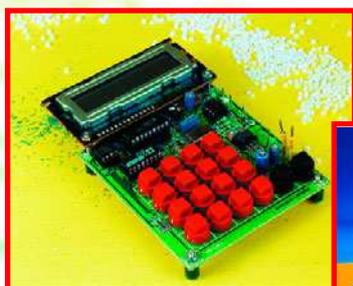


+ PC-PLUS :

- réactualisez votre PC pour 3 fois rien •
- clavier hexadécimal pour PC •
- programmation structurée en assembleur avec SXP •



régulateur P.I.D. numérique



**le pilotage
de moteurs
pas à pas**

Cours BASIC-Stamp *créez votre robot de toutes pièces*

le bus CAN
transmission
de données
pour l'auto,
la domotique
& l'industrie



INFORMATIONS

18 le bus CAN (1)

La mise en pratique de la communication de données intelligente décentralisée. CAN, Profibus, LON, ASI, Interbus-S, Bitbus, FIP, EIB, eBUS, il n'est plus possible d'endiguer l'avancée spectaculaire des systèmes de communication en réseau, les bus industriels.

34 potentiomètre numérique

la puce Xicor, X9440/9448, ne manque pas d'R.

64 moteurs pas à pas

Pour tout savoir sur la construction, le fonctionnement et la commande des moteurs PAP.

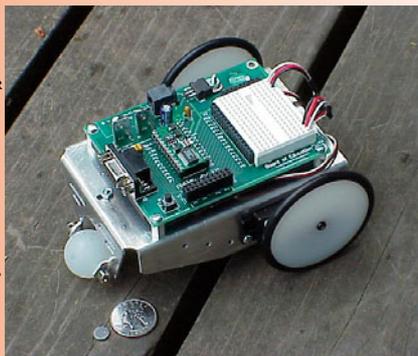


RÉALISATIONS

14 cours de programmation du BASIC Stamp II

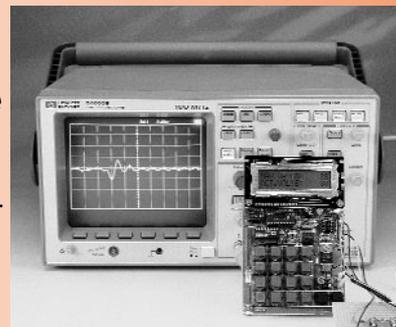
Tère partie : introduction & le matériel du BoE

Voici le premier article d'une série décrivant l'assemblage d'un robot rustique baptisé BoE-Bot, qui nous servira vous initier à la programmation du BASIC Stamp.



28 régulateur P.I.D. numérique

à PIC dans le système réglé Comment réaliser un régulateur P.I.D. numérique.



54 enregistreur de température DS1615

kit d'évaluation de la puce de Dallas Semiconductor Le DS1615 est un enregistreur de température intégré qui peut aussi bien servir d'enregistreur de données (température en fonction du temps) que créer des histogrammes (nombre de valeurs de température mesurées) entre -40 et +85 °C.

24 EDiTS Pro

Booster & Cie Interface simple pour utiliser un booster Märklin avec un système EDiTS Pro.

60 testeur de sonde lambda

comment contrôler l'exactitude du mélange Cet instrument vous permettra de vérifier le bon fonctionnement du pot catalytique de votre voiture.

RUBRIQUES

- 23 Elekture
- 53 électronique en ligne : le transfo de Tesla
- 63 tort d'Elektor
- 67 Infocartes : DS1615
- 69 Actualités Electroniques
- 73 Marché
- 76 Petites Annonces Gratuites Elektor
- 82 Publitronic Service
- 84 Avant-Première

CE MOIS-CI DANS PC-PLUS:

- précompilateur Sxp
- clavier hexadécimal pour PC
- PC-Upgrade, réactualisez votre vieille carte-mère dépassée
- recyclage de lecteur de CD-ROM

