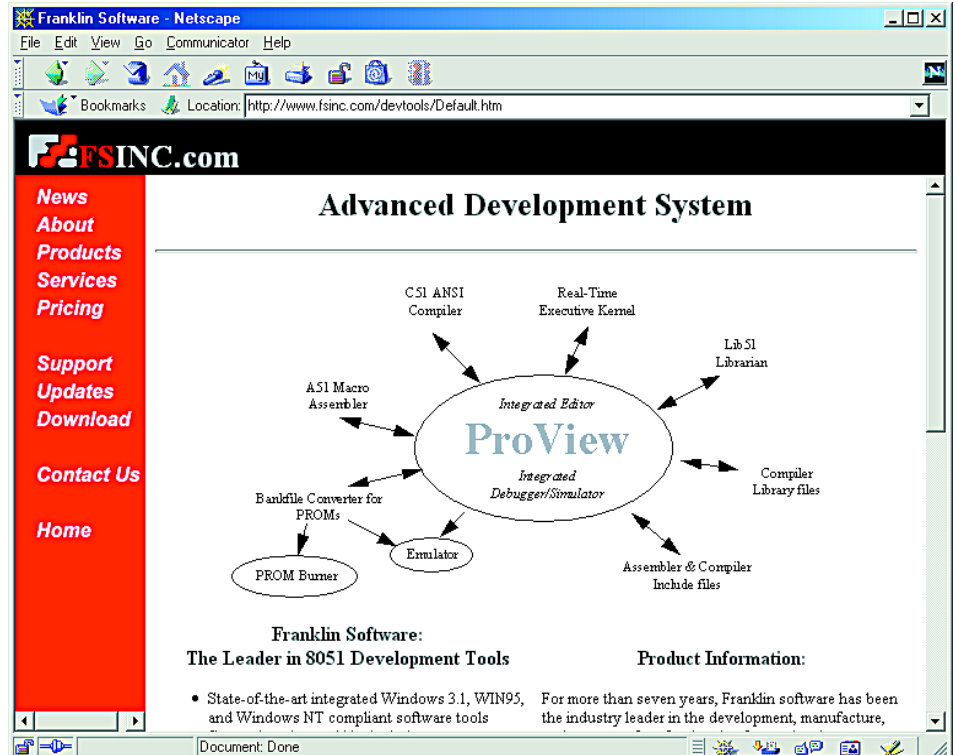
Dans l'article, nous avons mentionné un autre compilateur gratuit, à savoir *Reads51* de la société Rigel [2]. En principe, ce compilateur est destiné aux utilisateurs des cartes à microcontrôleurs proposés par Rigel, mais les lecteurs d'Elektor qui désirent suivre le cours « Microcontrôleurs » sont autorisés à le télécharger pour leur propre usage. Reads51 est une combinaison tous terrains associant un compilateur 8051 compatible small-C, un assembleur relatif, un éditeur de lien/locateur, un éditeur, un simulateur, un débogueur et un moniteur. Il s'agit d'un très bon set de programmes que l'on a vite fait de prendre en main.

Ceux d'entre nos lecteurs qui préfèrent Pascal au C peuvent s'essayer à *Pascal51*, l'un des compilateurs Pascal pour le 8051 les plus populaires.

Nous avons eu l'occasion de parler de ce programme à plusieurs reprises déjà. Les sites de téléchargement de ce programme (écrit au départ par un programmeur russe) ne cessent de changer, mais nous vous en proposons 2 où nous savons qu'il est encore disponible [3].

Derek Kennedy a conçu un certain nombre d'extensions que l'on retrouvera à la seconde page de la référence [3] (sous la dénomination de Prepro).

Embedded Pascal pour le 805x [4] est un compilateur Pascal avec un système de définition de composant (*device definition system*) qui permet d'adapter ce compilateur à pratiquement n'importe quel dérivé du 8051 actuellement sur le marché. Il s'agit d'un programme proposé en shareware permettant de créer un code-objet de taille limitée seulement...



Puisque nous en sommes à parler des versions shareware et de démonstration, revenons aux compilateurs C. La version d'évaluation des outils C51 (*C51-tools*) de Keil [5] comporte un compilateur 8051 parfaitement utilisable. La taille de code que peut générer le dit compilateur est de 1 Koctet. Il est malheureusement impossible, avec la version de démonstration de ce programme, de faire démarrer à 0000_{HEX} l'exécution du programme, ce qui rend ce programme inutilisable pour les microcontrôleurs mono-puce (*single-chip*).

Franklin Software propose un environnement de programmation pour le 8051 complet, le programme *Proview* (éditeur/débogueur/simulateur) en constituant le noyau. Cette société propose également un compilateur C51 ANSI (version 6). Il existe, pour l'ensemble, un set d'évaluation (d'une taille de 12 Moctets) à l'adresse [6] limité uniquement par la taille du code objet (4 Koctets maximum).

Raisonance [7], une société française propose plusieurs outils de programmation pour différents types de microcontrôleurs tels que le 8051, le XA de Philips et le ST6 de STMicroelectronics. Il existe une version de démonstration pour le 8051, ayant

une limitation de taille de programme de 4 Koctets au maximum.

Comme vous pouvez le constater, si l'on veut en savoir plus au sujet des compilateurs 8051, on a le choix entre plusieurs d'entre eux, certains gratuits, d'autres en version d'évaluation. À vos marques !

(025019)

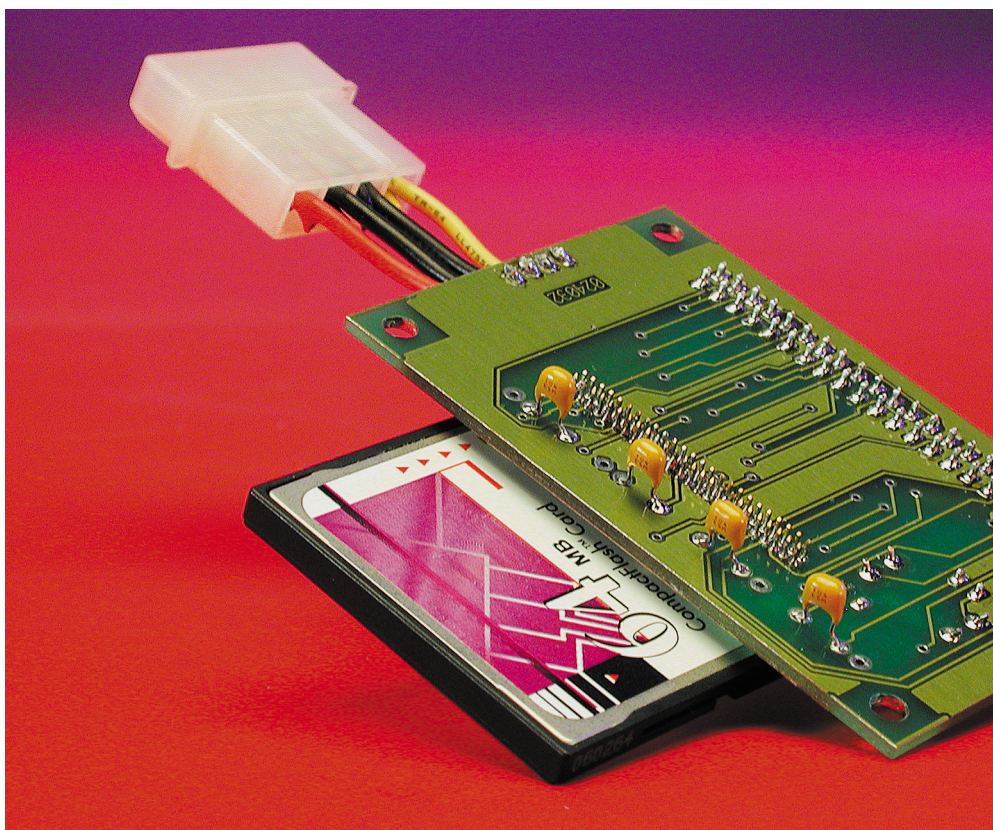
Adresses Internet :

- [1] SDCC:
<http://sdcc.sourceforge.net>
- [2] READS51:
www.rigelcorp.com/reads51.htm
- [3] Pascal51:
www.eedevl.com/files.html
Pascal51, Prepro51:
www.geocities.com/SiliconValley/Campus/9592/
- [4] Embedded Pascal:
<http://users.iafrica.com/r/ra/rainier/p8052.htm>
- [5] Keil C51 evaluation tools:
www.keil.com/demo/
- [6] Franklin Development Tools:
www.fsinc.com/devtools/Default.htm
- [7] Raisonance 8051-compiler:
www.raisonance.com/download/index.php

« Disque » Compact-Flash

De la mémoire « solid-state » pour votre PC

Les cartes Compact-Flash sont des supports de mémoire qui conservent les données qui y ont été stockées même en l'absence de tension d'alimentation. Le domaine où ils sont le plus utilisés est celui des appareils photo numériques. Grâce à « l'intelligence » présente dans une carte de ce genre il est également possible d'en envisager la connexion à un PC pour s'en servir, par exemple, en tant que « disque à semi-conducteurs ». La carte d'adaptation étonnamment simple décrite ici permet de coupler n'importe quelle carte Compact-Flash à votre ordinateur.



La carte Compact-Flash est une carte de mémoire développée à l'origine par la société Sandisk, carte dont les dimensions ne dépassent pas quelque 4 cm sur 4. La mise en oeuvre de cellules de mémoire non volatiles permet de conserver le contenu de la mémoire des années durant sans qu'il ne soit nécessaire de prévoir quelque alimentation que ce soit. Outre les appareils photo numériques évoqués plus haut, le monde des caméscopes est également fort demandeur de ce type de cartes de mémoire.

La capacité de mémoire atteint aujourd'hui jusqu'à 1 Goctet. Notons au passage qu'IBM propose même des mini-disques durs au format Compact flash ayant eux aussi une capacité pouvant atteindre 1 Goctet. On voit en **figure 1** les dimensions

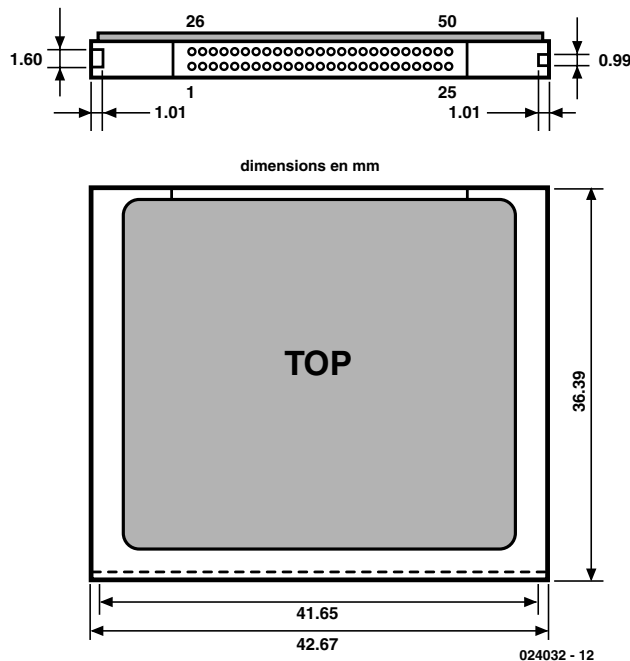


Figure 1. Dimensions et brochage d'une carte de mémoire Compact-Flash.

Figure 1. Brochage

Fonction	Broche	Fonction
GND	1	-CD1
D03	2	D11
D04	3	D12
D05	4	D13
D06	5	D14
D07	6	D15
-CE1/-CE1/-CS0	7	-CE2/-CE2/-CS1
A10	8	-VS1
-OE	9	-IORD
A09	10	-IOWR
A08	11	-WE
A07	12	-RDY,-BSY/-IREQ/INTRQ
VCC	13	VCC
A06	14	-CSEL
A05	15	-VS2
A04	16	RESET/RESET/-RESET
A03	17	-WAIT/-WAIT/-IORDY
A02	18	-INPACK
A01	19	-REG
A00	20	BVD2/-SPKR/DSAP
D00	21	BVD1/-STSCHG/-PDIAG
D01	22	D08
D02	23	D09
WP/-IOIS16/-IOS16	24	D10
-CD2	25	GND

Notes :

Le caractère – indique une inversion du signal

En ce qui concerne les broches à fonction triple (aa/bb/cc) :

- aa : en mode mémoire
- bb : en mode d'E/S (I/O)
- cc : en mode « true-IDE »

et le brochage de la carte Compact-Flash (on retrouve de plus en plus souvent la dénomination raccourcie de carte CF).

Les normes de cette carte définissent 2 épaisseurs : le type I se doit de posséder une épaisseur de 3,3 mm, le type II, dit de double épaisseur, aura lui une épaisseur de 5 mm. La grande majorité des cartes de mémoire à semi-conducteurs (*solid state*) sont de type I, les disques durs CF étant eux du type II.

Depuis un an, les capacités étant devenues de plus en plus importantes, les prix des cartes de mémoire de quelque type qu'elles soient, a fortement chuté; il est possible aujourd'hui, à la date d'écriture de ces lignes, mi-février, de trouver une carte CF de 64 octets de capacité pour moins de 50 € et de 128 Koctets pour moins de 80 €.

Si on passait une carte CF aux rayons X on verrait qu'elle est constituée d'un certain nombre de modules gérés par leur propre contrôleur. Le transfert des données, dans les 2 sens, se fait en parallèle, comme cela est le cas avec un disque dur. De par ce concept « intelligent » il est extrêmement facile de faire travailler une telle carte avec des systèmes à microcontrôleur, voire avec un PC.

Il existe, pour l'utilisation de ce type de cartes avec des ordinateurs portables des adaptateurs dits adaptateurs PCMCIA qui ne sont en fait rien de plus qu'une sorte de prolongateur du connecteur présent dans le portable.

On trouve sur le marché, en ce qui concerne les PC, toutes sortes de lecteurs de cartes de mémoire pouvant se brancher sur un port USB, parallèle voire sériel.

Il devient de plus en plus intéressant, en raison de leur prix de plus en plus abordable et de la caractéristique de « non volatilité » qui constitue l'un des points forts de la carte CF, d'envisager l'utilisation d'une telle carte en tant que support de mémoire additionnel pour le PC où il pourrait se substituer au lecteur de disquette. Pour cela, il n'est pas même nécessaire de faire l'acquisition d'un lecteur spécifique évoqué plus haut vu qu'il est possible de faire bien plus simple, comme le prouve le montage décrit dans le présent article.

Une électronique « passive »

L'intelligence intégrée sur la carte permet de la faire travailler en 3 modes différents : l'un de ces modes a été baptisé, ce qui tombe à pic pour la présente application, mode « *true-IDE* », obtenu par le forçage de la broche 9 à la masse. Dans ce cas-là la carte se comporte très exactement comme le ferait un disque dur doté d'une interface IDE. Tout ce dont nous avons encore besoin en fait est une

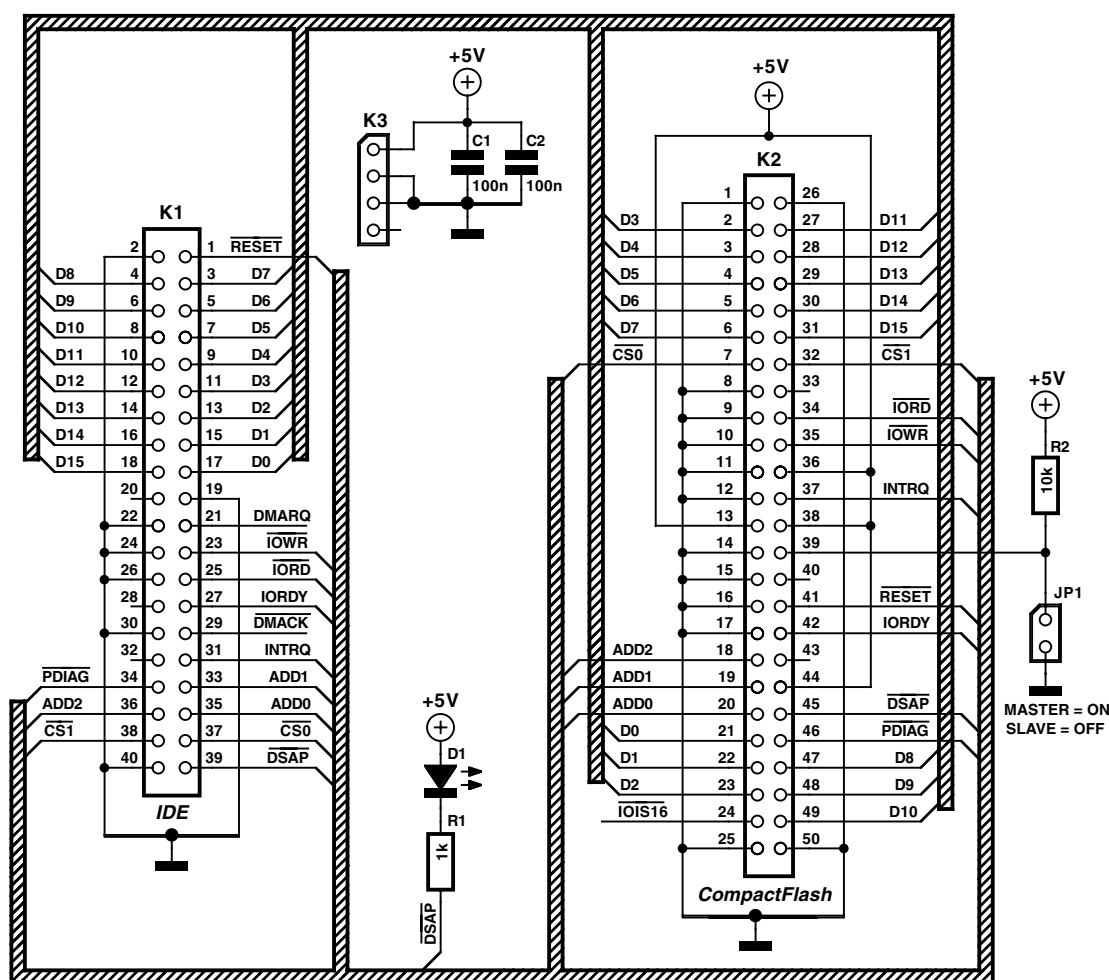
sorte d'adaptateur doté d'une paire de connecteurs, l'un pour recevoir la carte CF, l'autre pour la connexion au bus IDE du PC. La **figure 2** donne le schéma de l'électronique de cette réalisation. Elle se résume en fait à une paire de connecteurs épaulés par quelques rares composants connexes passifs. K1 est le connecteur allant vers le bus IDE, K2 assurant la connexion vers la carte CF. Il ne nous faudra guère plus de quelques lignes pour décrire le reste des composants. La LED D1 visualise les opérations de lecture et d'écriture de la carte CF, la résistance R1 limitant le courant circulant par la dite LED. La broche 39 du connecteur pour la carte CF, K2, est reliée au +5 V par le biais de la résistance R2. De ce fait, la carte CF travaille en « master » au niveau du bus IDE. Si, à l'inverse, nous forçons cette même broche 39 à la masse par la mise en place du cavalier JP1, la carte CF travaillera en esclave (slave) sur le bus. Les condensateurs C1 et C2 assurent le découplage de la tension d'alimentation.

On trouve, de plus, une troisième embase, K3, embase qui recevra le connecteur d'alimentation. Bien que la carte CF n'ait besoin que du +5 V et de la masse, nous avons choisi d'utiliser un connecteur PC quadri-polaire pour rester compatible avec les normes des connecteurs d'alimentation du PC. La ligne 12 V de ce connecteur n'est pas, répétons-le, utilisée.