Langage de programmation de haut niveau ou langage assembleur ? Dans la perspective de cycles de développement ne cessant de se raccourcir, l'ingénieur de développement de logiciel a tendance, s'il n'y est pas tout simplement forcé, à opter, de plus en plus, et ce même pour les développements ayant trait aux microcontrôleurs, pour des langages de programmation de haut niveau, le langage C par exemple. Ce sont bien souvent aussi les environnements, la documentation des programmes et leur tenue à jour, voire les délais à respecter qui l'y incitent. Cette approche n'est pas envisageable lorsqu'il ne dispose que de ressources de mémoire trop justes ou qu'il lui faut créer des applications à la chronologie critique. Un assembleur doté d'instructions orientées vers une programmation structurée et la permettant offre, pour le développeur, les avantages des 2 langages tout en garantissant une vitesse de traitement élevée et une utilisation optimale de la mémoire.

Dipl.-Ing. Eberhard Haug

précompilateur SXp

programmation en assembleur structurée pour le Scenix™ SX



En 1983 déjà, Rolf-Dieter Klein avait présenté son STRUKTA, un précompilateur tournant sous CP/M, qui compilait des instructions pour programmation en assembleur structurée pour en faire du code-source assembleur 8080/Z80. D'après le même principe que STRUKTA, l'auteur a développé le précompilateur SXp qui supporte la famille SX du fabricant de semi-conducteurs américain Scenix™. Il est possible, de par la présence d'instructions pour la programmation structurée au niveau des interrogations de valeurs, de sauts et autres boucles, de combiner la

vitesse de traitement et l'utilisation optimale offertes par le langage assembleur d'une part à la structure claire et facile à mettre en oeuvre que connaissent les langages de haut niveau de l'autre.

La programmation structurée

Il nous faut commencer par élucider une confusion typique au niveau des notions : le terme d'instructions de commande pour la programmation structurée ne désigne pas l'assemblage conditionnel (*conditional assembly*), technique que supporte la quasi-totalité

des assembleurs. Il arrive souvent que même les spécialistes ne soient pas conscients de la différence subtile mais importante (« IF-ELSE-ENDIF, mais cela mon assembleur sait aussi le faire... »). Les instructions de commande telles que les propose SXp permettent une programmation structurée en langage assembleur SX ce qui présente les avantages suivants :

- développement et débogage de programmes plus simples;
- lisibilité et documentation du programme très sensiblement améliorées;
- cycles de développement de programme sensiblement réduits et partant augmentation importante de la productivité;
- fiabilité et tenue à jour des programmes améliorées;
- plaisir accru lors de l'écriture de programmes pour le SX.

SXp met à la disposition du programmeur des instructions pour les situations d'interrogation de données, de sauts et de boucles, situations que l'on rencontre dans tout programme, mais qui sont, lorsque l'on programme en assembleur, le plus souvent complexes et sources d'erreur (« le carry est-il positionné, oui ou non ? »).

Les instructions SXp n'influent que très peu, si tant est qu'elles aient une quel-

conque influence à ce niveau, sur l'efficacité du code du programme. Ceci veut dire que, tant la vitesse de traitement atteinte que l'utilisation optimale de la mémoire réalisées sont pratiquement celles que permet une programmation en assembleur pur. L'utilisation de SXp permet de conserver tous les avantages de la programmation en assembleur vu que toutes les instructions non SXp sont (par définition), des instructions assembleur parfaitement normales. C'est bien là que se trouve l'astuce !

À qui est destiné SXp

SXp –qui tourne sous Windows® 95/98/NT– est un précompilateur qui traduit des instructions pour une programmation structurée (dites instructions SXp) en langage source assembleur pour la famille SX de Scenix™.

Les instructions assembleur SX proprement dites, auxquelles ne touche pas le précompilateur, viennent s'intercaler entre les instructions SXp.

On peut bien entendu intercaler, pour la structuration du programme, autant d'instructions SXp additionnelles que l'on voudra. Le code généré par SXp (seules sont utilisées les instructions de base du SX à 1 mot) doit ensuite, à l'aide d'un assembleur compatible Scenix™, être converti en code-objet SX définitif.

Le programme source en assembleur produit par SXp (le fichier de sortie) peut être utilisé pour la simulation et le débogage (à l'aide, par exemple, du SX-KEY™ de Parallax). Il est possible, optionnellement, en vue de faciliter le débogage en particulier, de conserver les instructions SXp sous la forme de commentaires dans ledit fichier.