La réalisation

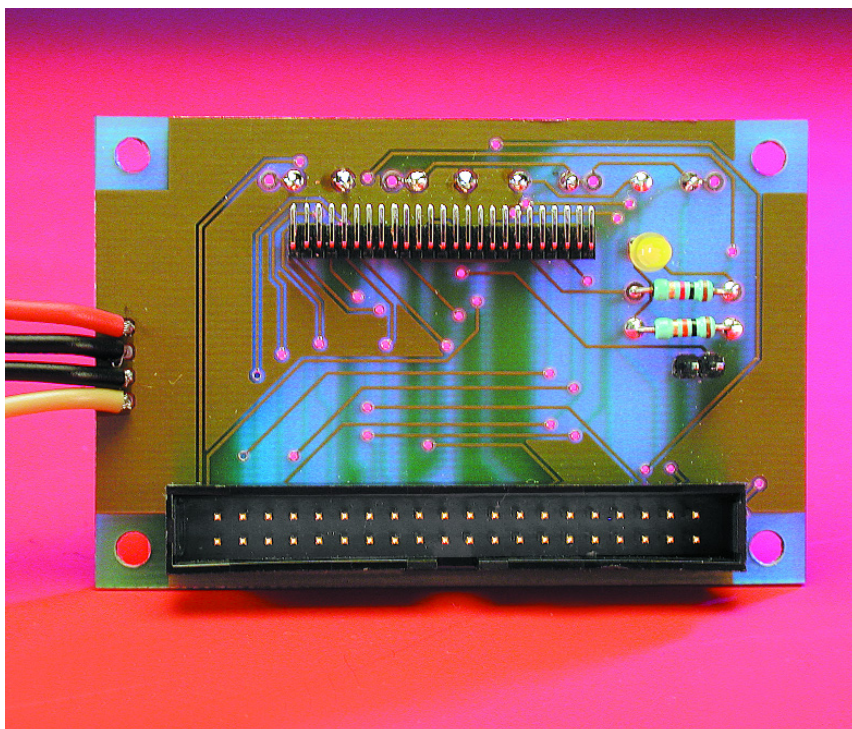
Nous avons dessiné un circuit imprimé double face à trous métallisés à l'intention de ce montage, platine dont on retrouve le dessin des pistes et la sérigraphie de l'implantation des composants en **figure 3**. Bien que nous n'ayons ici en fait affaire qu'à une paire de connecteurs associés à quelques composants

seulement, le connecteur à 50 contacts destiné à recevoir la carte Compact-Flash pose quelques problèmes. En effet, si l'écartement des broches d'un connecteur standard est au pas de 1/10 de pouce (2,54 mm), celui de ce type de connecteur est au 1/20ème de pouce, un peu plus de 1 mm... Il est partant pratiquement impossible d'interconnecter les deux connecteurs en câblage libre. Même si l'on dispose du circuit imprimé proposé ici, il faudra impérativement utiliser un fer à souder à pointe très fine et travailler avec soin et précision. Si ce montage vous a enthousiasmé et que vous envisagez de le réaliser, nous nous permettons de vous donner un conseil : commencez par vous assurer que vous pouvez bien mettre la main sur le connecteur CF. Il est disponible, entre autres sources, chez



024032 - 11

Figure 2. Le schéma de notre adaptateur : rien que des composants passifs.



Farnell, mais la plupart des revendeurs de composants n'auront pas ce type de produit à leur catalogue. Une fois que vous aurez ce connecteur en votre possession vous pourrez acheter la platine et le reste des composants, opération qui ne devrait pas poser de problème.

Les connecteurs, les résistances et la LED prennent place côté « composants » de la platine. Les 2 condensateurs de découplage viennent se monter sur le côté « pistes »; il faudra couper les queues de ces 2 composants au ras de la platine pour éviter qu'ils n'entrent en contact avec la carte CF.

Nous utilisons, en ce qui concerne le connecteur d'alimentation, un prolongateur ou un adaptateur en Y pour alimentation de PC (disponible dans tous les catalogues et dans n'importe quel magasin vendant des pièces détachées pour PC). Nous en coupons la fiche en laissant une certaine longueur à ses câbles. Les conducteurs allant vers la fiche (mâle) sont soudés aux points correspondants de K3. Le fil rouge est le fil du +5 V, les 2 fils de couleur noire sont ceux de la masse, le fil jaune étant celui du +12 V.

Ceci fait, vous pourrez, le cas échéant, implanter le cavalier JP1 si vous voulez que le PC reconnaisse la carte CF comme lecteur esclave (le connecteur

Liste des composants

Résistances :

R1 = 1 k Ω
R2 = 10 k Ω

Condensateurs :

C1, C2 = 100 nF

Semi-conducteurs :

D1 = LED jaune faible courant

Divers :

K1 = embase à 2 rangées de 20 contacts (HE-10)
K2 = embase en équerre à 2 rangées de 25 contacts au pas de 0,05 " (Farnell n° commande 3078127)
JP1 = embase autosécable à 2 contacts + cavalier de court-circuit
prolongateur miniature pour câble d'alimentation de PC

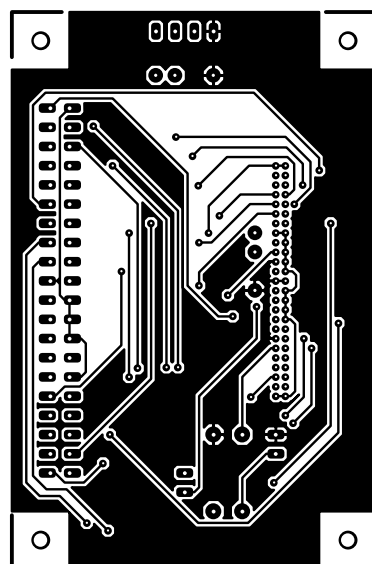
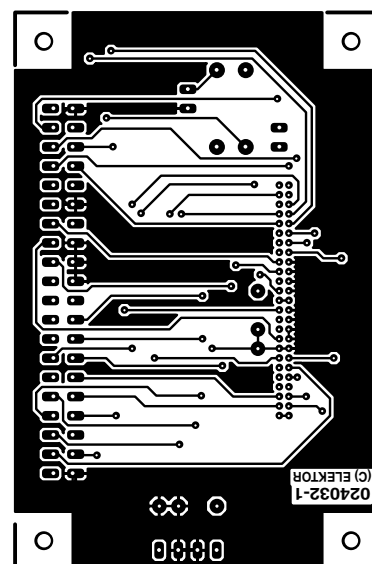
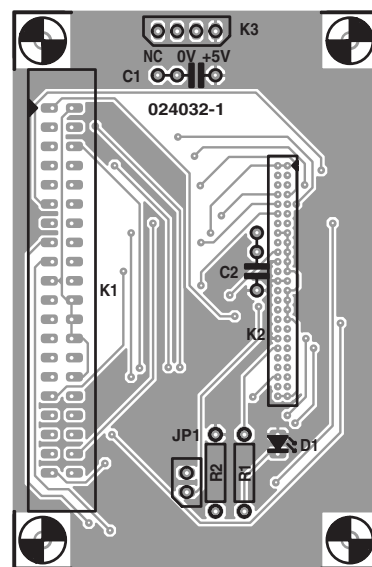


Figure 3. Ce minuscule circuit imprimé double face à trous métallisés simplifie très notablement la réalisation de ce projet, ne serait-ce qu'en ce qui concerne le montage de K2.

IDE correspondant comporte déjà un disque dur ou un lecteur de CD-ROM monté en maître). En l'absence de cavalier la carte est automatiquement montée en lecteur maître. On pourra ensuite monter la carte de l'adaptateur dans l'ordinateur et la connecter au système (par le biais du câble IDE correspondant). On pourra, par exemple, doter une plaquette de fermeture pour lecteur 5"1/4 présente d'une fente, plaquette sur laquelle la platine de l'adaptateur sera fixée, « **tête en bas** », à l'aide d'une paire d'équerres de fixation. K2 devra se trouver, de par la faible longueur de la carte CF, relativement près de la fente d'introduction de la carte. On relie, comme nous le disions plus haut, le circuit imprimé de l'adaptateur à l'un des bus IDE de l'ordinateur à l'aide de son embase K1 qui reçoit un connecteur de câble IDE standard. La contre-fiche soudée aux points embase K3 sera reliée à l'un des connecteurs d'alimentation libres du PC (si ce dernier n'en possède plus, il vous faudra acheter un câble distributeur en Y).

L'ensemble est maintenant prêt à être utilisé. Nous vous recommandons, pour éviter toute mauvaise surprise, de n'implanter ou de ne sortir la carte CF qu'ordinateur coupé.

Lorsque vous aurez implanté une carte CF pour la première fois dans le connecteur et démarré ensuite l'ordinateur, Windows vous annoncera qu'il faut installer un pilote (driver) pour ce nouvel « appareil ». En règle générale ce pilote est déjà présent et le reste du processus se déroule automatiquement, mais il peut arriver qu'il soit nécessaire de remettre le CD-ROM d'installation de Windows dans le lecteur de CD.

À partir de là, Windows reconnaîtra automatiquement la carte CF comme une sorte de disque dur que vous pourrez lire et écrire comme vous le feriez avec un disque dur classique.

Permettez-nous de terminer cet article par un avertissement :

Veillez à toujours implanter la carte CF dans le bon sens dans l'embase montée sur l'adaptateur. Cela signifie que **le dessus de la carte CF doit être orienté vers le circuit imprimé**. C'est d'ailleurs là la raison pour laquelle nous avons mentionné plus haut qu'il fallait monter la platine dans l'ordinateur composants sur le dessous. *La photo en début d'article illustre clairement le but de la manoeuvre*. Vous pourrez, si vous voulez éviter toute erreur de mise en place, doter la fente d'introduction d'un détrompeur. Comme le montre la figure 1, les fentes de guidage de la carte sont de largeurs différentes sur les chacun des côtés de la carte. Pourquoi ne pas mettre cette spécificité à profit ?

(024032)

B.P. 11 568

Nous ne pouvons malheureusement pas répondre in extenso à toutes les lettres relevant des questions techniques. Dans cette rubrique nous répondons à des lettres pouvant présenter un intérêt général et concernant des montages âgés de moins de 2 ans. Vu le nombre de lettres qui nous arrivent mensuellement, nous regrettons de ne pas pouvoir répondre séparément à chacune d'entre elles et sommes dans l'impossibilité de donner suite à des souhaits individualisés d'adaptation de montages publiés ou de réalisation de montages à publier ni même de répondre à des demandes d'information additionnelle concernant un montage décrit dans Elektor.

PICÉE, système de développement (I)

Bonjour, une petite erreur s'est glissée dans l'adresse du site de NTPicprog que ce soit dans la revue ou le site Internet.

La bonne adresse est :

<http://home.swipnet.se/~w-24528/NTPicprog/>

(il y a un "-" entre le "w" et le "2").

Par contre, je suis un peu interrogatif vis à vis du PIC16F84 qui est mentionné comme devant être programmé EPS010062-41. Or je n'ai pas trouvé cette référence au niveau de la rubrique platines & logiciels, et l'article ne fait pas référence à une programmation particulière du PIC, cordialement.

Hervé MASSON

Merci de votre correction.

J'en ferai usage dans un prochain courrier du lecteur d'autant plus qu'il soulève un problème qui a échappé à notre attention. En effet, il n'y a pas de PIC16F84 à programmer spécialement pour cette application.

PICÉE, système de développement (II)

Dans le numéro 284 d'elektor à la page 11 rubrique liste des composants ; divers : à quoi correspondent les deux étoiles.

Merci par avance.

Arnaud Laaniray

En règle générale, un astérisque dans la liste des composants (*) renvoie à un * cf texte en bas de liste, mais visiblement, cette ligne étant de trop lors de la mise en page elle a été supprimée.

Un * signifie partant que l'on parle de ce composant dans le texte.

Dans le cas de K4 cela signifie que l'on peut également envisager d'effectuer une connexion de l'afficheur par câblage direct au lieu d'un morceau de support

autosécable...

En ce qui concerne la valeur de X1, le quartz, sa valeur, comme le dit le texte, dépend de la fréquence de travail du PIC16F84 utilisé dans le montage, entre 4 et 20 MHz.

Nouveau logiciel pour le cours « Microcontrôleurs »

Un lecteur allemand, Ulrich Bangert (DF6JB), propose, à l'intention de la carte AT89S8252, un logiciel de programmation plus convivial et qui supporte les ports COM1 à COM8. Le programme lit et écrit les domaines de mémoire de programme et de mémoire de données du processeur. Ce programme comporte en outre un éditeur hexadécimal et un terminal RS-232. On pourra télécharger le programme **atmelisp.zip** (389 Koctets) depuis le site Internet d'Elektor à l'adresse :

www.elektor.fr

ainsi que sur le site de l'auteur du cours, Burkhard Kainka :

<http://B.Kainka.bei.t-online.de/basismi/ht>.

Burhard Kainka a placé sur sa page d'accueil sise à l'adresse <http://B.Kainka.bei.t-online.de>, un panorama (en allemand malheureusement pour nous les habitants de l'Hexagone) de ce qui a trait à la carte associée au cours et propose des liens et des fichiers d'aide intéressants.

Un autre lecteur, Harald Rilling, propose, sur sa page d'accueil (en allemand) située à l'adresse <http://home.t-online.de/home/Harald-Rilling>, une aide précieuse au sujet au BASIC-52 et le manuel original (en anglais) concernant ce composant d'Intel.

Ampli Hi-Fi P3 : tensions et courants

J'ai une question au sujet de l'amplificateur Hi-Fi P3 décrit

dans votre numéro de mai 2001. J'ai besoin de toutes les tensions et des valeurs de courant correspondantes (valeurs en V et en A) requises pour un fonctionnement optimal de cet amplificateur. Je n'ai malheureusement pas trouvé d'informations à ce sujet ni dans le magazine en question ni sur votre site Internet.

Martin Kunz

Dans le cas de la version mono de l'ampli Hi-Fi P3 à EL 34, le transformateur utilisé est un NTR-P/6 et pour la version mono à base de KT 88, c'est un transfo du type NTR-P/7. Les versions stéréo de cet amplificateur font respectivement appel à un NTR-P/3 (EL 34) et NTR-P/5 (KT 88). Les tensions et courants correspondants sont donnés sur la page correspondante du site Internet d'EXPERIENCE electronics sis à l'adresse :

www.experience-electronics.de

Gerhard Haas

Bande passante du GBDSO

J'ai une question à vous poser en ce qui concerne l'oscilloscope numérique pour Gameboy, le fameux GBDSO, décrit dans les numéros d'octobre et de décembre 2000 d'Elektor, en ce qui concerne la limitation de la bande passante en CC (DC) :

À quoi est en fait due cette limitation ? Comment pourrait-on faire (si tant est que cela soit possible, d'atteindre une bande passante plus importante, de 1 MHz par exemple ?

Richard Fritzen

La bande passante doit sa limite au théorème d'échantillonnage de Shannon qui fait que le taux d'échantillonnage est théoriquement égal à la moitié du taux d'échantillonnage. Dans le cas d'un oscilloscope numérique cette limite se situe, en pratique, approximativement au 1/10ème

du taux d'échantillonnage. | A un taux d'échantillonnage de 1 MHz on obtient tout juste 10 points de mesure par période du signal de 100 kHz mesuré. Il n'y a malheureusement rien à y faire, le convertisseur analogique/numérique et le GameBoy sont tout simplement trop lents pour un taux d'échantillonnage plus élevé.

Programme de dessin de circuit imprimé pour Mac

J'aimerais bien savoir s'il existe un bon programme de dessin de circuit imprimé et de saisie de schéma pour le Mac.

Holger Kueper

Nous ne connaissons malheureusement qu'un seul et unique programme de CAO tournant sur Macintosh, à savoir McCAD E.D.S., programme dont vous pouvez trouver un descriptif et une version de démonstration (6,5 Moctets) sur son site sis à l'adresse :

<http://www.mccad.com>.

Notons qu'il existe également une version tournant sous Windows de ce programme (version de démonstration de 5 Moctets). Nous ne doutons pas le moins du monde qu'il y en ait d'autres dont ne manqueront pas de nous parler nos lecteurs suite à cette réponse succincte.

Contrôleur de la carte 89S8252 Flash

Sur la carte 89S8252 Flash il est indiqué IC1 = AT89S8252 24PC. Mais il se veut que je possède encore un AT89C52 20PI. Puis-je l'utiliser comme alternative ?

Jokel

Malheureusement non, le AT89C52 20PI ne se laisse pas programmer « en circuit » (ICP).

Filtre-bouchon ajustable

Pour le domaine des BF

Harro Kühne

Il est possible, pour peu que l'on ajoute une paire d'amplificateurs opérationnels à un pont de Wien-Robinson, de réaliser, des filtres coupe-bande (dits bouchon) dont la fréquence de travail est ajustable à l'aide d'une seule et unique résistance. Comparée au réseau en double T souvent utilisé pour la réalisation de filtres-bouchon, cette approche présente un avantage indéniable.

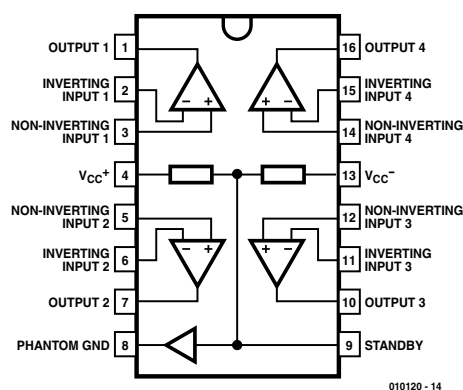


Figure 1. Cet amplificateur opérationnel rail à rail génère en interne un potentiel de masse tamponné.

Le filtre-bouchon décrit dans le présent article fait appel au TS925, un quadruple amplificateur opérationnel de STMicroelectronics. De par sa caractéristique rail à rail (*rail to rail*) il se laisse commander entre la masse et la tension U_S tant en entrée qu'en sortie et présente l'avantage additionnel de créer, par le biais d'un diviseur de tension interne et d'un tampon IC1.E, un potentiel de masse virtuel de $U_S/2$, nécessaire aux amplificateurs opérationnels (figure 1).

Le schéma de la figure 2 montre un pont de Wien-Robinson modifié ayant une fréquence de résonance f_o . En combinaison avec C2, C4, R3, R6, R7 et R8, IC1.B constitue la branche complexe du pont, R1+R2, R5 et IC1.C en constituant la branche

réelle. IC1. C travaille en amplificateur inverseur dont il est possible, par le biais de R1, entre $|v'_{IC1.C}| = 0,93$ à 1,07, en vue d'obtenir la meilleure atténuation de filtrage possible (nous avons mesuré, sur notre prototype, $u_{a_eff}/u_{e_eff} > -55$ dB) de

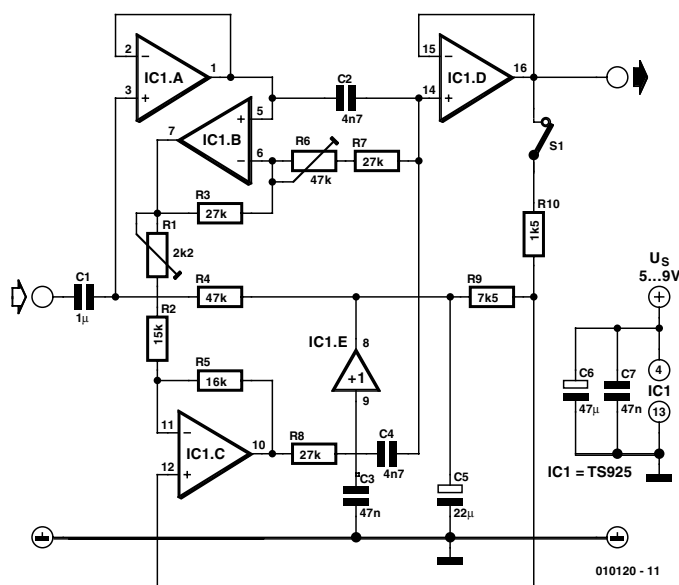


Figure 2. Filtre-bouchon facile à calibrer.

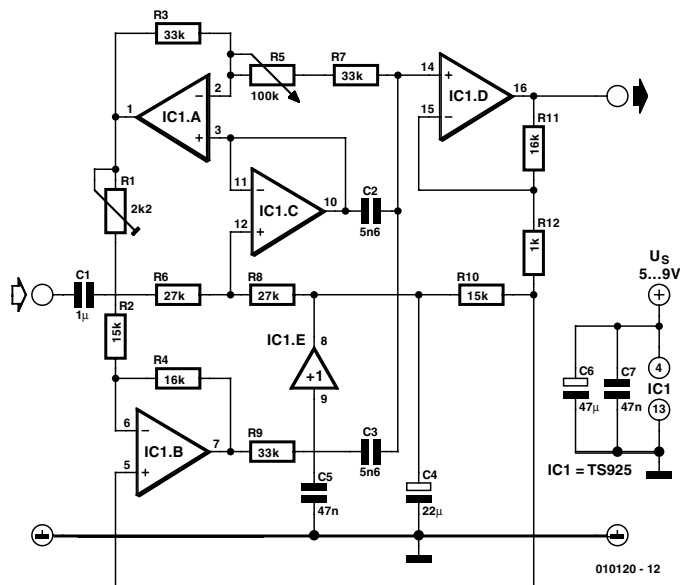


Figure 3. Filtre-bouchon pour tensions d'entrée élevées.

corriger à des tensions d'entrée de fréquence f_o . Pour des tensions d'entrée de fréquence $f_e < f_o$, C2 et C4 sont pratiquement sans effet de sorte que le circuit travaille en suiveur de tension avec $|v_{SF}| = 1$. Dans le cas inverse, où $f_e > f_o$, C2 et C4 sont à considérer comme des courts-circuits, de sorte que le circuit travaille à nouveau en fonction suiveur de tension désirée dans le domaine des fréquences passantes du filtre-bouchon. Avec le respect impératif du dimensionnement requis où $R3 = R8$, $C2 = C4$ et $R1 + R2 = R5$, la fréquence f_o du pont de Wien-Robinson dotée des amplificateurs opérationnels IC1.B et IC1.C répond à la formule suivante :

$$f_o = 1 / [2\pi \cdot R8 \cdot C4 \sqrt{(R6 + R7)/R3}]$$

En principe, le dimensionnement supérieur ou inférieur à 1 de $(R6 + R7)/R3$ n'a pas d'importance. $R6 + R7)/R3 \geq 1$ est, compte tenu de la tension d'entrée maximale du filtre, un repère plus convenable. En effet, indépendamment de la position du potentiomètre de réglage R6, les valeurs de crête des tensions de sortie de IC1.B et IC1.C atteignent alors, au maximum, le niveau $\pm \hat{u}_{IC1.B} = \pm \hat{u}_{IC1.C} \leq \pm 2\hat{u}_e$ ceci si l'on suppose que la tension d'entrée est de forme sinusoïdale. On pourra partant, admettre, avec les amplificateurs opérationnels rail à rail utilisés et le dimensionnement adopté, à savoir $(R6 + R7)/R3 \geq 1$, des tensions d'entrée sinusoïdales dont le niveau ne doit pas dépasser respectivement $\pm \hat{u}_e = U_S/4$ et $u_{e_eff} = U_S/4\sqrt{2}$ au maximum.

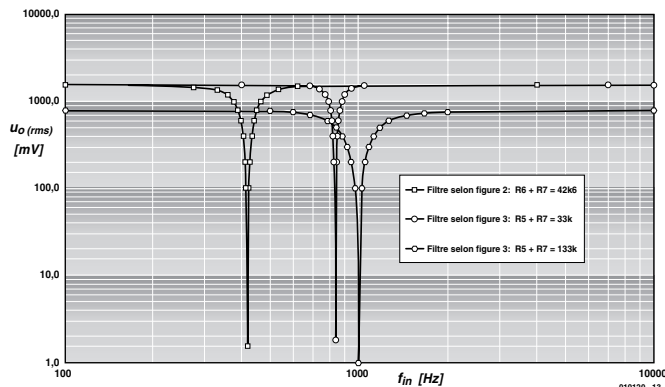


Figure 4. Courbe de réponse en fréquence du filtre-bouchon décrit ici.

Le rapport $(R6 + R7)/R3$ exerce également une influence sur le facteur de qualité, Q_{SF} du filtre-bouchon. Si l'interrupteur S1 est ouvert et que $R6 + R7 = R3$, le facteur Q du filtre atteint, avec un $Q_{SF} = 0,5$, une valeur que l'on peut également dériver du pont de Wien-Robinson mis en oeuvre ici. Le facteur Q croît en cas de réinjections d'une faible partie de la tension de sortie sur l'entrée non-inverseuse de IC1.C. Q_{SF} répond alors à la formule suivante :

$$Q_{SF} = (R9 + R10) / [2 \cdot R10 \sqrt{(R6 + R7)/R3}]$$

Si $R9 = 0$ ou que l'interrupteur S1 est ouvert, cette réinjection est bloquée. Le filtre atteint dans ce cas-là un facteur Q $Q_{SF} = 1/2 \cdot \sqrt{(R6 + R7)/R3}$. Inversement, si $R10 = 0$, le facteur Q du filtre tend vers l'infini : le circuit devient instable et entre en oscillations. Nous avons dimensionné notre prototype pour des tensions d'entrée sinusoïdales comprises entre $u_{e_eff} \leq 0,8 \text{ V}$ à $U_S = 5 \text{ V}$ et une plage de fréquences de $f_o = 0,76$ à $1,25 \text{ kHz}$.

Une alternative

L'alternative proposée en **figure 3** admet, avec un gain de filtre dans la plage des fréquences passantes de $v_{SF} = 1$, des tensions d'entrée sinusoïdales de $\pm \hat{u}_e = U_S/2$. Ceci est obtenu par le biais du diviseur de tension d'entrée $R6/R8$ dont l'atténuation compense le gain de $v'_{IC1.D} = 2$ induit par l'amplificateur IC1.D. Si $C2 = C3$, $R3 = R9$ et $R1 + R2 = R4$, la fréquence et le facteur Q de ce filtre-bouchon répondent aux formules suivantes :

$$f_o = 1 / [2\pi \cdot R9 \cdot C3 \sqrt{(R5 + R7)/R3}]$$

$$Q_{SF} = (R10 + R12) / [2 \cdot R12 \sqrt{(R5 + R7)/R3}]$$

Nous avons effectué des mesures sur ce circuit avec des tensions d'entrées sinusoïdales de $u_{e_eff} \leq U_S/\sqrt{8}$, $U_S = 5$ à 9 V et pour la plage de fréquences $f_o = 420$ à 834 Hz . En concordance avec la dernière équation ci-dessus, le facteur Q en fonctionnement du filtre varie avec la taille de la résistance de réglage R5, comme l'illustre la courbe de réponse de transfert du filtre-bouchon représenté en **figure 4**. Il est intéressant de noter que la bande passante $B_{SF} = f_o/Q_{SF}$ du filtre reste, à sa syntonisation, constante :

$$B_{SF} = R12 / [\pi \cdot R9 \cdot C3 \cdot (R10 + R12)]$$

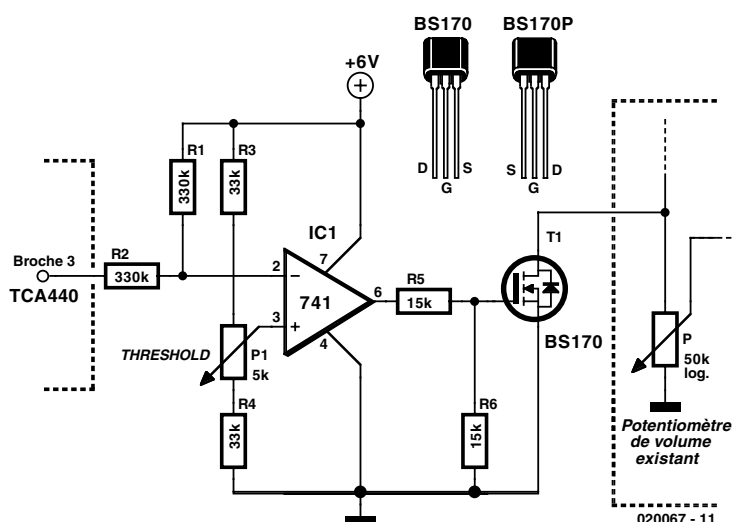
Pour $f_o = 420 \text{ Hz}$ et $f_o = 834 \text{ Hz}$ nous avons relevé, à chaque fois, des bandes passantes de $B_{SF} = 110 \text{ Hz}$, alors que les calculs donnent $B_{SF} = 108 \text{ Hz}$. Pour une fois, les valeurs de théoriques et pratiques « collent » pas mal.

Module de silencieux

pour le récepteur « aviation »

Projet : G. Baars

Le magazine de mars vient à peine de tomber dans les boîtes aux lettres et d'arriver sur les étagères, que les premières réactions de lecteurs concernant le Récepteur pour la bande « aviation » nous arrivent.



Si l'intérêt semble être suscité, certains n'ont pas manqué de signaler une absence qui leur semble préjudiciable au fonctionnement confortable de ce récepteur, celle d'un dispositif de suppression de bruit en l'absence de signal. Comme nous nous attendions à cette réaction, voici l'électronique additionnelle qu'il vous faut.

Silence, please!

Un silencieux (*squelch*) est un dispositif utilisé pour éviter que le radio-amateur ne soit confronté, en usage courant du récepteur, à des signaux de niveau faible et partant inintelligibles et aux bruits de fond (produits par le démodulateur, le récepteur HF et les amplificateurs de la fréquence intermédiaire, IF). Ce dispositif prend

la forme d'un seuil ajustable en deçà duquel les signaux reçus sont purement et simplement éliminés et partant inaudibles. Seuls les signaux de niveau supérieur au seuil de silencieux ainsi défini, arrivent à la circuiterie audio du récepteur, ce qui a pour résultat que seuls sont reçus des signaux utiles satisfaisants et que l'on ne se trouve pas confronté à des sifflements ennuyeux en l'absence de signal en entrée.

Le terme anglais *squelch* a une double signification, tout d'abord rendre silencieux, acception que nous avons reprise dans la traduction de silencieux, et d'autre part faire un bruit gras de succion tel que le produirait un déplacement dans de la boue épaisse. On comprendra mieux la seconde acception lorsque l'on aura entendu ce que l'on appelle

le *squelch trailer*, qui est une courte salve du bruit produit par le récepteur juste après que le circuit de silencieux se soit désactivé. Cette queue (*trailer*) est plus sensible en modulation de fréquence (FM) qu'en modulation d'amplitude (AM) en raison du niveau de bruit de sortie plus élevé produit par le démodulateur dans le premier mode.

Comment cela fonctionne-t-il ?

L'électronique présentée ici est une extension optionnelle destinée au récepteur pour la bande « aviation ». Tous les signaux et ses tensions d'alimentation sont dérivées du récepteur lui-même.

L'amplificateur opérationnel IC1 est monté en comparateur dont la sortie reste au niveau haut tant que le niveau du dispositif de commande automatique de gain (CAG = AGC pour *Automatic Gain Control*) relevé en broche 3 du TCA440 (du récepteur) se situe en deçà d'un certain niveau défini par le biais du potentiomètre P1. Tant que cela est le cas, le transistor FET T1 court-circuite le signal à la masse de sorte que le « silence règne sur la fréquence ».

Dès que le niveau de CAG dépasse le seuil prédéfini, ce qui signale la réception d'un signal intelligible, la sortie du 741 bascule au niveau bas et le signal audio est autorisé à passer vers le potentiomètre de volume du récepteur.

La tension d'alimentation de +6 V requise par cette extension de silencieux est dérivée du récepteur. Il faudra utiliser du câble blindé pour établir la liaison requise entre le drain du BS170, notre FET, et la patte chaude (le haut, par opposition à la patte reliée à la masse) du potentiomètre de volume.

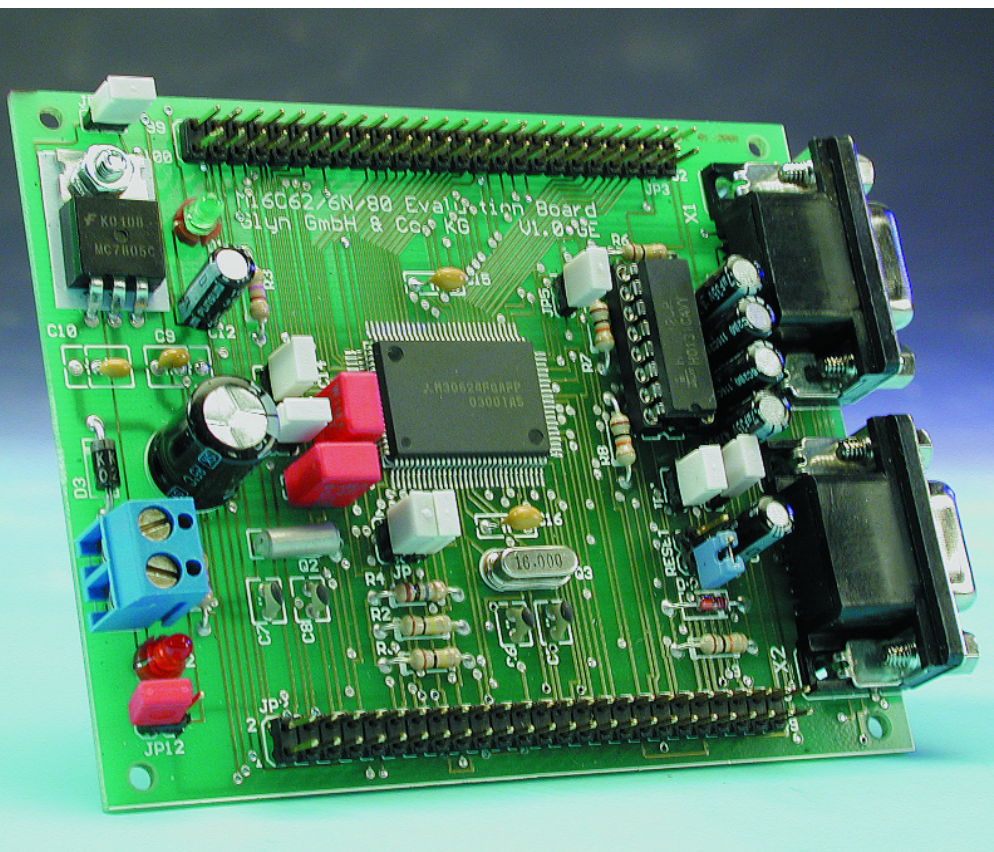
(020067-1)

MCI 6C/62 microcontrôleur 16 bits Flash (I)

Carte d'évaluation à prix « super »

Dipl. Ing. (FH) Gunther Ewald

Il existe, pour la famille des M16C de Mitsubishi, des microcontrôleurs très puissants et universels, depuis quelque temps déjà, une carte d'évaluation disponible auprès des distributeurs de ce fabricant pour très exactement 50 €. Et ce compilateur C, assembleur, moniteur et logiciel Flash compris !



Cela fait plus de 2 ans que Mitsubishi propose des microcontrôleurs Flash 16 bits à faible consommation de courant. Fabriqués en technologie 0,5 μm , ces microcontrôleurs dotés d'une mémoire Flash réinscriptible jusqu'à 100 000 fois dont la taille peut atteindre jusqu'à 256 Koctets, conquièrent une part de plus importante du marché. Il est possible, par le biais de ses 11 temporisateurs 16 bits universels, qui peuvent être cadencés indépendamment l'un de l'autre, d'effectuer des opérations classiques et spéciales telles qu'entrées incrémentales pour capteurs, sortie MLI (PWM), sorties MLI 3 phases de même que mesures d'impulsions, de périodes et de largeur d'impulsion.

Les temporisateurs acceptent un signal d'horloge externe d'une fréquence pouvant aller jusqu'à 10 MHz et sont cascadables, ce qui signifie qu'un temporisateur peut servir de cadenceur pour n'importe quel autre. Le convertisseur A/N 10 bits extrêmement rapide à 10 entrées peut

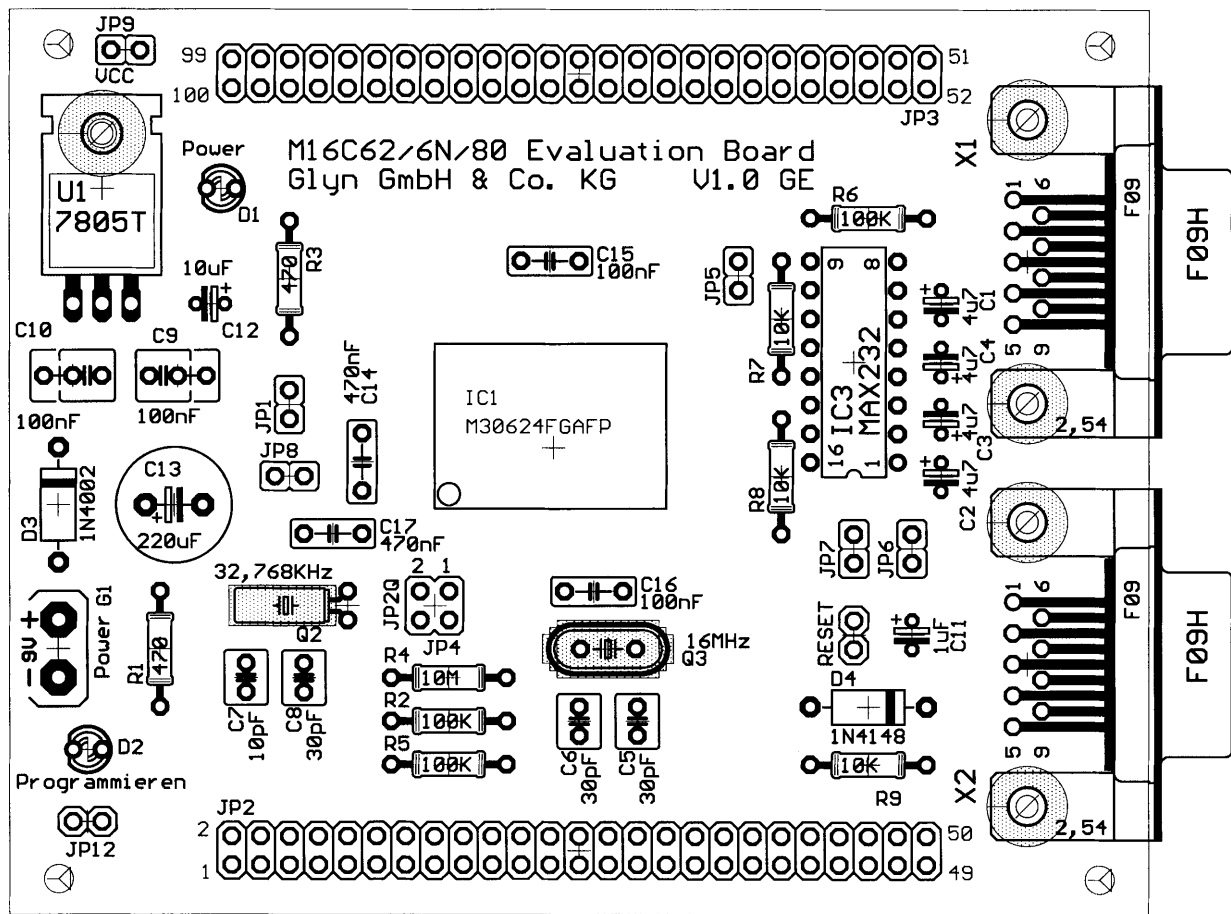


Figure 1. La platine de la carte de développement pour le M16C.

fournir, fonction d'échantillonnage/blocage (S&H pour *Sample and Hold*) au choix activée ou non, un résultat de mesure en 3,3 μ s (avec S&H) seulement.

L'échantillonnage connaît plusieurs modes, mono-coup (*One-shot*), répétition (*Repeat*), balayage simple (*Single-sweep*) et balayage multiple-0 et -1 (*Repeat-sweep-0* et -1), sachant qu'au cours de ce dernier mode on a non seulement répétition de la mesure sur chaque entrée, mais également « favoritisme » de l'entrée AN0 sachant qu'elle est interrogée toutes les 2 mesures (AN0, AN1, AN0, AN2, AN0, AN3 etc.).

Les 2 convertisseurs numérique/analogue (CNA) à 8 bits en technologie R-2R très rapides eux aussi peuvent fournir des tensions en 3 μ s seulement, étant partant même en mesure de reproduire des enregistrements de parole.

L'unité CRC remplit une fonction particulière; elle utilise la formule $X^8 + X^4 + X^3 + X^2 + X + 1$

pour, à raison de 2 cycles d'horloge par octet, vérifier, par exemple, la validité de données entrantes par le biais de l'interface série.