Autre option : la possibilité de tenir compte de l'extension de transfert du SX (DEVICE CARRYX), une spécificité fort utile du SX, qui ne va cependant pas sans quelques pièges. Le SX ne connaît pas en effet d'instructions distinctes pour ni l'addition ni la soustraction avec respectivement prise en compte ou non de la retenue.

L'une des caractéristiques importantes de SXp est qu'il se limite à l'utilisation du registre de travail W et, le cas échéant, du registre de boucle spécifié. Il ne se sert sinon d'aucun autre registre SX et partant ne les modifie pas (il est vrai que le compteur ordinal et, le cas échéant, le registre d'état sont modifiés eux).

SXp ne se sert pas non plus de la pile SX. Ceci signifie que le développeur SX garde le plein contrôle de tous les registres SX.

En fonction du type de programmation, SXp devrait, comparé à une pro-

grammation 100% assembleur, donner une efficacité de code identique –tout en garantissant une lisibilité et une documentation sensiblement meilleures, un débogage plus aisé et un plaisir de programmation accru.

Il est possible, en fonction de la capacité de mémoire de l'ordinateur, sur lequel tourne SXp, d'imbriquer l'une dans l'autre autant de structures SXp que l'on voudra. Comme, lors des sauts de retour depuis les structures en question, SXp n'utilise pas la pile SX, qui, avec ses 8 niveaux est relativement petite, il devient possible, si le besoin s'en fait sentir, de sauter à n'importe quel endroit du programme SXp, voire même dans d'autres structures SXp. À ce niveau, le langage STRUKA exigeait d'énormes précautions, vu que ledit précompilateur effectuait le saut de retour en provenance de structures par l'intermédiaire, par exemple, de la pile.

Ce que le SXp ne sait pas faire

SXp ne génère pas, à partir du programme-source SXp, directement du code-objet SX. Le SXp ne connaissant pas d'adresses de programme absolues, il est dans l'impossibilité, entre autres, de détecter des sorties de page (la plage de mémoire adressable du SX est subdivisée en « pages »). Il faudra, dans ce cas-là, utiliser l'une des options de mode de page que connaît SXp.

De plus, SXp n'est pas en mesure de vérifier si les symboles et données définis et utilisés se trouvent dans les limites admissibles. C'est là la tâche de l'assembleur utilisé en aval.

Instructions SXp

Le **tableau 1** donne les instructions que connaît la version 0.9.x de SXp. On trouvera, à l'adresse Internet www.scantec.de, une description plus étoffée (en anglais et en allemand) de toutes ces instructions.

Tableau 1. Instructions SXp (SXp-Version 0.9.x)

IF	<i>Condition</i> [THEN]	; première interrogation
ELSEIF	<i>Suited'instructions</i> <i>Condition</i> [THEN]	; interrogation(s) suivante(s), multiplicité optionnelle
ELSE	<i>Suited'instructions</i>	; dernier aigillage, optionnel
ENDI	<i>Suited'instructions</i>	; fin de la structure IF
LOOP	<i>Suited'instructions</i>	; boucle sans fin ; sortie de la boucle LOOP à l'aide uniquement de ; EXIT, EXITIF, JMP, RET, Interrupt ou SX-Reset
ENDL		; fin de la boucle LOOP
WHILE	<i>Condition</i>	; interrogation de début de boucle
ENDW	<i>Suited'instructions</i>	; fin de la boucle WHILE
REPEAT		; début de la boucle
UNTIL	<i>Suited'instructions</i> <i>Condition</i>	; interrogation en fin de boucle
FOR	<i>Attribunderegistre</i> <i>Suited'instructions</i>	; boucle avec initialisation de registre
ENDF		; fin de la boucle FOR
EXIT		; Sortie d'un niveau de la structure
EXITIF	<i>Condition</i>	; Sortie conditionnelle d'un niveau de la structure
EXITIF	<i>Condition TO Adresse</i>	; Saut conditionnel d'un niveau de la structure