Le composant comporte pas moins de 5 interfaces sérieles, 3 d'entre elles pouvant travailler non seulement en mode synchrone mais également en mode asynchrone. L'UART2 peut même être utilisé en interface I²C (*multi-master*) ou en tant qu'interface de carte SIM. Les interfaces synchrones atteignent des taux de transmission de données grimper jusqu'à 8 Mbauds (horloge MCU/2). les interfaces asynchrones jusqu'à 1 Mbaud (horloge MCU/16). La technologie des ports est du HCMOS standard de sorte qu'ils en ont les caractéristiques de niveaux de tension. Il est possible, sur certains des ports, de programmer des résistances de forçage au niveau haut (*pull up*).

Le M16C/62 dispose de 2 canaux DMA permettant, indépendamment l'un de l'autre, de transférer des données d'une adresse vers un champ

de données et inversement. Il est possible, sachant que DMA1 suit DMA0, en s'aidant d'une adresse de stockage intermédiaire, de déplacer en mémoire des blocs de mémoire entiers (jusqu'à 64 Koctets). Cette opération ne ralentit l'unité centrale (CPU) que très faiblement. Différentes sources d'interruption pourront faire office de sources de signal d'horloge DMA.

Il existe différentes possibilités de connexion de mémoire externe :

1. bus multiplexé d'une largeur de 8 ou de 16 bits,
2. bus séparé d'une largeur de 8 ou de 16 bits et
3. bus multiplexé (mode « *entire space* ») d'une largeur de 8 bits et bus d'adresse de 8 bits seulement. Ce mode convient idéalement aux périphériques externes, sachant qu'en cas d'utilisation de lignes de sélection de puce (CS) chacune de ces lignes se voit attribuer 256 octets.

On pourra, avec chacune des 4 lignes CS, intercaler un état d'attente (*wait state*). On dispose en outre, pour des modules de mémoire ou des périphériques plus lents, d'une ligne RDY (*ReadY*) qui, par sa connexion à BCLK, permet d'intercaler un

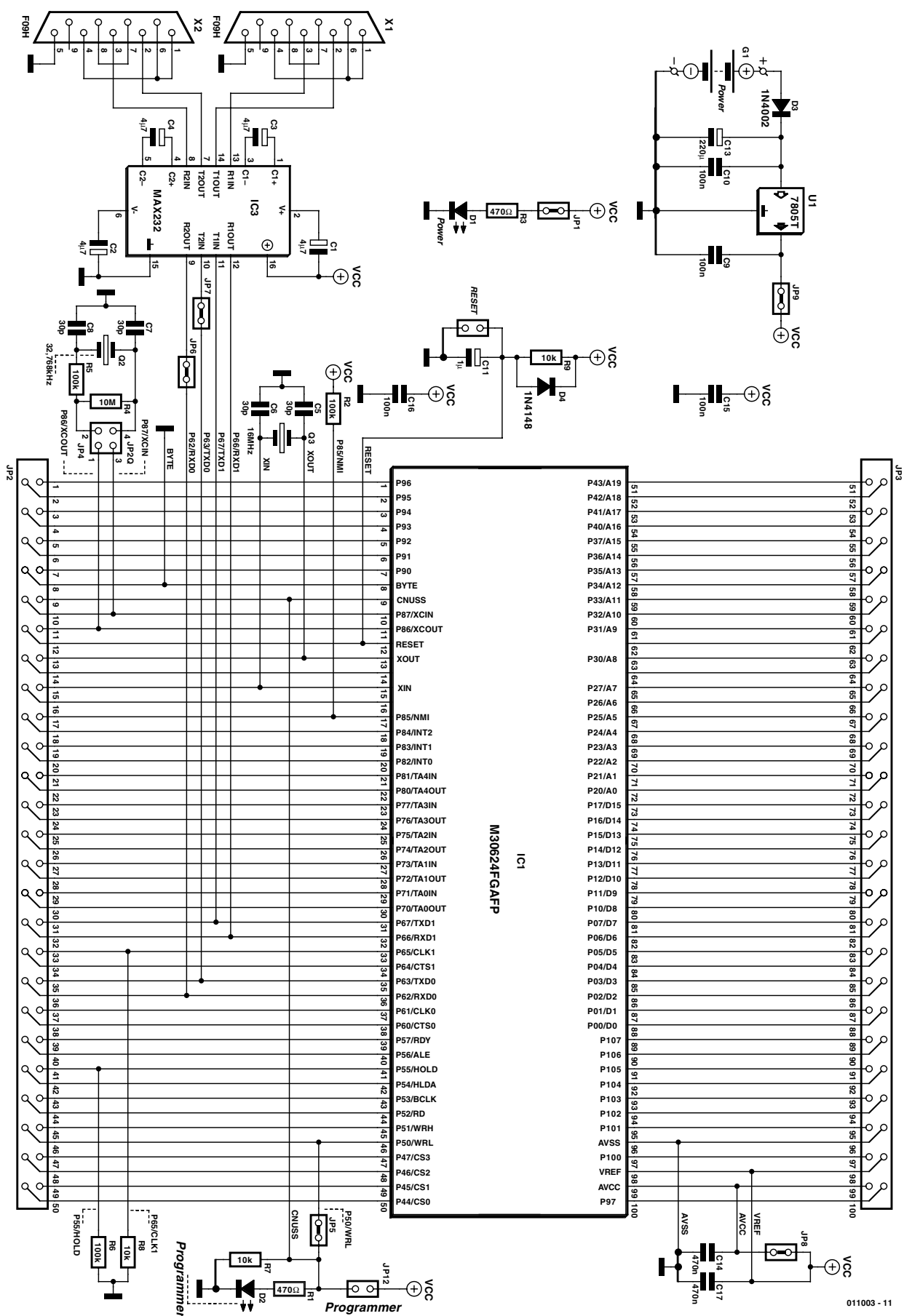


Figure 2. Schéma électronique de la carte d'évaluation.

nombre additionnel quelconque d'états d'attente.

Il existe, en marge de RDY, une entrée HOLD qui permet de stopper le système et fait basculer les lignes d'adresses et de données à l'état de haute impédance (*tri-state*).

La CPU

Un premier coup d'oeil à la structure des registres rappelle celle d'un 68000, un processeur 16 bits du siècle dernier. Un examen plus approfondi montre qu'effectivement nombre d'instructions sont similaires à celle de ce microprocesseur de Motorola.

Il n'en reste pas moins que le M16C traite les instructions de façon plus parallèle et avec moins de cycles d'horloge par instruction, ce qui le rend, à fréquence d'horloge identique, sensiblement plus rapide. En moyenne, une instruction d'un programme ne requiert que 3 cycles d'horloge pour son exécution, sachant que le mode 4 *channel instruction queue* (queue d'instructions à 4 canaux) ne fait qu'accroître les performances. De ce fait, à une fréquence de quartz de 16 MHz, le M16C/62 atteint une puissance de calcul de 8 MIPS (*Million Instructions Per Second*, nous vous laissons le plaisir de traduire cette notion).

Le set d'instruction est doté de caractéristiques spéciales qui permettent de rendre le code plus compact. Il peut ainsi, par exemple, par déclarer MOV immédiate 0,R0 à l'aide d'un seul octet. Le set d'instruction compacte également les valeurs immédiates (dont la description requiert 3 bits) au niveau du code opération (l'ensemble de l'instruction ne requérant en général que 2 octets), ceci en vue d'économiser de l'espace mémoire. Il connaît les modes d'adressage *General* et *Short* ce qui lui permet de réduire de moitié la taille du code d'un modèle de CPU plus ancien telle que la MELPS7700. À l'image du 6800, la CPU M16C dispose de registres FB et SB. Par le biais de ces registres la CPU est en mesure, lors de l'adressage, d'adresser avec 8 bits, voire moins encore, des adresses qui, normalement, requièrent 16 bits. Ceci permet à nouveau de gagner de l'espace mémoire et d'accroître la vitesse de

traitement. Le noyau de la CPU comporte un registre ISP et un registre USP qui permettent d'isoler l'un de l'autre la pile du système d'exploitation et la pile des interruptions matérielles. (*hardware interrupt stack*).

La fonction de comparaison (*compare*) a été partant étendue de sorte que l'on a, en fonction de l'état de certains indicateurs (*flag*), exécution de saut (*Jump on condition*) ou transfert de bits (*Bit move condition*). L'ajout de 2 interruptions d'égalité d'adresse (*address match*) permet, dans le cadre du système de développement présenté ici par exemple, le positionnement de points d'arrêt (*breakpoint*) dans le programme utilisateur.

Signalons une spécificité remarquable, l'instruction RMPA (*Repeat Multiply & Add*) qui permet de réaliser des filtres numériques qui, grâce à la présence de convertisseurs A/N et N/A rapides, sont même utilisables dans le domaine audio.

La mémoire Flash de la série M16C/62 est réalisée en technologie DINOR, ce qui signifie qu'elle est programmable à la tension d'alimentation (3 ou 5 V), en mode soit sériel synchrone, soit sériel asynchrone, soit encore en parallèle voire même par le biais d'un programme spécifique (mode CPU Read/Write). La programmation de la mémoire se fait page par page (*page-wise*), c'est-à-dire 256 octets d'un coup. Lorsque l'on sait que cette opération ne dure qu'entre 6 et 8 ms, un programme de 256 Koctets ne prend qu'entre 6 et 8 s pour être chargé.

Une fois la mémoire (de programme) « flashée », la version 3 V, baptisée M30624FGMFP peut travailler à une tension d'alimentation comprise entre 2,4 et 3,6 V, la version 5 V, dite M30624FGAFP ayant elle une plage de tensions d'alimentation plus large, allant de 2,7 à 5,5 V. Plus la tension d'alimentation est élevée, plus la fréquence d'horloge du microcontrôleur pourra elle aussi être importante.

Mitsubishi a doté tous les microcontrôleurs Flash M16C/62 d'une mémoire Flash additionnelle de 8 Koctets baptisée *Flash Bootloader*. Le fabricant fournit cette Flash dotée d'un programme de démarrage (le *Bootloader*) qui permet de charger un programme sériellement soit en

synchrone soit en asynchrone (et même en CAN dans le cas du M306N0FGTFF). Ce bout de programme peut être effacé à l'aide d'un programmeur parallèle, la Flash Bootloader pouvant ensuite être dotée d'un programme de démarrage propre à l'utilisateur.

La carte

Comme, depuis la fin de l'année dernière, il existe, sur Internet, un débogueur au niveau source tournant sous Windows 95/98/NT, le moniteur de ROM correspondant, le programme de flashage ainsi qu'une version d'essai du compilateur C de Mitsubishi (pratiquement sans restriction pendant jusqu'à 4 mois) et d'une version d'essai du compilateur C chez Tasking (jusqu'à de l'ordre de 16 Koctets sans limite de temps), on pourra fort bien envisager d'utiliser ces composants pour créer un mini-émulateur.

Les coûts de logiciel étant nuls et ceux du matériel relativement faibles, le distributeur de Mitsubishi Electric en RFA, la société Glyn (il nous a été impossible de tracer un quelconque distributeur de cette marque en France en dépit de nos tentatives de contact avec Mitsubishi), propose une carte d'évaluation au prix coûtant de 43,10 € seulement (+ TVA et frais de port). L'envoi comporte : la platine dotée de ses composants ainsi que les programmes évoqués plus haut sans oublier des programmes de démonstration et des aides pour des applications potentielles. S'il vous venait à l'idée de réaliser la platine vous-même sachez que l'on trouve les fichiers (au format .pdf et Eagle) de la platine (**figure 1**) au téléchargement sur le site Elektor (www.elektor.presse.fr) dans la rubrique correspondant à ce numéro. La totalité des liens vers les programmes, les fiches de caractéristiques et tutti quanti sont disponibles par le biais de la page d'accueil du contrôleur de Mitsubishi sise à l'adresse : <http://www.m16c.de>.

Après initialisation (*Reset*), le microcontrôleur M16C connaît 3 modes de fonctionnement dans lesquels la carte peut travailler (croquis comparatif de la **figure 2**) :

1. Le mode « Single Chip/Memory Expansion » (la ligne CNVss se trouve au niveau bas)
2. Mode « Microprocesseur » (CNVss se trouve au niveau haut)
3. Mode « Bootloader » (les lignes CNVss et EPS se trouvent au niveau haut, HOLD au niveau bas). Ce mode permet le « flashage » en circuit sériel synchrone ou asynchrone.

En cas d'implantation du cavalier JP12 sur la carte il est partant très facile de passer du mode « Single Chip » au mode de programmation. Dans ce dernier mode, la LED D4 s'allume. Si l'on a, en outre, laissé l'embase de court-circuit JP5 ouverte et qu'on l'a forcée au niveau bas à l'aide d'une résistance, on pourra également travailler en mode « Microprocesseur ».

Le flashage du microcontrôleur se fait par le biais de l'UART1 (véhicule défini par le programme *Bootloader*); il est en outre possible de procéder à un débogage à l'aide du moniteur ROM. Pour cela, le fameux MAX232 convertit les niveaux de tension et force, par le biais d'une résistance de forçage au niveau haut (*pull up*) intégrée la sortie TXD1 au niveau haut pendant l'opération de flashage (cette sortie devient, sous l'égide du programme *Bootloader*, momentanément une entrée). Les cavaliers JP6 et JP7 permettent d'interconnecter UART0 au MAX232.

L'alimentation fait appel à un 7805 classique; il est possible, lors de mesure par exemple, de déconnecter la ligne de la tension d'alimentation à l'aide du cavalier JP9. L'entrée de tension de référence VREF est reliée, au travers du cavalier JP8, à la ligne d'alimentation positive (VCC), mais rien n'interdit non plus, pour peu que ce cavalier ne soit pas implanté, de la forcer à une valeur de tension différente. Le circuit d'initialisation prend la forme d'un réseau RC (résistance/condensateur). La diode D4 sert à décharger le

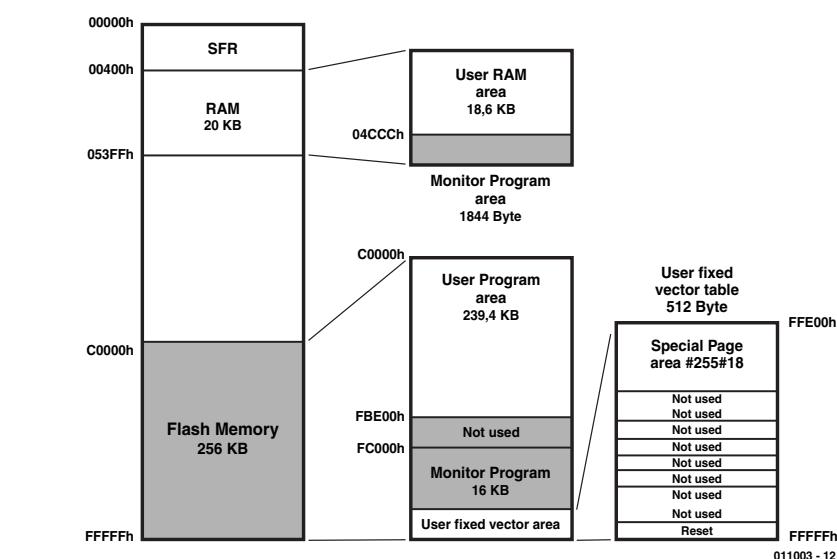


Figure 3. Cartographie mémoire du M16C.

condensateur lors de la coupure de la tension d'alimentation. Ce type de circuit d'initialisation convient à la majorité des applications. AVCC et les 2 lignes VCC sont reliées à la tension d'alimentation. Le M16C/62 est extrêmement bien protégé contre les parasites, des parasites « *ground-burst* » ne dépassant pas 5 kV n'ont pas de quoi « impressionner » ce microcontrôleur. Le M16C est en outre très peu gour-

mand, ne consommant, sous 5 V et à 16 MHz et 0 état d'attente (*Wait*), que 32,5 mA (M30624FGAFP), à 3 V, 10 MHz et 0 état d'attente que 12 mA (M30624FGMFP). En mode « Stop » à 25 °C, Mitsubishi annonce un maximum de 1 µA. La puce du microcontrôleur étant du type statique, elle pourra être cadencée à n'importe quelle fréquence (si tant est qu'elle soit inférieure à la valeur maximum bien entendu).

M16C/62 avec Flash Panorama des différents types courants				
Boîtier QFP 100 broches 0,65 mm				
M30624FGAFP	Flash DINOR 256 Koctets	5V	SRAM 20 Koctets	
M30624FGMFP	Flash DINOR 256 Koctets	3V	SRAM 20 Koctets	
M306N0FGTFP	Flash DINOR 256 Koctets	5V	SRAM 10 Koctets	2x CAN
M30620FCAFP	Flash DINOR 128 Koctets	5V	SRAM 10 Koctets	
M30620FCMFP	Flash DINOR 128 Koctets	3V	SRAM 10 Koctets	
M3062GF8NFP	Flash HDINOR* 64 Koctets	3V	SRAM 8 Koctets	
Boîtier QFP 100 broches 0,5 mm				
M30624FGAGP	Flash DINOR 256 Koctets	5V	SRAM 20 Koctets	
M30624FGMGP	Flash DINOR 256 Koctets	3V	SRAM 20 Koctets	
M30620FCAGP	Flash DINOR 128 Koctets	5V	SRAM 10 Koctets	
M30620FCMGP	Flash DINOR 128 Koctets	3V	SRAM 10 Koctets	
M3062GF8NGP	Flash HDINOR* 64 Koctets	3V	SRAM 8 Koctets	
* La nouvelle Flash HDINOR Flash (0,35 µm) est, en mode CPU R/W, programmée non pas page par page (page wise) mais mot par mot (word wise).				

Le moniteur ROM

On pourra télécharger le Moniteur d'Internet tant sous la forme de fichier *.mot que son code-source. Dans la mémoire Flash il occupe le domaine allant de FC000_{HEX} à FFFFF_{HEX}, en RAM statique (SRAM) le domaine allant de 5300_{HEX} à 53FF_{HEX} (figure 3).

Le programme Moniteur attend, après mise sous tension du système, le programme de déboguage PC KD30 sur l'UART1. La fréquence de communication est de 38 400 bds dérivée d'une fréquence de quartz de, selon le cas, 16 ou 10 MHz. On pourra cependant opter pour une fréquence de quartz quelconque. L'unité centrale du microcontrôleur est flashée lors de chaque téléchargement. Le bloc supérieur est cependant protégé (automatiquement !) vu qu'il ne saurait être question d'écraser le Moniteur ROM. Le vecteur de Reset du programme utilisateur (qui se trouve de l'adresse FFFFC_{HEX} à FFFFF_{HEX}) est transféré à l'adresse

de poids fort du premier bloc inférieur à suivre (adresse FBFFC_{HEX} à FBFFF_{HEX}), ceci de manière à positionner, après chargement du programme utilisateur, le compteur ordinal directement à la bonne adresse de début de programme.

Le Moniteur ne reconnaît pas les interruptions de chien de garde (*watchdog*), ni celles de *Adress-Match*, *NMI* (*Non Masquable Interrupt* = interruption non-masquable) *Overflow* (Dépassement) et *Undefined-instruction* (instruction non-définie). Le mode d'extension de mémoire (*Memory expansion*) est utilisable, la broche HOLD ne devant cependant pas se trouver au niveau bas (la carte comporte une résistance de forçage au niveau bas (*pull down*)).

(011003-1)

Dans l'article (suite et fin) du mois prochain nous nous intéresserons au programme de flashage et au compilateur C de Mitsubishi.

Sources:

Produits Mitsubishi

Mitsubishi Electric Europe
Branche française
25, Boulevard de Bouvets
92741 Nanterre Cedex, France
Tél. : (01.55.68.56.79)
Fax. : (01.55.68.57.39)
(nous n'avons pas pu savoir dans quelle mesure le bureau français vendait les composants Mitsubishi à l'unité)

Circuits à l'unité

Reichelt Elektronik
Elektronikring I
D 26452 Sande (RFA)
Tél. : +49 (0) 4422-955-333
Fax. : +49 (0) 4422-955-111

pour des quantités plus importantes ainsi que la carte, le distributeur de Mitsubishi en RFA :

Glyn GmbH & Co. KG
Am Wörtsgarten 8
D 65510 Idstein (RFA)
Tél. : +49 (0) 6126-590-255
Fax. : +49 (0) 6126-590-155
E-mail : mcu@glyn.de

Pupitre d'éclairages

À huit canaux

à la technique : Jürgen Mack

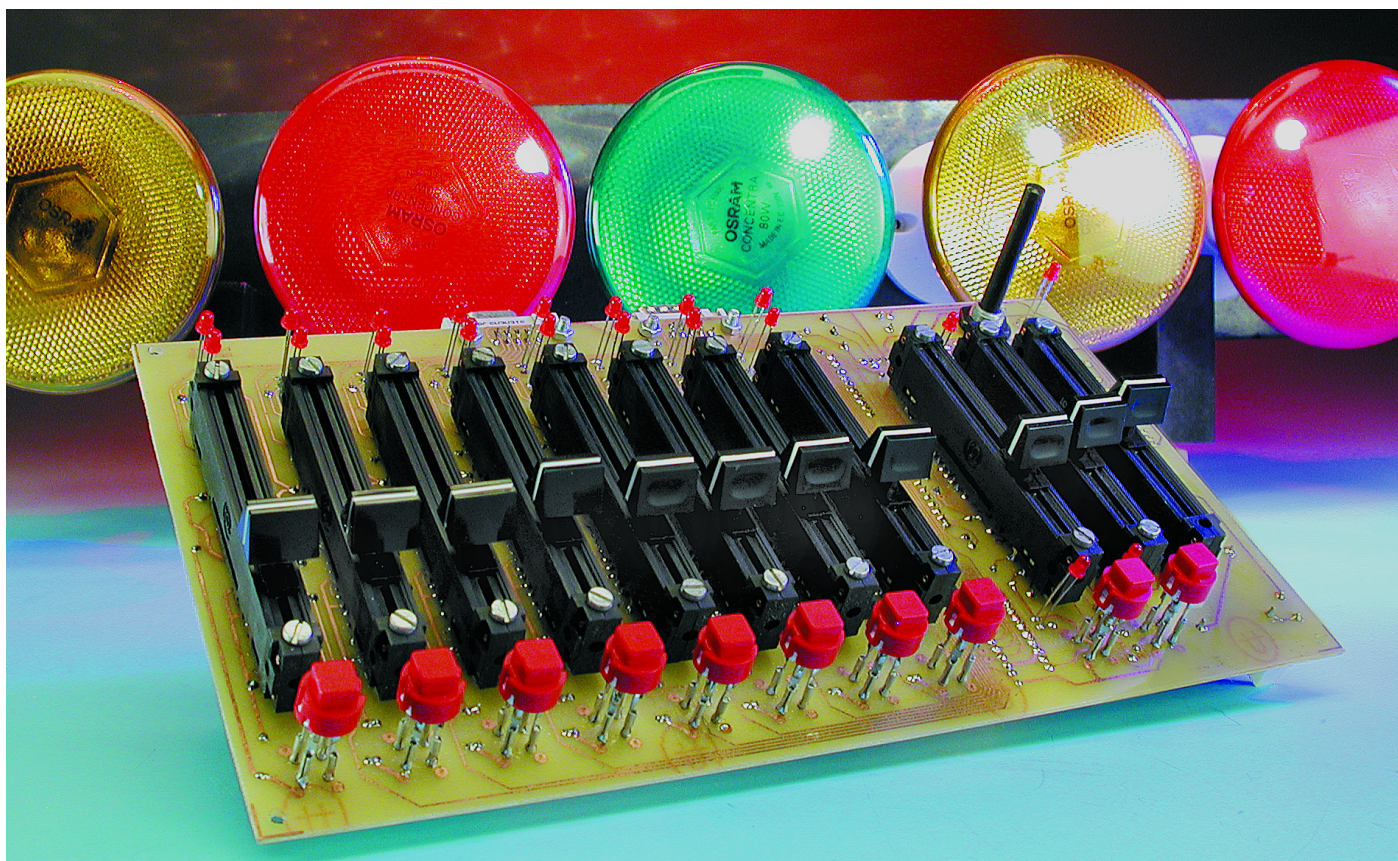
Ce panneau de réglage d'éclairages de scène à 8 canaux, d'un prix très abordable, dispose de toutes les possibilités de jeu et compte même quelques ajouts, tel ce générateur de rythme ou l'indication lumineuse de la tension de sortie. De plus, le montage est extensible, avec la latitude de lui adjoindre un PC.

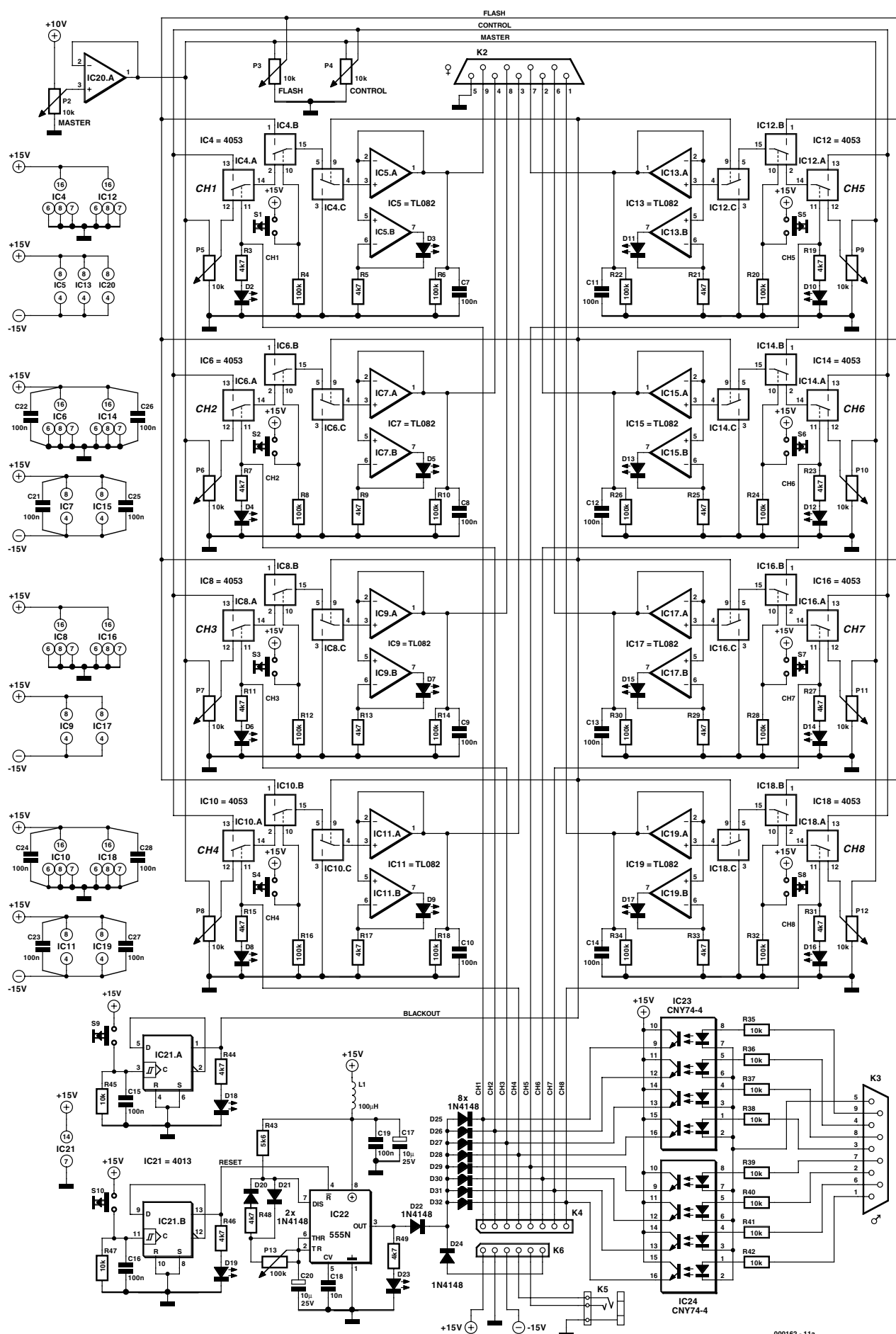
Les théâtres, les salles de concert, les discothèques, ce qui leur donne éclat, c'est la lumière, le jeu d'orgue, la commande sophistiquée de l'éclairage, de la couleur. Il fut un temps où l'on éteignait la salle pour allumer

les spots. Mais voilà des lustres que la lumière fait partie intégrante du spectacle. Ce que la table de mixage est pour le son, le jeu d'orgue lumineux en exécute la mission pour l'éclairage. Ce pupitre de com-

mande, qui fournit les signaux de réglage des atténuateurs et commutateurs, constitue un genre d'interface spécialisée entre l'opérateur et l'installation d'éclairage.

On en trouve de différentes sortes et exécutions, de la minuscule boîte à la régie numé-





000162 - 11a

rique la plus impressionnante, format bureau directorial. Le modèle de base, pour une salle petite ou moyenne, compte quatre à six canaux, chacun muni d'un potentiomètre (à glissière), les accessoires se résument souvent à un réglage général pour influencer la totalité de l'éclairage et, sur chaque canal, un poussoir appelé flash, question de faire donner brièvement à un projecteur sa pleine intensité, quelle que soit la position du potentiomètre.

Possibilités

L'auteur a voulu davantage qu'une version spartiate, pour cette table analogique.

D'entrée de jeu, il a opté pour huit canaux. Voilà qui en fait déjà un appareil plus universel : un potentiomètre à glissière sur chaque canal pour fournir au gradateur une tension dans la plage standardisée de 0 à 10 V. Il n'a pas oublié la commande générale qui agit sur toutes les voies simultanément.

Les boutons de flash, il les a aussi considérés comme indispensables, il les a même flanqués d'un autre potentiomètre linéaire pour en modifier l'éclat, ne plus être lié à 100 % de l'intensité. À l'inverse, il y a un bouton de « *blackout* » pour l'obscurcissement de la scène, en forçant toutes les sorties à 0 V.

Et puis, sur chaque canal, on trouve encore une entrée de contrôle, de manière à permettre la commande par la musique. En outre, une série de photo-coupleurs et un connecteur sub-D autorisent le branchement d'un PC comme fournisseur de

signaux de commande, dont le niveau est réglable par un potentiomètre indépendant.

Et si la musique s'arrête ? D'une pression du doigt, on enclenche un petit générateur de rythme qui va animer l'intensité des spots, à une fréquence réglable par un autre potentiomètre, rotatif celui-ci.

Quoi d'autre encore ? Le pupitre est doté d'une série d'indicateurs qui renseignent l'éclairagiste sur les fonctions en cours. Tous les canaux possèdent une LED dont la luminosité imite la tension de sortie réglée. Ce n'est pas indispensable, mais bien utile quand on travaille dans l'obscurité ou la pénombre. Huit LED supplémentaires indiquent l'activité sur les entrées de commande et d'autres signalent la mise en fonction du *blackout* et le rythme du générateur.

Coup de proje sur le schéma

Au premier regard, le schéma de la **figure 1a** peut sembler horriblement compliqué, mais en réalité, il compte huit blocs identiques, les réglages de canaux.

Nous allons tout d'abord nous offrir un panoramique de haut en bas sur le schéma pour avoir une idée des divers éléments qu'il comporte. Nous mettrons en lumière les détails plus tard.

En haut, à gauche, le *master*, le réglage général (P2). Il se situe entre l'alimentation et le dosage des voies et leur fournit une tension comprise entre 0 et 10 V. Vers la droite, nous trouvons les potentiomètres pour le

flash et le contrôle de niveau (P3 et P4), puis le connecteur K2 auquel aboutissent les tensions de commande de tous les canaux.

Plus bas, nous rencontrons les huit réglages de ces canaux. Ils représentent ensemble environ les 2/3 du dessin. À gauche, les raccords d'alimentation pour les différents circuits intégrés et les inévitables condensateurs de découplage.

Encore plus bas, à gauche, le bistable pour la fonction de *blackout* (IC21.A) ; une pression sur S9 l'enclenche ou l'arrête et la LED D18 indique la situation. Immédiatement en dessous, un bistable identique (IC21.B), actionné par S10, lance le générateur de rythme construit autour de IC22. C'est D19 qui affiche l'état de la bascule. P13 modifie la fréquence de l'oscillateur, que la LED D23 permet de vérifier.

Au milieu et à droite se trouvent les accessoires. Toutes les entrées de commandes sont disponibles sur le connecteur K4, en vue d'extensions éventuelles. K6 rend possible la commande musicale, par exemple. Les alimentations ± 15 V y sont également accessibles et le connecteur jack K5 permet d'introduire (ou de prélever) un signal. Le connecteur Sub-D à 9 broches K3 est destiné à la liaison sérielle avec un PC, entre autres. Des signaux externes de commande peuvent transiter sous isolation galvanique grâce aux photo-coupleurs IC23 et IC24.

Fonctionnement

Après ce travelling, nous pouvons nous focaliser sur les vedettes du schéma, avec en tête de distribution, le réglage de voie. En fait, pour certains des figurants passés en revue, nous avons déjà déclamé leur rôle en entier. Reprenons par le potentiomètre général P2, relié à une tension de +10 V délivrée par l'alimentation, si bien que l'on retrouve sur le curseur une tension comprise entre 0 et 10 V. Pour éviter toute réaction, la tension du curseur est d'abord tamponnée par IC20.A pour servir de niveau d'entrée aux réglages de flash et de contrôle, P3 et P4, communs à tous les canaux, ainsi que pour les potentiomètres de canaux individuellement, P5 à P12.

Puisque les réglages de canaux sont tous les huit identiques, nul besoin de répétition, nous ne parlerons que du canal 1, la partie centrée sur IC4 et IC5.

C'est avec P5 que l'on règle le niveau souhaité. La tension prélevée au curseur trouve d'abord sur son passage trois interrupteurs électroniques (IC4.A, B et C). Aussi longtemps qu'aucune fonction spéciale n'est activée, elle les traverse sans encombre, après quoi, elle est tamponnée par IC5 et envoyée au connecteur de sortie K2. Si maintenant on envoie un

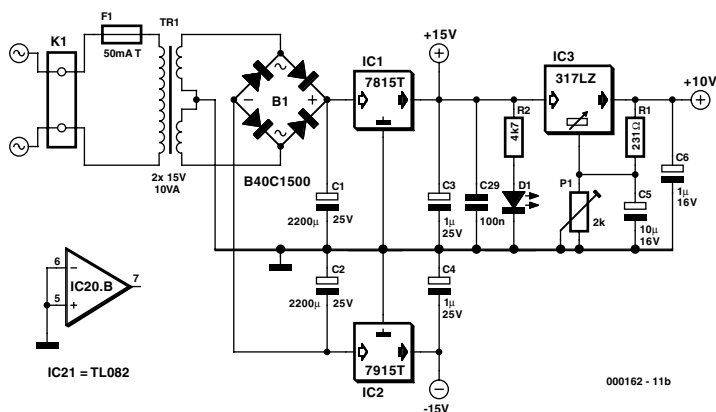
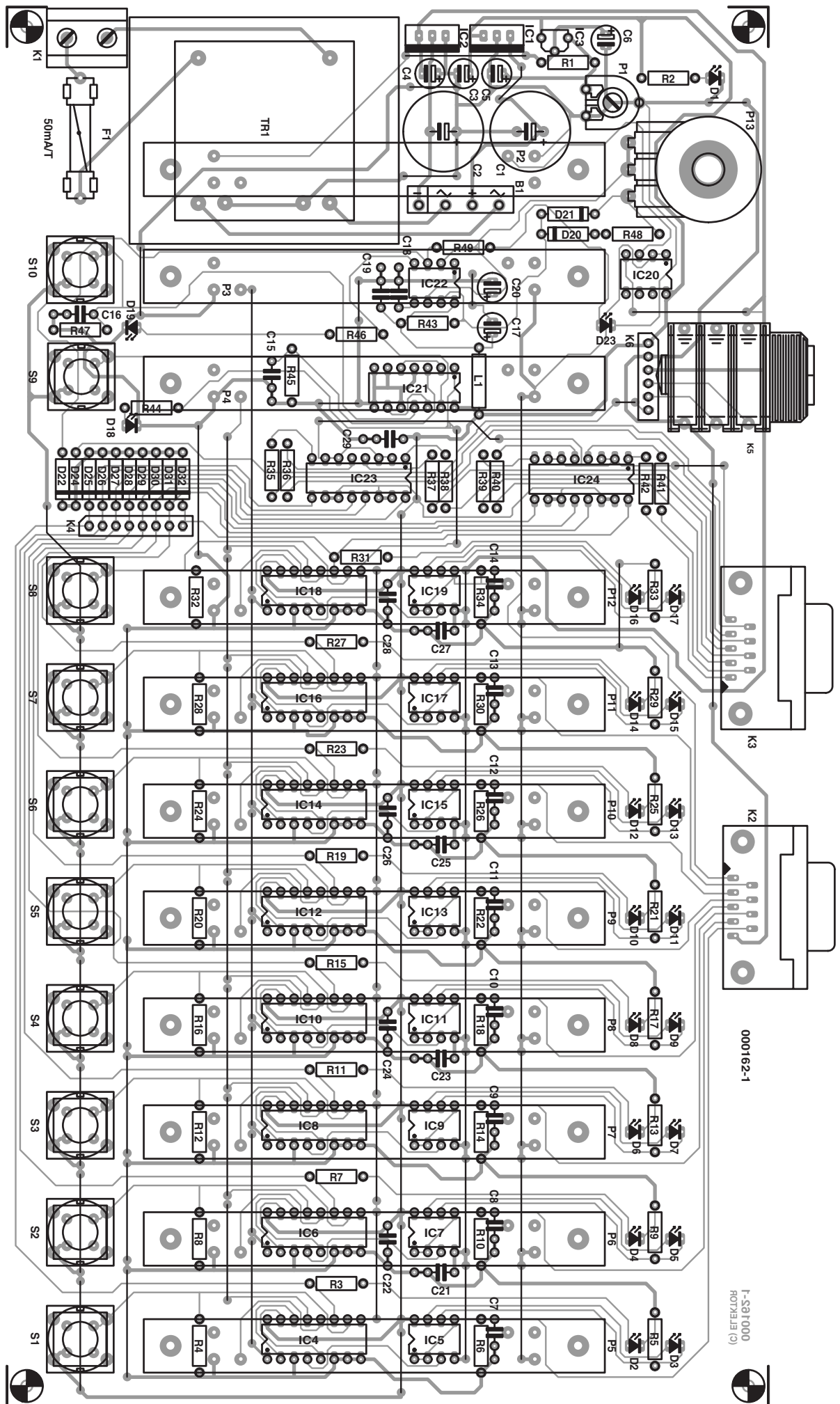


Figure 1. Vaste plutôt que compliqué, le schéma ci-contre. Pour l'alimentation, par exemple, ci-dessus, on trouve trois régulateurs de tension.