Listage 1 du programme SXP

```

; Racine carrée, Version SXP: Eberhard Haug, 27.06.1999
; Résultat 8 bits INT(SQRT(ZAHL)), d'où précision de -0.99

; utilise CARRYX pour arithmétique sur 16 bits!

; device pins28,pages4,banks8,turbo,stackx,optiox,carryx,oscxt
; device SX28L,turbo,stackx_optiox,carryx,oscxt2 ; silicium A

id 'ROOT'

reset start ; -adresse Reset de début

freq 1_000_000 ; nécessite <10 s pour toutes les 64K racines possibles
; freq 5_000_000 ; nécessite ±2 s pour toutes les 64K racines possibles
; freq 50_000_000 ; nécessite ±0,2 s pour toutes les 64K racines possibles
; en moyenne de l'ordre de 3 µs par SQRT @ 50 MHz

; Variable

org $10 ; tous les registres nécessaires se trouvent dans une banque

input1 ds 1
input2 ds 1
root_mask ds 1

lbyte ds 1 ; compteur de boucle pour MSByte
mbyte ds 1 ; compteur de boucle pour LSByte

;*****
; org 0 ; Le programme débute à l'adresse $000 (fixe)
start

; les 2 boucles FOR suivantes servent au calcul de la durée d'exécution

FOR mbyte = #256 ; 256 ...
FOR lbyte = #256 ; * 256 passages, calcule toutes les 64K racines possibles
mov w,mbyte
mov input2,w ; l'input2 ayant changé
mov w,lbyte
mov input1,w ; l'input1 ayant changé
call sq_root ; W = INT(SQRT(input2_1))
ENDF
ENDF

LOOP ; boucle d'attente simple (le programme SXP le plus petit qui soit!)
ENDL ; on pourrait, à cet endroit, définir un point d'arrêt pour la mesure de la durée
d'exécution

;*****
; SQ_ROOT
; Cette routine calcule la racine carrée d'un nombre 16bits non signé
; à l'entrée: input1 - octet de poids faible du nombre détruit ensuite
; ; input2 - octet de poids fort du nombre détruit ensuite
; à la sortie: le registre W donne le résultat sur 8 bits: W = INT(SQRT(input))
;*****

sq_root
mov root_mask,#$c0 ; initialiser root_mask
mov w,$40 ; initialiser le résultat
REPEAT
stc ; arithmétique 16 bits-(CARRYX)!
sub input2,w ; soustraire le résultat obtenu jusqu'à présent
IF NC ; supprimer la soustraction si Carry est effacé
add input2,w ; Carry n'est pas positionné ici (pas mis à 1)
ELSE
or w,root_mask ; mettre le bit actuel à 1
ENDIF
LOOP ; Carry peut ici être ou non mis
rl input1 ; décaler le nombre d'une position vers la gauche
rl input2
rr root_mask ; prendre MSBit de input2
EXITIF NOT root_mask.7; fin de LOOP si MSBit n'est pas mis à 1

```

```

        snc                ; SKI P plus effectif ici qu'une structure IF
        ret p             ; terminé, si Carry se trouve à 1
        cl r b    root_mask. 7
        xor        w, root_mask
        st c                ; arithmétique 16 bits (CARRYX) !
        sub        i nput 2, w
        or         w, root_mask ; mettre le bit actuel à 1
    ENDL
    xor        w, root_mask
UNTIL C                ; fin de boucle si LSBit de root_mask
                    ; est décalé vers Carry, sinon refaire
sub        i nput 2, w
IF C
    IF NZ                ; si résultat < 0 ...
        xor        w, #1 ; ... correction finale du bits LSB
    ELSEIF i nput 1. 7 ; ... ou si = 0 et bit MSB mis à 1
        xor        w, #1 ; ... correction finale du bits LSB
    ENDI
ENDI
ret p                ; terminé !
end                  ; fin du programme

```

Quelques remarques concernant la syntaxe :

Suited'instructions représente un nombre quelconque (y compris zéro) d'instructions d'assembleur SX et/ou d'instructions Sxp. Les instructions Sxp peuvent ensuite être imbriquées à autant de niveaux que l'on veut.