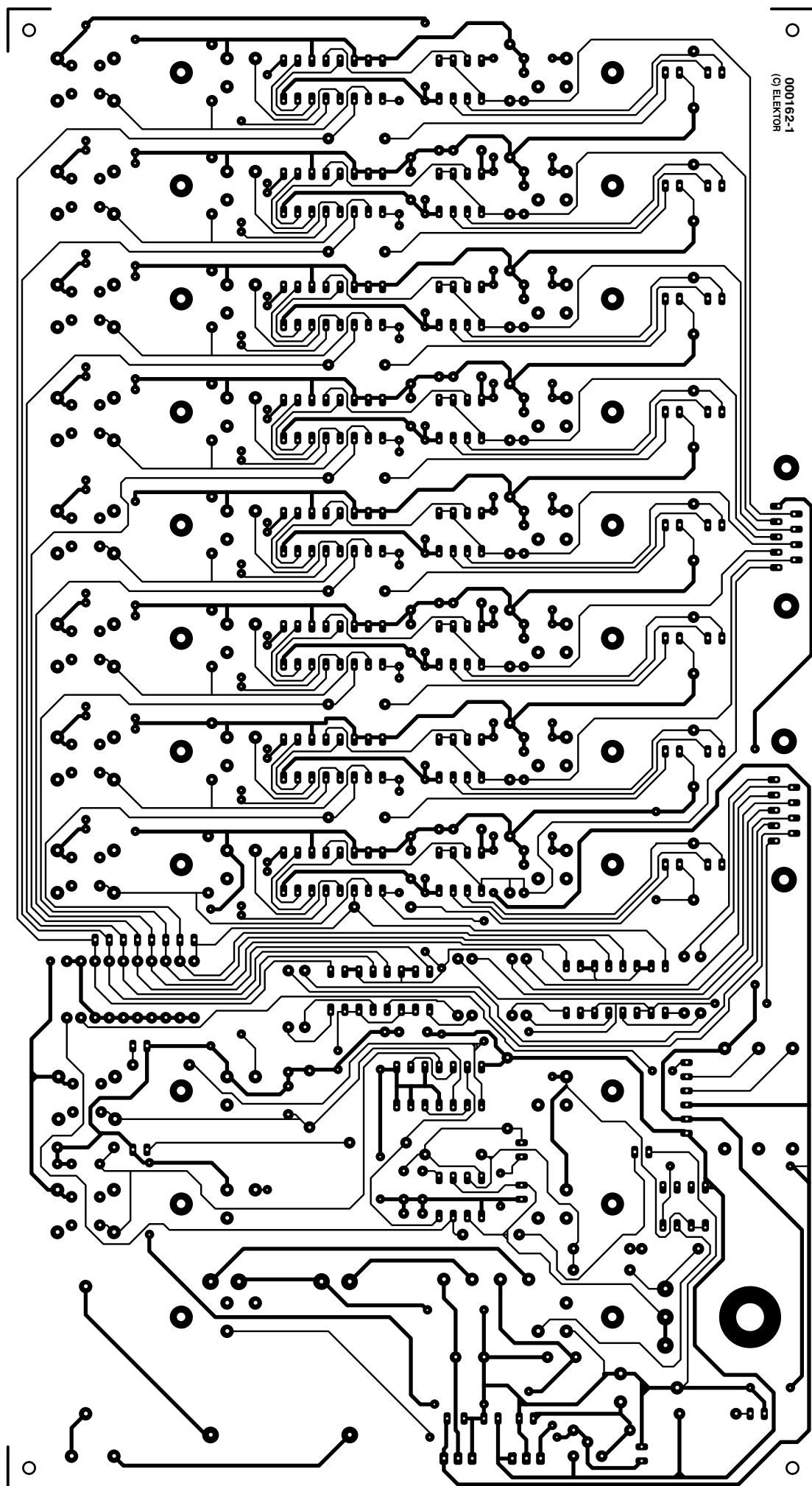


Figure 2. La platine, que vous trouverez aux adresses habituelles, offre asile à tous, y compris connecteurs, régulateurs et alimentation.

Liste des composants

Résistances :

R1 = 231 Ω
R2, R3, R5, R7, R9, R11, R13, R15, R17, R19, R21,
R23, R25, R27, R29, R31, R33, R44, R46, R48,
R49 = 4k Ω
R4, R6, R8, R10, R12, R14, R16, R18, R20, R22,
R24, R26, R28, R30, R32, R34 = 100 k Ω
R43 = 5k Ω
R35 à R42, R45, R47 = 10 k Ω
P1 = ajustable 2k Ω
P2..P11 = potentiomètre linéaire 10 k Ω
(Radiom, Conrad 44 14 49-60)
P13 = potentiomètre 100 k Ω

Condensateurs :

C1, C2 = 2 200 μ F/25 V radial
C3, C4 = 1 μ F/25 V radial
C5, C17, C20 = 10 μ F/25 V radial
C6 = 1 μ F/16 V radial
C7..C16, C19, C21 à C29 = 100 nF
C18 = 10 nF

Bobines :

L1 = 100 μ H

Semi-conducteurs :

B1 = B40C1500
D1 à D18 = LED 3 mm (rouge)
faible courant
D2 à D17, D19, D23 = LED 3 mm (verte)
faible courant
D20 à D22, D24 à D32 = 1N4148
IC1 = 7815
IC2 = 7915
IC3 = LM317LZ
IC5, IC7, IC9, IC11, IC13, IC15, IC17, IC19,
IC20 = TL082
IC4, IC6, IC8, IC10, IC12, IC14, IC16,
IC18 = 4053
IC21 = 4013
IC22 = 555
IC23, IC24 = CNY17-4

Divers :

S1 à S10 = bouton-poussoir unipolaire à
contact travail tel que, par exemple, D6-R
F1 = fusible 50 mA T + porte-fusible
TR1 = transfo encartable 2x15 V/10 VA au
secondaire tel que, par exemple, ERA
EL48/16,8
K1 = bornier encartable à 2 contacts au pas
de 7,5 mm
K2 = embase Sub-D femelle à 9 contacts
encartable en équerre
K3 = embase Sub-D mâle à 9 contacts
encartable en équerre
K4 = embase autosécable à 1 rangée de
8 contacts
K5 = connecteur jack stéréo 6,35 mm
encartable
K6 = embase autosécable à 1 rangée de
6 contacts
boîtier 310 x 170 mm tel que, par exemple,
Teko 52 39 84-60

signal sur l'entrée de commande, de l'extérieur via K4 ou en lançant le générateur de rythme, le commutateur IC4.A est activé par la ligne CH1, la liaison avec P5 s'interrompt, le signal de contrôle est transmis à la sortie et c'est P4 qui en détermine le niveau. La LED D2 est allumée.

Une action sur le poussoir S1 fait s'ouvrir IC4.B et le niveau de flash réglé par P3 est transmis en sortie.

La fonction *blackout* travaille sur tous les canaux en même temps. Si elle est mise en service par le poussoir S9, IC4.C s'ouvre, interrompant la liaison et l'entrée du tampon IC5.A est mise à la masse. La sortie est alors à 0 V, indépendamment de tous les autres réglages.

Pour l'indication de la tension de sortie, on fait appel à un autre amplificateur opérationnel, IC5.B. Il fonctionne en convertisseur tension-courant et transforme la plage de réglage de 0 à 10 V en une variation presque linéaire de la luminosité de la LED D3, qui comme toutes les autres, doit être un modèle à faible consommation.

En sortie, on trouve encore C7 dans un filtre passe-bas destiné à éliminer toute influence du câble. La résistance R6 ferme le circuit en absence de charge.

Nous avons ainsi parcouru tout le dispositif, la seule partie pour laquelle nous avons encore une tirade au scénario, guère davantage, c'est le générateur de rythme. Vous dire que ce temporisateur 555 (IC22) est monté en multivibrateur astable classique « comme dans le livre ». Que sa fréquence se règle par P13 entre 1 et 10 Hz, que D20 et D21 assurent un rapport cyclique de 50 %. Encore une précaution, qui n'est pas oratoire, L1 et C17 empêchent que les bavardages du 555 ne troublent la quiétude de la partie analogique du montage. Par l'entremise de D22, qui empêche la rétroaction, le signal de sortie quitte la scène par le réseau de diodes D25 à D32, lequel régit simultanément toutes les entrées de commande.


Alimentation

Le pupitre d'éclairagiste réclame pas moins de trois tensions d'alimentation différentes : une de +15 V pour les interrupteurs électroniques, une

symétrique de ± 15 V pour les amplificateurs opérationnels, sans oublier un +10 V précis comme tension de référence des réglages.

Ainsi qu'on le voit sur la **figure 1b**, les circuits d'alimentation sont, malgré tout, très simples. L'ensemble habituel transformateur, pont redresseur et condensateur électrolytique fournit d'abord une tension redressée mais non stabilisée de 2×19 à 20 V, suffisante pour attaquer les régulateurs tripodes 7815T et 7915T (IC1 et IC2) dont on obtient les +15 V et -15 V bien propres.

À l'issue de nos auditions, c'est au régulateur LM317LZ que nous avons attribué le rôle de stabilisateur à 10 V. Pour lui éviter tout essoufflement, il a le droit de puiser, à la source +15 V déjà stabilisée, le courant nécessaire. Sa tension de sortie s'ajuste à l'aide de P1. Pour s'affranchir des pertes éventuelles, mieux vaut mesurer la tension sur le connecteur de sortie K2. On raccorde le voltmètre entre la broche 5 (masse) et la broche 9, par exemple (sortie canal 1). On place P2 et P5 en butée, à fond d'échelle, et on règle P1 pour obtenir exactement 10 V.

ELEKTOR		
230V ~	50Hz	
No. 000162		
F = 50 mA T		

Belle besogne, le soudage !

La **figure 2** vous dévoile le côté pistes et le côté composants de la platine destinée à servir de base au pupitre de commande d'éclairages. Elle supporte la totalité du montage, y compris l'alimentation et, comme il faut y caser onze potentiomètres à glissière, elle n'est pas spécialement petite.

Un mot, cependant, en prologue à l'examen de la construction. Pour limiter le coût, cette platine ne comporte qu'une seule face imprimée. La conséquence, c'est qu'il faudra câbler soi-même les ponts, il y en a pas moins de 79 ! Ajoutez à cela qu'il y aura des composants sur la face du même nom, mais aussi côté cuivre,

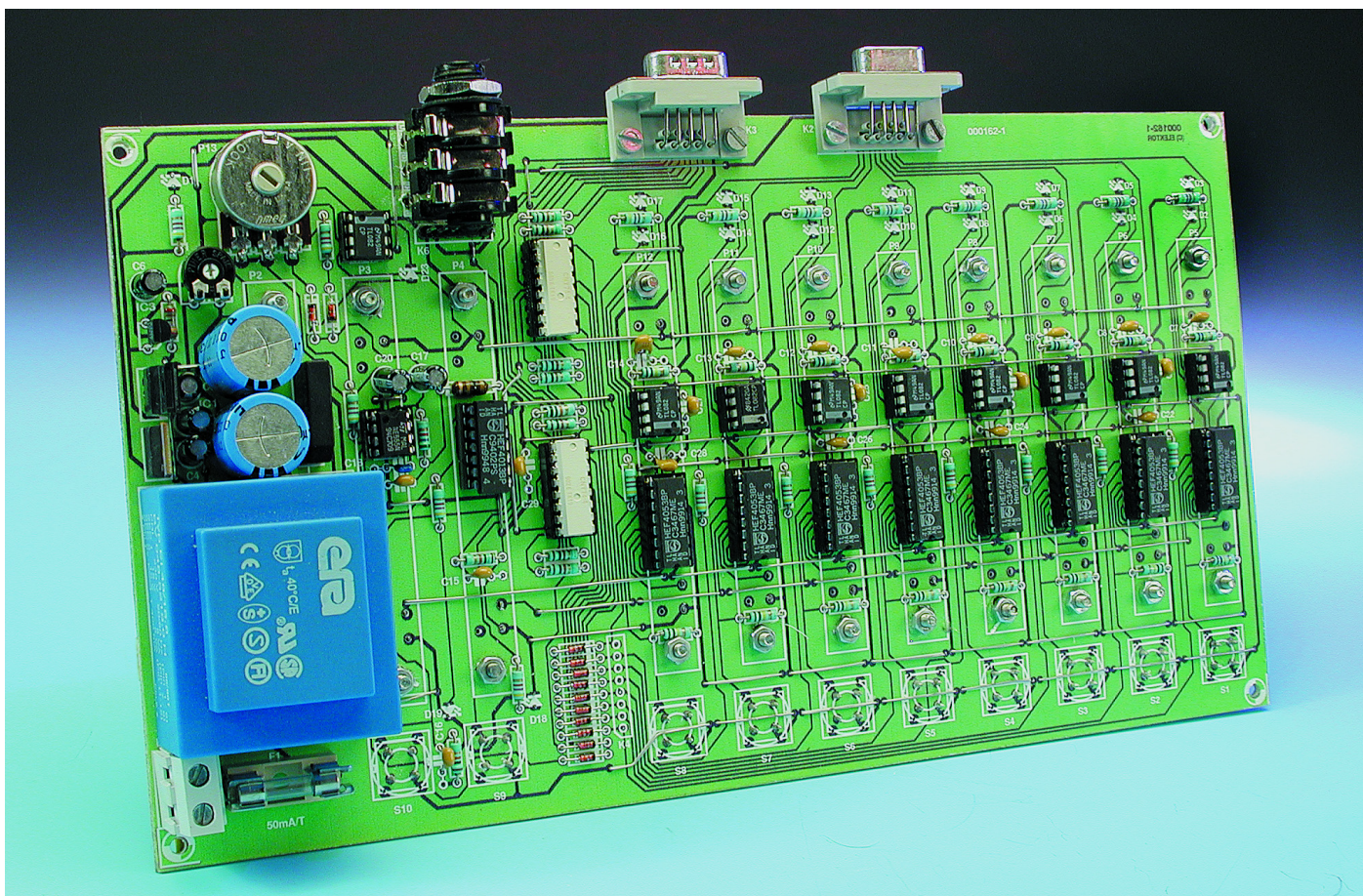


Figure 3. Côté composants, façon de parler, puisqu'il y en a, ici, sur les deux faces !

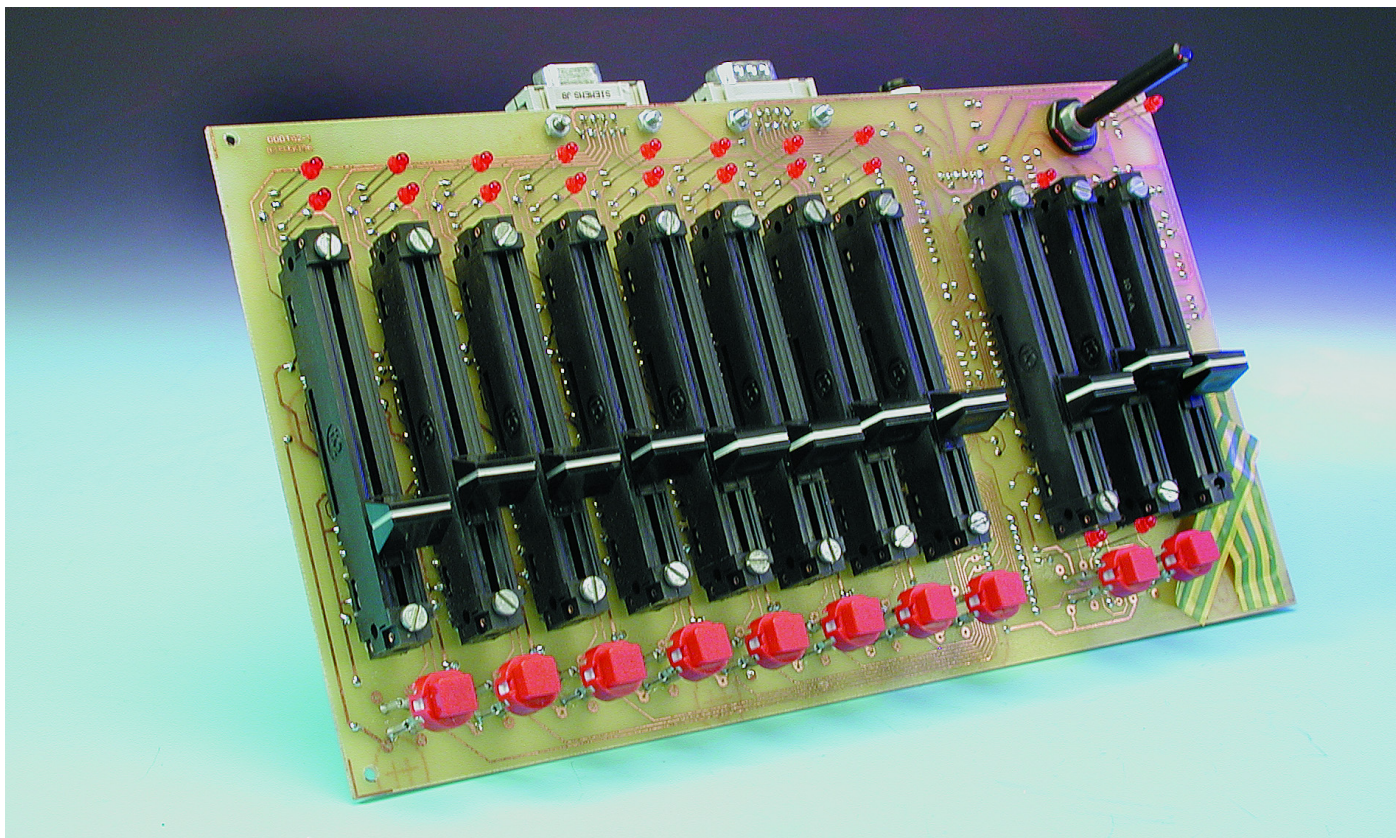
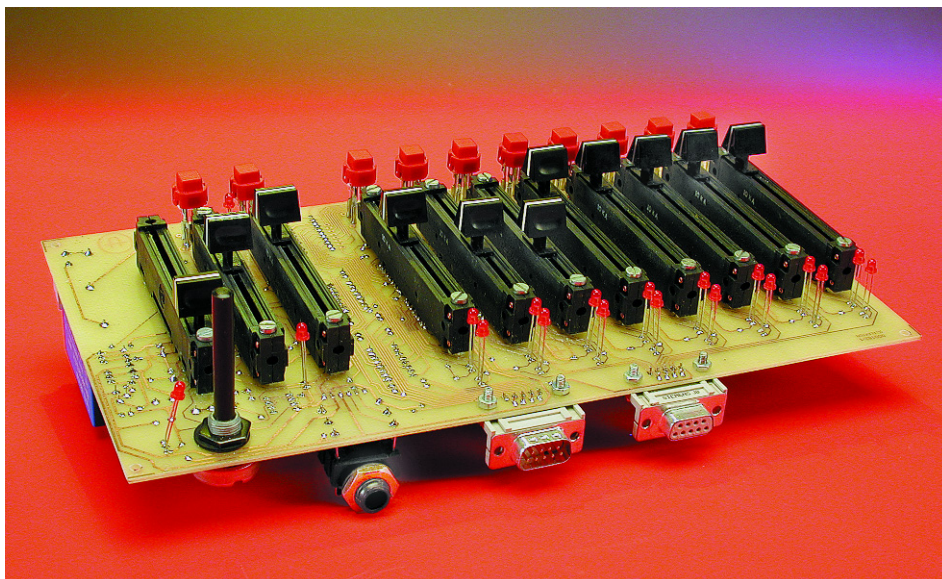


Figure 4. Les interrupteurs, potentiomètres à glissière et LED, c'est du côté cuivré que vous les trouverez, ce coup-ci !



et vous aurez compris qu'assembler tout cela convenablement devient une affaire sérieuse. Aussi conseillons-nous à l'amateur peu expérimenté de s'entourer des conseils d'un « vieux routier » de la piste imprimée.

Il faut donc garnir les deux faces de la platine. Tous les organes de commande, potentiomètres, LED (y compris D1, le témoin de marche) et poussoirs seront implantés sur le côté cuivre, les autres pièces sur le côté isolant. On commence par ce dernier et en particulier, les ponts câblés. Un bon système, c'est de faire une copie de la figure 2 et de cocher au fur et à mesure ceux qui sont soudés pour n'en oublier aucun. Il y faut de l'attention, parce que certains d'entre eux se dissimulent avec malice, comme celui qui va de part et d'autre de C2, par exemple, celui situé près de IC1/IC2 ou le minuscule, entre K6 et P4. Il y en a d'autres qui doivent être pliés en oblique, comme ceux qui jouxtent IC21 et S8. On peut alors planter les autres composants, comme d'habitude les plus bas d'abord (résistances, diodes, supports de CI), puis les plus hauts, (condensateurs électrolytiques, régulateurs de tension, transformateur).

Les différents connecteurs peuvent également se fixer à la platine, K2, K3 et K5 groupés sur un côté, question de faciliter plus tard l'installation dans le boîtier.

Une fois le côté composants complété, on peut retourner la platine et s'attaquer à la tâche d'y monter, l'un après l'autre, les potentiomètres à glissière. Ils se fixent à l'aide de boulons M3, en intercalant des bagues, en matière plastique de préférence, de manière à laisser un vide de quelques millimètres au dessus de la platine, sinon le fer à souder ne pourra plus atteindre les cosses de raccordement. Pour la même raison, il y a intérêt à commencer le montage des onze potentiomètres à glissière par la droite, donc

P2, puis P3 et P4, ensuite P12, P11 et ainsi de suite.