THEN n'est utilisé qu'en combinaison avec des instructions PAGE; elle peut sinon être omise.

Condition sert aux comparaisons (un registre SX au minimum), à l'examen de bit de registre ou de d'indicateur d'état (*status flag*).

Attribution de registre rend le nombre de passage par la boucle. Le registre concerné peut avoir été chargé auparavant mais l'attribution peut faire partie de l'instruction. Le registre de boucle peut être chargé à l'aide d'une constante, du contenu d'un autre registre de fichier SX voire du registre W. *Adresse* symbolise une adresse (non locale) à un emplacement quelconque du programme Sxp (attention cependant aux sauts revenant de routines). EXITIF-TO est la seule directive Sxp non structurée; elle permet d'économiser du code et une « sortie rapide » d'une structure Sxp. Un saut inconditionnel se fait tout simplement par exécution de l'instruction assembleur JMP Adresse.

Nous le disions, Sxp ne se sert que du registre W du SX et des registres de boucle désignés. Seuls changent le compteur ordinal le registre d'état (en cas d'examen).

Sxp tient compte, au choix, d'un DEVICE CARRYX optionnel (et quelquefois secret).

Sxp évite la collision avec des instructions d'assemblage conditionnelles : on a, juste devant des instructions

entrant en collision (ce que ne connaissent d'ailleurs que fort peu d'assembleur tels que, par exemple, le SX-KEY™ de Parallax), mise d'un « ? » dans le code-source Sxp, code que Sxp, sans la moindre activation de cette ligne de code, éliminera automatiquement de sorte que les instructions se trouvent, après le passage par l'assembleur, à disposition comme si de rien n'était.

L'écriture des instructions Sxp peut se faire tant en minuscules qu'en majuscules voire en utilisant une combinaison des deux corbeilles.

Sil devait se faire que des structures Sxp devaient sortir des limites de PAGE du SX, Sxp offre un moyen adéquat pour contourner ce problème, technique que ne supporte pas la version de démo de SXp (c'est d'ailleurs là la seule différence par rapport à la version complète).

Un exemple de Sxp

Il serait bon, pour la compréhension de l'ensemble, après autant de théorie, de proposer un exemple de programme source Sxp. Il s'agit du calcul de la racine carrée d'un nombre de 16 bits.

La fonction exécutée s'appelle très exactement : SQ ROOT = INT(SQRT(ZAHL)).

Cela signifie que le résultat peut être, à la suite de l'arrondi des entiers, trop faible d'un maximum de 0,99, ce qui est dû à la représentation du résultat sous la forme d'un nombre à 8 bits (il suffit de s'imaginer que la racine carrée de 65 535 soit égale à 255,998...). Il serait bien entendu possible, en compliquant quelque peu les choses, de calculer un résultat sur 9 bits qui aurait alors une précision de $\pm 0,5$.

Ce n'est que dans un but d'une

meilleure lisibilité que toutes les instructions Sxp sont reproduites en gras et en majuscules, les instructions d'assembleur SX étant elles écrites en minuscules (cf. **listage 1**).

Le source Sxp sera saisi à l'aide d'un éditeur de texte quelconque avant d'être sauvegardé avec l'extension « .SXP ». Depuis l'environnement du SXp, le dit fichier .SXP est, après sélection d'un certain nombre d'options (dans le cas présent avec l'option « use CARRYX », bien que dans le présent exemple il n'y ait pas d'instructions Sxp à base de comparaison et qu'il n'y ait que des interrogations d'indicateurs (*flag*).

Lors de la compilation, Sxp génère un programme source en assembleur SX doté de l'extension « .SFC », qui, le cas échéant, comporte des messages d'erreur en clair reprenant le numéro d'origine de la ligne du programme-source Sxp de sorte que l'on peut plus facilement cerner et corriger les erreurs.

Après compilation, le source en assembleur SX produit peut être assemblé à l'aide de l'assembleur SX-KEY™ ou de tout autre assembleur SX (même sous DOS) à condition qu'il reconnaisse les mnémoniques et la syntaxe spécifiées par la fiche de caractéristiques.