Si vous voulez que les interrupteurs S1 à S10 restent accessibles, quand la platine sera insérée dans le coffret, il faut aussi les surélever quelque peu avant de les fixer. Sur notre prototype, nous avons utilisé quatre picots à souder sur la platine et des douilles correspondantes sur les contacts des interrupteurs. Mais ce n'est évidemment pas la seule méthode possible.

Les témoins à LED aussi devront se tenir « sur la pointe des pieds » pour être encore sur la photo après la mise en boîte, mais comme ils sont dotés, de naissance, de longues pattes, il n'y a rien à craindre pour eux.

À propos de photo, celles des **figures 3 et 4** mettent en scène une foule de détails qui vous permettront, nous l'espérons, de vous inspirer dans votre propre réalisation.

Mis à part le réglage de P1, dans l'alimentation, le pupitre de commande d'éclairages ne nécessite aucun ajustement. Il n'y a donc aucune raison pour qu'il ne fonctionne pas du premier coup et la tentation est grande de brancher un cordon secteur à K1, la soudure du dernier composant à peine refroidie, et de faire commencer le spectacle ! Il y a cependant deux raisons de rester sur ses gardes. D'abord, avec un si grand nombre de ponts à câbler et des composants répartis sur les deux faces, mieux vaut s'assurer que tout a été correctement installé et soudé. Et

puis, il faut se rappeler qu'une partie du circuit imprimé est soumise à la tension du secteur et que la moindre inattention mène à des conséquences fâcheuses. Prudence donc !

Dans le décor

Comment encadrer son chef d'œuvre, à ce stade, voilà une question à laquelle chacun a le droit de répondre comme il l'entend. N'empêche, l'ergonomie suggère qu'une façade inclinée comme le pupitre du professeur, jadis, serait la plus appropriée à ce genre d'appareil.

Pour notre prototype, nous avons eu le coup de cœur pour le boîtier 52 39 84-60 de Teko. Sa largeur de 310 mm et sa profondeur de 170 mm offrent à notre montage un logement princier, mais sans encombrement inutile. La platine pourra, par l'entremise d'entretoises, s'amarrer à la face oblique, après y avoir pratiqué les découpes nécessaires pour les potentiomètres, les poussoirs et les diodes électroluminescentes. C'est une opération qui demande évidemment du soin, puisque l'aspect final en dépendra.

Et puis il y a les questions de sécurité électrique, un interrupteur secteur irréprochable, des accès au secteur dont l'isolation est à toute épreuve, comme la liaison au connecteur K1, sans oublier les connexions à F1, les pistes entre F1 et K1, ainsi qu'à TR1, qui doivent rester à distance raisonnable de toute pièce métallique, surtout du boîtier. Enfin, au rayon sécurité, il y a cette plaque d'identification à coller à l'arrière du coffret, porteuse des mentions de tension d'alimentation et de dimension du fusible. Rien ne doit être négligé, dans le domaine des mesures de sécurité.

K2 est en mesure de relier les différentes lampes, spots, projecteurs et tout ce que vous pouvez imaginer. Certains de ces luminaires, disponibles dans le commerce, sont dotés d'origine d'un gradateur standardisé dans la gamme de commande de 0 à 10 V. Si ce n'est pas le cas, sur le matériel dont vous disposez, encore un peu de patience, votre revue préférée, Elektor évidemment, va vous proposer incessamment un projet dans ce sens.

(000162)

Cours « Microcontrôleurs », 2^{ème} partie, Elektor n°284, février 2002, page 20 et suivantes

Le listage 3 de la page 24 comporte une petite erreur qui n'a pas de conséquence sur le fonctionnement du programme :

La ligne loop doit être la suivante :

```
Loop djnz r1,loop ; 2 * 255
```

La seule différence est que la première boucle sera aléatoire, la variable r3 n'ayant pas été initialisée. (010208-III)

Télécommande par portable et SMS (I), Elektor n°284, février 2002, page 40 et suivantes

Le tableau 1 manque de précision. Comme il n'existe pas de broche 0, il faut lire broche 1 à broche 16 au lieu de broche 0 à broche 15. Ces broches sont reliées dans l'ordre aux ports P1 et P2. L'utilisateur a toute liberté de définir le mode, entrée ou sortie de chaque broche. (010087-I)

Télécommande par portable et SMS (II), Elektor n°285, mars 2002, page 32 et suivantes

La platine comporte une petite erreur. L'un des contacts du jack d'alimentation K8 se trouve relié par erreur à la masse. Il faudra donc, lorsque vous réalisez votre propre platine, veiller à éliminer, avant reproduction photographique, ces 4 petites liaisons visibles sur le plot situé en regard, côté « composants », du (c) de notre copyright. On pourra, si l'on a déjà gravé sa platine, enlever l'ensemble de l'îlot à l'aide d'un foret de 5 mm en veillant cependant à ne pas aller trop loin. Signalons que les platines disponibles auprès de Publitrone et les reproductions des dessins de platines proposées au téléchargement ont elles été corrigées. (010087-II)

Fourches caudines, Elektor n°285 mars 2002, page 8 et suivantes

Une première remarque pour rassurer tous nos lecteurs. Merci Mr Giroud d'avoir tiré la sonnette d'alarme. Le dessin de platine proposé ne comporte pas d'erreur. Les dessins de la figure 4 comportent chacun une erreur. Dans la figure du haut il faut intervertir les fonctions récepteur et émetteur, IC3 et IC4 sont partant les récepteurs et se trouvent en haut, les émetteurs, IC1 et IC2, en bas.

Dans la figure du bas le brochage du connecteur K1 a été présenté en miroir, la broche 1 devant se trouver en bas, la broche 5 en haut. (010206-I)

Analyseur de code IR, Elektor n°280, octobre 2001, page 12 et suivantes

Il existe une certaine incohérence entre le schéma et la liste des composants. En fait, il s'avère qu'ils sont tous deux améliorables. Il faut en fait lire, tant sur le schéma que dans la liste des composants :

R1 = 100 Ω

R2 = 220 Ω

Notons en outre qu'il est préférable d'utiliser, pour C1, un condensateur d'au moins 47 μ F plutôt que les 10 μ F proposés.

S'il devait se faire que votre analyseur ne réagisse pas, cela peut tenir au fait que IC2 n'arrive pas à forcer sa sortie au niveau bas. Cela tient au fait que tous les équivalents du récepteur IR IC2 du type SFH505 prévu à l'origine ne sont pas en mesure de drainer (sink) un courant d'importance égale. Il est préférable dans ce cas-là d'augmenter à 2k Ω voire 2k Ω 7, la valeur de R3, ce qui a cependant l'inconvénient de se traduire par une diminution de la luminosité de la LED. (010029-I)

Télécommande PGM miniature (I), Elektor n°280 octobre 2001, page 70 et suivantes

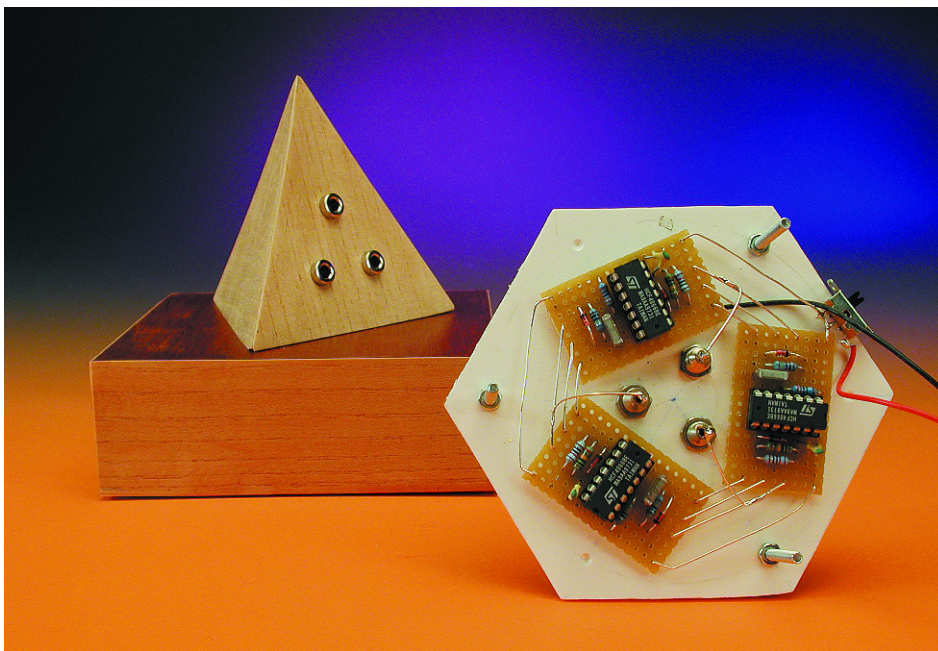
Le FET à canal P du récepteur a été monté de façon erronée ce qui se traduit par la prise de sa diode interne en parallèle avec la diode D2. Pour remédier à cette situation qui met le moteur en court-circuit il faudra utiliser pour T4 un FET à canal N, un SUP75N03 par exemple, dont on intervertira, lors de la mise en place, les contacts de source et de drain. La source se trouvera partant reliée à la ligne positive de la tension d'alimentation.

Cette interversion est facile à réaliser lors de la mise en place du composant sur la platine. Il suffira de modifier de 180 ° l'orientation du FET à canal N utilisé pour T4 en remplacement du FET à canal P prévu à l'origine, ce qui se traduira automatiquement par une interversion des contacts de source et de drain. (010205-II)

Escher-Tronic

Impossible n'est pas français –
contradiction des lois de Kirchhoff!

Nombre d'idées vraiment révolutionnaires doivent souvent au hasard le fait qu'elles aient vu le jour. Ceci est également vrai de la découverte présentée ici, découverte qui ébranle les fondements de l'électrotechnique.



Les premières nouvelles concernant ce raz-de-marée arrivèrent au laboratoire d'Elektor le 1.04.2001, par le biais d'un informateur secret et ce en provenance d'une Haute École allemande de renom. C'est là que furent observés les phénomènes auxquels nous allons nous intéresser dans le cadre de cet article. En collaboration avec la dite Haute-École nous sommes même arrivés à réaliser un montage que tout lecteur de ce magazine pourra reproduire en vue de vérifier les résultats énoncés. Cependant, les résultats sont tellement surprenants et inexplicables que nos honorables correspondants de la Haute École ont préféré, au premier abord, ne pas être nommés pour ne pas risquer les foudres des scientifiques de l'Establishment. La

rédaction d'Elektor se réjouit d'autant plus de pouvoir présenter dans ce magazine réputé pour son sérieux les résultats des expériences innovatrices de ces intrépides scientifiques de manière à les mettre à l'épreuve dans un cadre plus large.

La loi des mailles de Kirchhoff

Les lois de Kirchhoff (puis ensuite la loi des noeuds) sont l'une des premières choses qu'apprend tout électrotechnicien en devenant. En 3 mots, les lois disent que la somme des tensions dans un réseau maillé (ou à

noeuds) est nulle. C'est ce qu'illustre le schéma de la **figure 1**.

La maille M est constituée ici des 2 résistances R1 et R2 et du condensateur C. On procède à la mesure des tensions présentes entre les noeuds A, B et C. Les tensions sont dotées de flèches vectorielles. Les tensions ayant une flèche respectant le sens de la maille reçoivent, dans la somme, un signe positif, les tensions dont la flèche pointe dans le sens contraire reçoivent un signe négatif. Dans notre exemple cela se traduit tout simplement par l'équation :

$$U_1 + U_2 + U_3 = 0$$

Sur de l'électronique mise en pratique, ces lois de maille sont souvent utilisées de la façon suivante : la tension existant entre les noeuds A et C est égale à la somme de celles qui existent entre A et B et B et C. Prenons un exemple chiffré :

si $U_{AB} = 3 \text{ V}$, c'est-à-dire que A est positif par rapport à B et que $U_{BC} = 5 \text{ V}$, B est partant positif par rapport à C, cela entraîne que A est positif par rapport à C et que $U_{AC} = U_{AB} + U_{BC} = 3 \text{ V} + 5 \text{ V} = 8 \text{ V}$. Les choses ne sont pas plus compliquées que cela jusqu'au jour mémorable où tout se vit chamboulé.

Escher-Tronic

Cette loi des mailles vaut bien entendu également dans le cas de phénomènes non-électriques. Si vous faites une promenade dans les

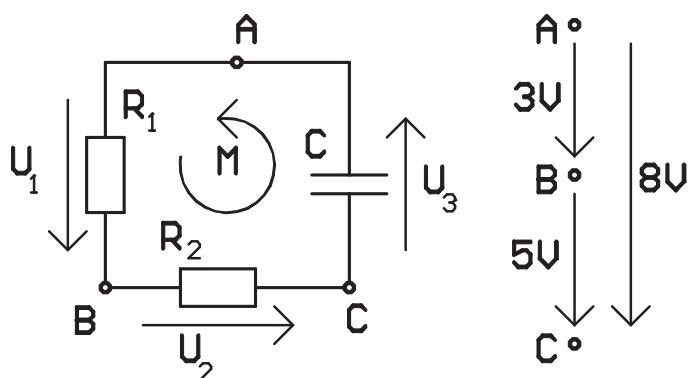


Figure 1. La loi des mailles de Kirchhoff.

Vosges et que vous revenez au point A d'où vous êtes parti, et que vous commencez par gagner 300 m d'altitude et qu'ensuite vous gagnez encore 500 m, il vous faudra partant

redescendre 800 m. Il s'agit d'un fait indiscutable établi depuis la nuit de temps. Mais stop : depuis quelque temps déjà il existe une preuve de l'inverse, qu'illustre le dessin de la

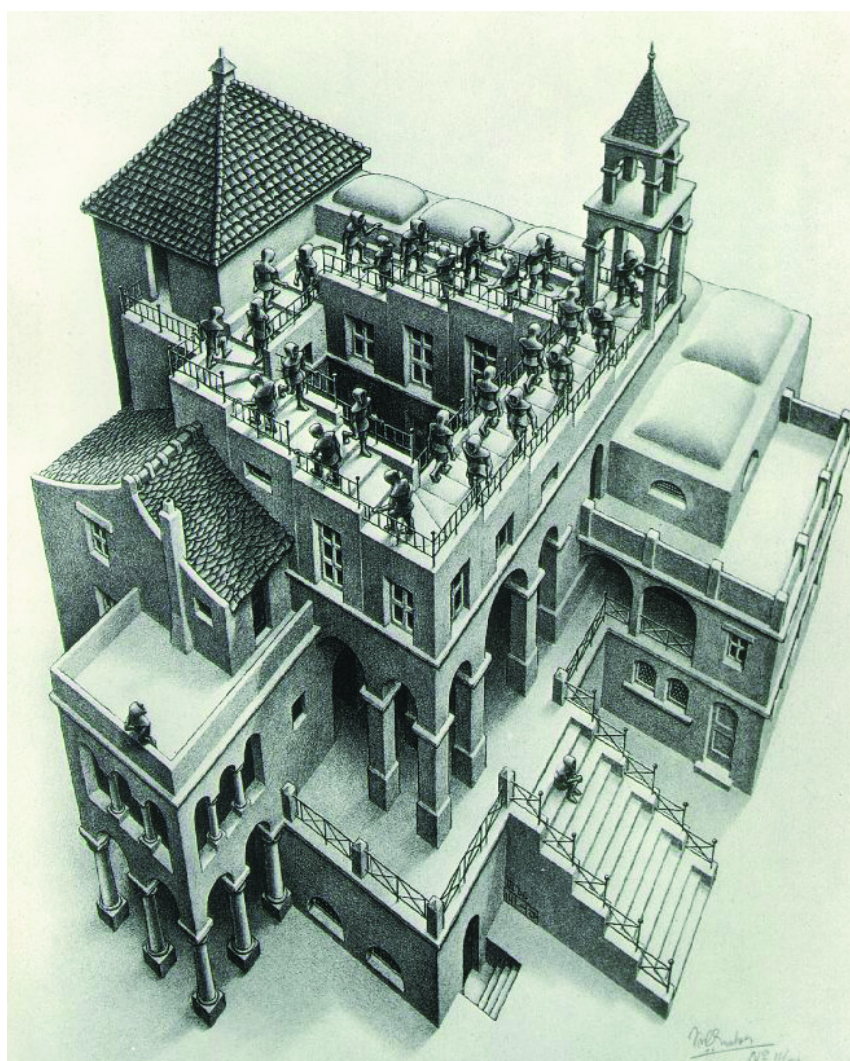


Figure 2. M.C.Escher: montée et descente.

figure 2. Un amateur de croquis néerlandais du nom de M. C. Escher (1898-1972) a montré avec brio qu'il existait des escaliers qui ne font que monter. Pourquoi ne rencontrerait-on pas la même situation en électrotechnique ? Un montage qui fournit en circuit une tension en croissance constante mérite bien la dénomination de Escher-Tronic. On peut s'attendre, avec un tel montage, à des effets intéressants et grâce au flair des chercheurs de la Haute École mentionnée plus haut, nous avons réussi à créer une telle réalisation que nous publions dans le cadre de cet article. Nous allons commencer par la description de cette réalisation avant d'aborder les expériences que permet notre Escher-Tronic. Ces expériences contredisent d'une façon irréfutable et impressionnante les lois de Kirchhoff citées plus haut. Si la réalisation vous effraie par sa complexité, qui n'a cependant rien pour effrayer un lecteur assidu de ce magazine, vous pourrez commencer par lire les résultats des expérimentations que permet ce montage. Nous ne serions guère étonnés que vous décidiez ensuite de réaliser vous aussi cet incroyable montage.

La réalisation

Si l'on veut contredire la loi des mailles de Kirchhoff sur le plan de travail d'un laboratoire, et ce avec des moyens communément disponibles, cela requiert inévitablement une approche d'une certaine complexité. L'attaque contre quelque chose que l'on a toujours considéré comme la vérité ressemble à expédition dans l'occulte et le paranormal. Il est prouvé, dans le paranormal que la plupart des phénomènes (psychokinésie, télépathie...) disparaissent lorsqu'on les examine scrupuleusement en s'aidant de méthodes scientifiques. Il nous faut reconnaître que nous sommes confrontés à la même situation dans le cas du présent montage. Dès que l'on prend la peine de dessiner un schéma précis la tentative de contredire la loi des mailles tourne au vinaigre et l'expérience avorte. Ceci explique que, contrairement à nos habitudes chez ELEKTOR, nous ne pouvons pas vous proposer de schéma. Tout ce que nous pouvons vous offrir est un mode de construction de notre Escher-Tronic. Il faudra, respecter les informations (celles de la liste des composants également) tant à la lettre que dans l'ordre indiqué pour pouvoir reproduire les expériences.

Triangle magique

Notre montage comporte 3 circuits identiques montés circulairement sur un triangle équilatéral (cf. photo en début d'article). Ce mode



Figure 3. Triangle sur timbre-poste.

de construction n'est guère utilisé dans le monde de l'électrotechnique classique en raison des risques importants de court-circuit qu'elle favorise. Il nous faut bien entendu nous lancer dans l'Inconnu de la technique de la connexion. Nous pouvons, comme l'illustre la **figure 3**, nous référer au grand Art. Tout le monde sait que les triangles cachent toujours l'une ou l'autre surprise. Il nous vient à l'idée les concepts de la trinité et du triangle des Bermudes. Avec notre montage, la loi des mailles disparaît, pour ainsi dire dans un triangle des Bermudes électronique...

Circuit partiel

Nous sommes ici en présence de 3 circuits partiels identiques. Ils ne requièrent que peu de composants et pourront partant prendre place sur des morceaux de platine d'expérimentation à pastilles. On pourra bien entendu également réaliser un petit circuit imprimé. Chaque circuit partiel comporte 4 connexions à gauche et autant à droite, points utilisés pour la connexion en anneau

des 3 platines. Nous vous proposons en **figure 4** un exemple de dessin des pistes et de sérigraphie de l'implantation des composants.