On pourra utiliser ce même programme source en assembleur SX pour le débogage du code-source (à nouveau à l'aide, par exemple, du SX-KEY™). On peut, optionnellement, reprendre les instructions Sxp au coeur du programme-source en assembleur SX sous la forme de commentaires, ce qui simplifie sensiblement le débogage (la cartographie de la structure est également indiquée pour une meilleure vue d'ensemble et un contrôleur plus aisé. De par WINDOWS® il n'est pas nécessaire de fermer Sxp, qui peut

Listage 2 du programme SXp

```
SC
JMP  _L1
SNZ
JMP  _L3
xor  w, #1      ; ... correction finale du bits LSB
JMP  _L4
_L3
SB   input 1.7
JMP  _L5
xor  w, #1      ; ... correction finale du bits LSB
_L4
_L5
_L2
_L1
```

Listage 3 du programme SXp

```
sc           ; comme SXp
ret p       ; est déjà terminé ici !
snz         ; comme SXp
snb  input 1.7 ; fonction logique d'interrogation:
xor  w, #1   ; NZ OR (Z AND input 1.7)
```

Listage 4 du programme SXp

```
REPEAT      ; attendre ...
UNTIL Port.5 ; ... que le bit 5 du port soit mis à 1
```

Listage 5 du programme SXp

```
; 01// REPEAT      ; attendre ...
_L1
; 01// UNTIL Port.5 ; ... que le bit 5 du port soit mis à 1
SB   Port.5
JMP  _L1
_L2
```

ainsi rester en arrière-plan pendant toute la « séance de développement »; il est facile ainsi de passer rapidement de l'éditeur au compilateur ou à l'ensemble assembleur/débugueur/programmeur.

Efficacité du code

Les instructions dont dispose, actuellement, SXp, permettent déjà la génération de code SX très efficace. On obtient un code moins efficace que celui produit par une programmation en assembleur SX pur, si l'on met, comme cela arrive 2 fois dans la dernière structure IF de notre exemple, dans une structure SXp une instruction assembleur à un mot (pas de macro !). La raison en est que l'assembleur SX permet, par le biais d'instructions de passer-oltre (*skip*) conditionnelles, de sauter une instruction SX à un mot, mais qu'il ne reconnaît pas le SXp en tant que tel et que, partant, il utilise dans ce cas-là des instructions de saut moins efficaces qu'une instruction de *skip*.

Il n'y a pas, normalement, dès lors qu'une structure SXp comporte plus d'une instruction SX à un mot, de différence d'efficacité.

Ladite structure IF donne, prise à part, après compilation, le code-source SX listé en **listage 2** (nous avons supprimé l'option « *keep SXp-Statements in the output file* » pour une meilleure lisibilité. Un code assembleur [5] bardé d'astuces produit, pour le précédent exemple, le résultat donné dans le **listage 3**.

Ce code assembleur extrêmement compact, mais également très abscons, pourrait bien entendu se substituer directement, dans le code source SXp, à la structure IF; le programme SXp aurait alors la même efficacité. On pourrait sans doute opter pour une unique instruction SKIP conditionnelle (cf. l'exemple) sans que cela n'ait de conséquence sur la lisibilité. Répétons notre conseil : attention lors de l'utilisation de macros !

Notons que, dans le cas de boucles d'attente ne servant qu'à cela, la

structure REPEATUNTIL est plus efficace que la structure WHILE-ENDW, vu qu'avec REPEAT on a interrogation en fin de boucle ce qui se traduit, comme l'illustre le **listage 4**, par une instruction de saut de moins.

Il est difficile, comme le prouve le **listage 5**, d'améliorer encore plus le code SX résultant.

SXp génère, lors de comparaisons avec la constante zéro, du code optimisé. Les boucles FOR au nombre de 256 (nombre à 9 bits !) de l'exemple précédent sont, signalons-le, exécutées correctement. On trouvera d'autres trucs et astuces concernant des erreurs de programmation classiques dans le manuel consacré au SXp.

En conclusion

Le précompilateur SXp décrit ici permet à son utilisateur de créer des programmes en assembleur structurés pour le microcontrôleur SX de Scenix™. Contrairement à ce qui est le cas avec les langages de haut niveau, il n'y a que fort peu de nouvelles choses à apprendre vu que la syntaxe utilisée est restée très simple, claire et facile à assimiler. Le code généré est quant à lui très efficace vu que le programmeur se trouve, en dépit de la programmation structurée, très proche de l'environnement assembleur.

Il serait possible d'accroître encore l'efficacité de ce précompilateur si, lors de conditions, des fonctions logiques multiples (AND-OR) et des interrogations en vrac (IN, comme sous PASCAL) étaient possibles. SIFUKIA offrait ces 2 possibilités. Des directives telles que SWITCH-CASE, ON-GOTO, etc, qui ne sont pas encore implémentées dans la version actuelle de SXp pourraient se traduire également par une amélioration de l'efficacité. On pourrait faire évoluer la boucle fort toute simple vers une instruction FOR-TO-STEP plus complexe.

Il serait souhaitable que les éditeurs (lire sociétés qui vendent) d'assembleurs intègrent au moins dans leurs produits les possibilités offertes par SXp. Notons que l'on a déjà tenté ce genre d'approche il y a quelques années déjà sur certaines familles de microcontrôleurs (dans le cas, par exemple, des assembleurs 68000 de Motorola, Mostek et Quelo) et même aujourd'hui (chez Mitsubishi et Zilog par exemple, mais de loin pas aussi évolué que SXp, tout en étant d'utilisation plus complexe). Certains compilateurs C supportent une programmation en assembleur structurée, mais on trouve cela également sur des assembleurs intégrés de chez IAR par exemple. Dans la plupart des cas ce luxe se ressent au niveau du prix.

(992037-1)

Une fois atteint l'âge de la retraite, tout ce que savent encore faire nos vieux ordinateurs c'est accumuler de la poussière. Bien dommage, vu qu'elles restent en mesure de remplir nombre de fonctions très utiles, ces vieilles « bêtes ». Le présent article va vous montrer comment, sans trop de problèmes, doter d'une sortie optique le lecteur de CD-ROM d'un PC mis au rancard, lecteur que l'on pourra ensuite, par exemple, connecter à un lecteur/enregistreur de Minidisc.

projet : K Schlott

recyclage de lecteur de CD-ROM nouvelle jeunesse pour les vieux PC

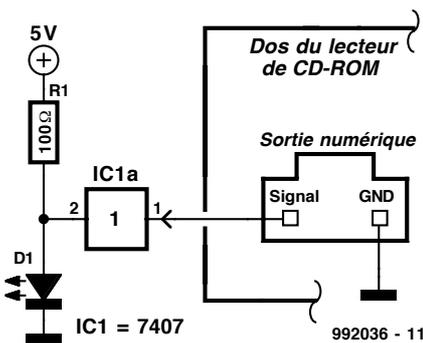


Figure 1. 3 composants, il n'en faut pas plus pour transformer une sortie numérique « ordinaire » en sortie optique.

Quels que soient leur marque, leur modèle et leur prix, tous les ordinateurs ont une caractéristique commune : leur vieillissement démarre dès à même les étagères du magasin. Après un, deux, voire trois ans, une réactualisation permet d'éliminer les premières rides, mais deux ans plus tard il n'y plus rien à y faire. On se paie un ordinateur de la dernière génération et on « range » l'ancien au grenier.

Bien souvent on envisage de le vendre, mais il apparaît vite qu'il ne vaut plus grand chose et qu'il vaut mieux le garder « en réserve ». Cette fonction de réserve n'est en fait, en réalité, quasiment jamais remplie de sorte que le dit appareil ne sert vraiment plus à rien. À y regarder de près, cela est bien dommage, un vieux 386 ou 486 pouvant parfaitement servir à bien des

choses utiles. On pourrait, par exemple, le mettre à contribution pour le pilotage d'un réseau ferroviaire miniature, l'utiliser pour commander un automate pour volets roulants sophistiqué, voire en faire un thermostat « haut de gamme » pour son chauffage central. Des possibilités à foison !