Ce montage est à faire en 3 exemplaires. Ces 3 platines seront, comme l'illustre la **figure 5**, interconnectées en anneau. Les 3 embases A, B et C sont, comme le montre cette figure, reliées aux interconnexions S-S. On connectera une pile compacte de 9 V, en respectant sa polarité, entre les points « + » et « - ». La ligne positive de l'alimentation (+) comporte un interrupteur permettant de couper l'alimentation. On voit, en figure 5, le plan de câblage des platines ceci vu côté « composants ». Les douilles-banane A, B et C sont vissées dans une face avant et vues ici de l'arrière vu que c'est aussi à cet endroit que se fait la soudure (cf. la photo de l'un de nos prototypes).

Il faudra veiller tout particulièrement, lors du montage des platines, à ce que l'ensemble réalisé soit parfaitement symétrique. Seul un assemblage optimal dans un triangle équilatéral donne les résultats optimum. Il est important également que les poids des circuits partiels soient aussi proches l'un de l'autre que possible, ce qui explique qu'il faudra prendre soin, en particulier, à réali-

ser des points de soudure de même taille partout !

Une fois que l'on en aura terminé avec la réalisation, il sera temps de passer aux expériences on ne peut plus étonnantes.

Résultats de mesure incroyables

Nous pouvons maintenant nous attacher à contredire, sur notre plan de travail, en nous aidant de méthodes scientifiques, la loi des mailles. Pour ce faire, nous allons appliquer la tension au Escher-Tronic en fermant l'interrupteur et, armé d'un multimètre que l'on aura placé en fonction voltmètre, en calibre 10 ou 20 V, (note : dans la Haute École en question, cette mesure a été effectuée à l'aide d'un multimètre à aiguille d'une marque de matériaux de construction en raison des restrictions financières qui empêchent l'achat d'instruments de meilleure qualité. La reproduction des mesures effectuée dans un laboratoire de recherche renommé tel que le laboratoire d'Elektor avec des instruments de précision numérique a d'ailleurs confirmé les résultats obtenus).

Nous allons commencer par mesurer (cf. **figure 6a**) la tension entre les points A et B. Le résultat est que le

Liste des composants

Résistances :

(toutes 1/8 W, film métal tolérance de 1%)

R1 = 1 MΩ

R2 = 330 Ω

R3 = 100 kΩ

R4 = 10 kΩ

Condensateurs :

C1 = 5nF6, (50 V, MKS ou MKT)

C2 = 1 nF (50 V céramique)

Semi-conducteurs :

D1, D2 = 1N4148 ou 1N914

IC1 = HCF4066B

(STMicroelectronics)

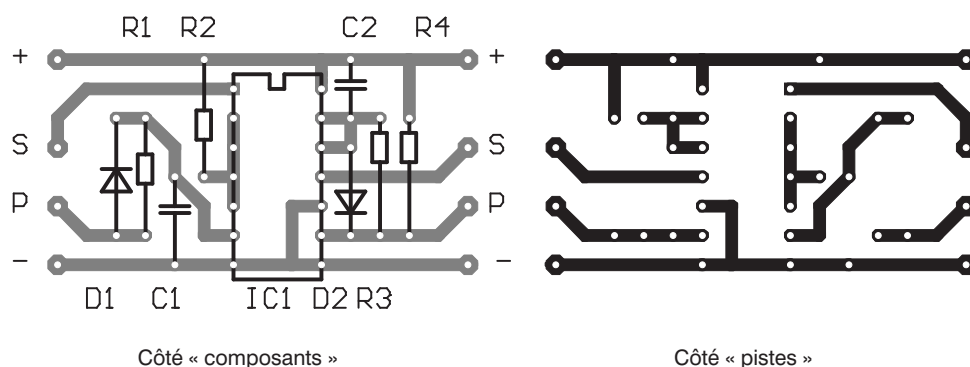


Figure 4. Dessin des pistes simple et sa sérigraphie de l'implantation des composants.

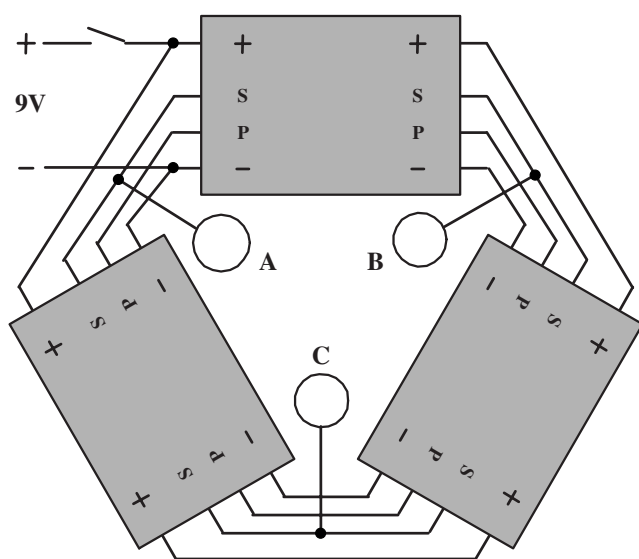


Figure 5. L'anneau magique de notre Escher-Tronic.

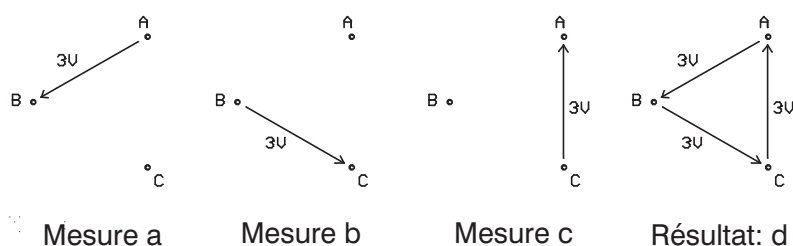
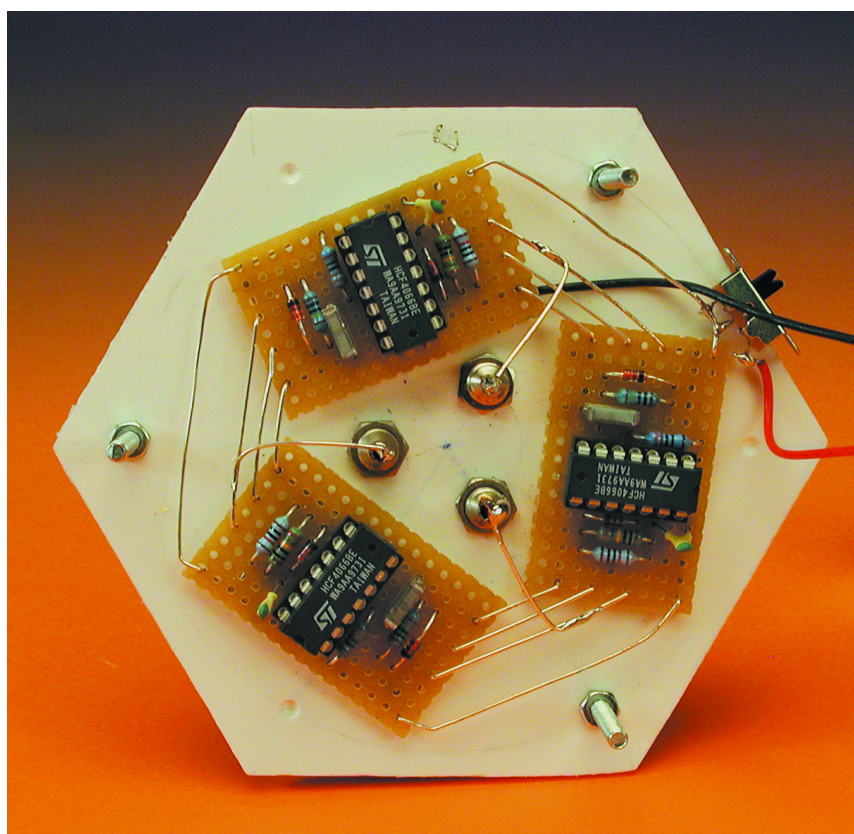


Figure 6. Les résultats de mesure contredisent la loi des mailles.

point A est positif par rapport au point B et que la tension entre A et B est de l'ordre de 3 volts.

Invertissons, à titre de vérification, les sondes de mesure et nous verrons que l'instrument affiche -3 volts, ce qui confirme que nous avons bien fait une mesure en courant continu correcte.

Mesurons maintenant la tension entre les points B et C. Le résultat de cette opération nous permet de constater que le point B est positif par rapport au point C et que la tension entre B et C est elle aussi de quelque 3 volts (cf. **figure 6b**). La loi des mailles nous permet de prédire la tension entre les points A et C : si Kirchhoff ne s'est pas trompé, le point A doit être positif et cela de 6 volts par rapport au point C !

Pour nous en assurer nous allons mesurer la tension entre les points A et C. À la surprise générale, le point C s'avère positif par rapport à A et cela d'une tension de 3 volts (cf. **figure 6c**). Cela prouve à l'évidence que la loi des mailles est fautive ! Grâce à notre montage, nous avons réussi à établir un exemple contredisant la loi des mailles. Notre électronique reproduit la situation représentée en **figure 6d**, situation dans laquelle la somme des tensions des mailles est de 9 volts et non pas de 0 V. Ce résultat est tellement incroyable que nous ne pouvons pas l'accepter sans autre forme de procès, raison pour laquelle nous allons procéder à d'autres tests.

(Une) évidence lumineuse

Les électroniciens invétérés commenceront sans doute à se poser la question de savoir si nos instruments de mesure sont bien en bon état de fonctionnement. Peut-être s'agit-il, avec le Escher-Tronic, de sources présentant une impédance tellement élevée que le moindre courant de fuite entraîne des mesures erronées. Nous allons partant commencer par vérifier si les douilles du Escher-Tronic fournissent bien du courant. Pour ce faire, nous allons, comme l'illustre la **figure 7a**, connecter une LED entre les points A et B du circuit et que constate-t-on, comme nous pouvions nous y attendre, la LED s'allume vu qu'elle se trouve connectée dans le sens direct.

À titre de vérification nous inversons la polarité de la LED et nous constatons qu'elle ne s'allume pas. Cela prouve à l'évidence que les points A et B constituent les pôles d'une source de tension continue. Nous pouvons effectuer le même test avec la LED sur les paires de points B et C et C et A (figures 7b et 7c). À chaque fois la LED s'allume lorsque l'anode se trouve reliée à la douille identifiée auparavant comme étant positive. Cela

confirme les mesures effectuées précédemment avec notre instrument de mesure. On en déduit qu'indubitablement, la loi des mailles est mal en point.

Triangle lumineux

De manière à prouver que notre Escher-Tronic a effectivement réalisé la situation représentée en figure 6d nous allons connecter simultanément les 3 LED selon le plan de câblage de la **figure 7d** et que se passe-t-il ? Toutes les diodes électroluminescentes s'allument toutes en même temps, une preuve irréfutable de l'incroyable : la loi des mailles de Kirchhoff est réfutée !

Explications

Tous ces résultats sont purement et simplement incroyables et Elektor ne serait pas Elektor et les Hautes-Écoles allemandes des Hautes-Écoles si nous n'étions pas en mesure de fournir l'une ou l'autre explication. Il n'en reste pas moins il en est toujours de même lors de la découverte d'une réalité nouvelle aux conséquences révolutionnaires : il est difficile au début de savoir où chercher la vérité. La possibilité d'explication la plus simple est que notre montage comporte en fait 3 **convertisseurs à pompe de charge** qui simulent 3 sources de tension fournissant chacune une tension de 3 volts, sources interconnectées comme l'illustre la **figure 8a**. Ceci expliquerait parfaitement les résultats de mesure; nous pourrions partant réaliser ce montage en utilisant 3 piles de 3 volts distinctes. Attention cependant : cette explication ne tient pas debout (il ne saurait partant être question de l'essayer !). La prise en anneau de nos 3 piles se traduit par leur décharge rapide et cela avec un échauffement important en raison de leur résistance interne *cf. **figure 8b**). En même temps, on ne trouve pas de tensions aux douilles vu que la tension de chacune de pile tombe aux bornes de leur résistance interne. Loin d'aider, une résistance interne plus faible ne fait qu'augmenter le danger de la situation.

Une autre idée géniale (?) serait de supposer que notre électronique génère 3 tensions continues impulsionnelles déphasées selon le **principe du courant triphasé**, situation illustrée par le schéma de la **figure 9a** (le chronogramme des signaux obtenus étant celui représenté en **figure 9b**). Cette explication est boiteuse elle aussi : en présence de telles tensions, les instruments de mesure en CC (Courant Continu) affichent un beau « 0 » pour la simple et bonne raison que la tension entre chacune des paires de points A, B et C, est nulle. D'autre part, les LED s'allumeraient quelle que soit leur polarité de connexion, ce qui n'est pas, nous l'avons constaté plus haut,

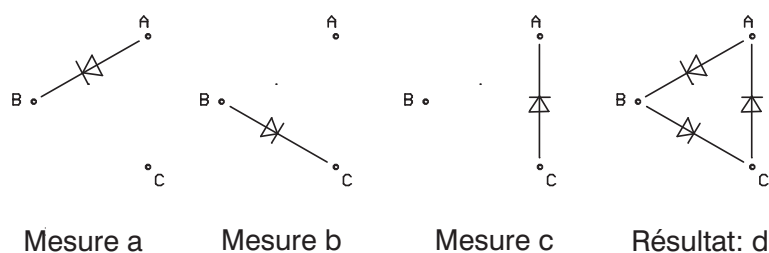


Figure 7. Toutes les diodes électrochimiques s'allument.

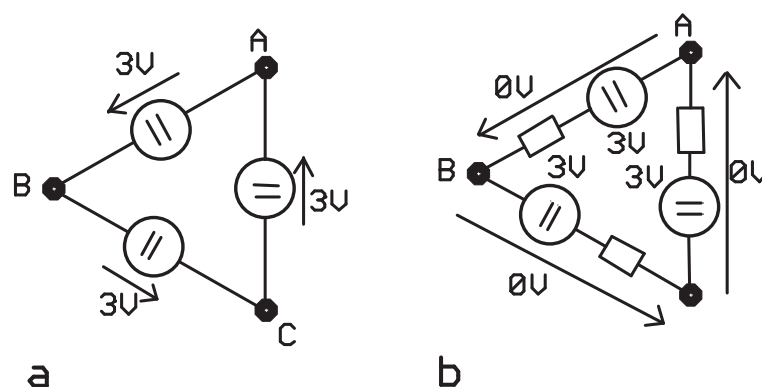


Figure 8. Sources de tension en anneau !

pas le cas de notre Escher-Tronic. Il n'est pas non plus possible d'expliquer le comportement de notre Escher-Tronic à partir de tensions de formes différentes; il nous faut partant continuer nos recherches pour trouver une explication plausible. Nous n'avons, jusqu'à présent, parlé que de la loi des mailles de Kirchhoff, qui dérive naturellement de la théorie des réseaux. Son principe constitue une **égalité de Maxwell** dont l'équation répondant à la formule suivante :

$$\oint \vec{E} \cdot d\vec{s} = \frac{d}{dt} \int_A \vec{B} \cdot d\vec{A}$$

Le membre gauche représente la tension de révolution produite par le champ électrique circulant dans une maille. Le membre droit n'a rien d'un zéro ! Nous venons de découvrir le coupable : la tension de révolution n'est pas nulle mais égale au moment d'évanouissement magnétique (*fading*) capté par la maille.

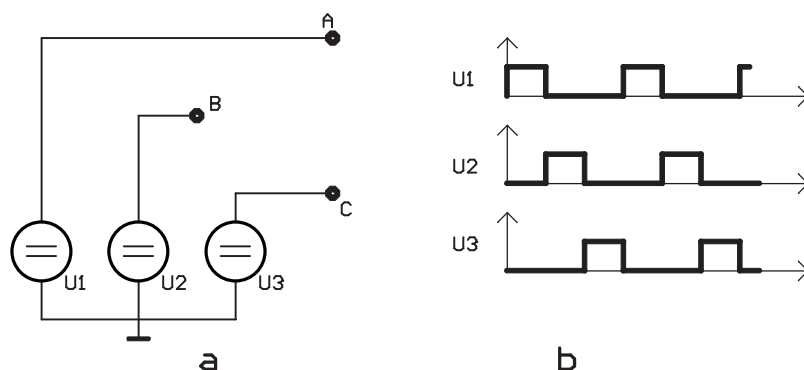


Figure 9. Anneau de courant triphasé.

Tout ce dont nous avons besoin partant, pour expliquer les effets de notre Escher-Tronic est un champ magnétique variable dans le temps suffisamment grand. Il n'est pas difficile de le trouver ce champ magnétique puisque nous pouvons utiliser le champ magnétique terrestre (qui varie légèrement dans le temps) et d'en amplifier électroniquement l'effet. Nous avons enfin la solution !! Notre montage est un GMFA (*Geo-Magnetic-Fading-Amplifier* = amplificateur à fading géo-magnétique). Il reste cependant un petit problème : lorsque nous mettons le Escher-Tronic sur la tête l'orientation des mailles par rapport au champ magnétique terrestre change ce qui partant devrait se traduire par une inversion de signe de toutes les tensions. (Note : permettez-nous d'attirer votre attention sur les dangers présentés par les LED non fixées correctement et les instruments de mesure qui tombent lorsque l'on tente des expériences la tête en bas !). Le résultat de cette acrobatie est aussi révélateur que déprimant : les tensions ne voient pas leur signe changer, ce qui prouve que notre nouvelle explication, elle non plus, ne tient pas debout. Même un pilier de la science comme Maxwell ne nous sort pas du pétrin.

Dernière tentative d'explication

Comme le sait tout homme cultivé, la mécanique quantique et partant **l'électro-dynamique quantique** permet des exceptions aux équations macroscopiques de Maxwell, si tant est que ces exceptions soient plus faibles et plus brèves que ce que requiert la **relation de flou** de Heisenberg. Il semblerait donc que notre montage soit en mesure de forcer les électrons brièvement dans le sens inverse de la loi des mailles, d'amplifier ce phénomène et de le visualiser. Ceci nous semble être une explication tenant la route. Il n'est pas évident non plus de démêler tout cet embrouillamini. Il est peut-être possible de voir les choses sous un autre angle :

Si l'on a de bonnes notions d'électricité on voit immédiatement que notre montage repose en fait principalement sur l'existence de charges

positives (manque d'électrons ou protons présents) et de charges négatives (électrons). Cela se traduit immédiatement par la constitution d'un pôle positif (+) et d'un pôle négatif (-), c'est-à-dire, comme nous pourrions le dire autrement un couple Yin et Yang. Et c'est précisément cette simple paire qui ne nous permet pas d'expliquer le fonctionnement de notre Escher-Tronic. Le monde a besoin d'un nouveau paradigme, que l'on connaît d'ailleurs

depuis belle lurette dans la physique des particules. Notre bouée de sauvetage sont les **quarks** dont il existe en fait 3 sortes qui tombent à pic pour le comportement « triangulaire » de notre montage. Vu d'autre part qu'il est possible de créer des protons et des électrons à partir de quarks, ceci explique que nos instruments de mesure électriques classiques soient en mesure de visualiser un phénomène quelconque. Il devrait être possible d'expliquer les choses sous cet angle. Nous lançons un appel aux physiciens.

(011006)

Ndlr : comme le problème paraît loin d'être résolu, nous sommes ouverts à toutes les explications que nous serions heureux de recevoir par E-mail à l'adresse redaction@elektor.presse.fr ou par fax (cf. numéro dans l'ours). Les explications les meilleures et les plus plausibles seront publiées en ligne sur notre site www.elektor.fr (notez la disparition de l'intercalaire .presse dans cette adresse) avec le nom de leur auteur et cela jusqu'à ce que nous ayons enfin une réponse à notre énigme et que le groupe de travail chargé du projet Escher-Tronic chez Elektor ait enfin une réponse satisfaisante. Cette dernière explication ultime, si tant est qu'elle voit le jour, sera publiée dans le numéro d'Elektor suivant sa présentation. Sinon, nous clôturerons cette action définitivement et une fois pour toutes le 01.04.2003 et mettrons le Escher-Tronic dans la collection de PEI (Phénomènes Électriques Inexpliqués) que les membres d'une Haute-École allemande renommée restée anonyme envisagent de constituer en vue de symposium à tenir dans le futur.