Il reste, si l'on ne sait vraiment pas que faire d'un PC complet, l'option d'utiliser individuellement certains de ses composants. On peut ainsi fort bien utiliser le lecteur de CD-ROM pour le transfert de CD audio vers un lecteur/enregistreur de Minidisc.

Sortie numérique – entrée optique

Le seul obstacle à un mariage réussi entre ce lecteur et un enregistreur de Minidisc est que le premier comporte uniquement une sortie numérique électrique et que le second ne dispose que d'une entrée numérique optique. Cette situation n'a rien d'impossible pour un amateur d'électronique « bien né ». Un examen rapide du signal disponible à la sortie d'un lecteur de CD-ROM nous apprend qu'il s'agit d'un signal biphasé de forme trapézoïdale présentant une amplitude de 5 V. Il n'y a rien de bien sorcier, à condition de disposer d'un convertisseur adéquat, à transformer ce signal en un train de zéros et de uns optiques. Les modules TOSLINK, qu'il n'est pas nécessaire de vous présenter, sont spécialement conçus à cet effet. Le sport consiste bien évidemment à

voir s'il n'est pas possible, plus simplement, à l'aide de quelques composants que l'on aurait encore dans son tiroir, d'arriver au même résultat.

Comme vous pouvez vous en douter, cela est possible. En moins d'un quart d'heure nous avons doté le lecteur de CD-ROM en question d'une sortie optique parfaitement fonctionnelle, et cela en n'utilisant en tout et pour tout que 3 composants : un tampon TIL tout ce qu'il y a de plus courant, une résistance et une LED. Le schéma de la figure 1 illustre la simplicité de notre solution.

Cela nous amène à nous intéresser au câble. Si les câbles de liaison optiques spéciaux sont fort coûteux, ils ont également l'inconvénient d'être dotés de connecteurs qui ne s'enfichent que dans les contre-embases correspondantes. Il existe certaines sources (Conrad par exemple) qui vendent la fibre optique au mètre. On pourra, si tant est que l'on utilise, pour D1, une LED dite pour « fibre optique », enficher purement et simplement l'extrémité de la fibre optique dans l'orifice prévu à cet effet dans la LED.

Le seul problème qu'il nous faille encore résoudre est d'arriver à connecter l'autre extrémité de notre fibre optique à l'entrée optique de l'enregistreur de Minidisc. Le câble optique y entre sans le moindre problème – ce n'est pas à ce niveau que se situent les difficultés – mais on constate que le signal véhiculé par la fibre optique n'arrive pas au cœur du lecteur. Bizarre. Un examen critique de l'embase de connexion optique nous permet de découvrir la raison de ce dysfonctionnement : on y trouve un minuscule petit interrupteur chargé d'assurer la commutation du mode analogique au mode numérique. Le diamètre de la fibre optique est trop faible pour activer ledit mini-interrupteur. Une épaisseur de gaine thermorétractable montée à l'extrémité de la fibre optique résout le problème.

Quoi d'autre

Signalons, pour éviter tout malentendu, qu'il n'est pas requis que le lecteur de CD-ROM se trouve dans un vieux PC. Si vous avez un PC récent et que vous teniez à utiliser son lecteur de CD-ROM comme station-maître pour des enregistrements, vous pouvez tout aussi bien monter l'électronique présentée ici dans ce PC-là. Si vous veillez de plus à ce que la LED soit disponible sur la face avant vous aurez créé une liaison optique accessible à tout moment.

(992036)

De nombreux ordinateurs et microcontrôleurs sont programmés en code hexadécimal, donc avec des valeurs composées des chiffres de 0 à 9 et des lettres de A à F. Si il est très simple d'introduire les chiffres au moyen du pavé numérique du clavier du PC, il n'en va pas de même des lettres, dispersées dans toute la partie alphanumérique. L'article suivant décrit la construction d'un clavier de 20 touches comportant tous les chiffres et lettres nécessaires ainsi que quelques caractères de commande.

Projet : Uwe Reiser

clavier hexadécimal pour PC *programmer sans problème*



Le clavier hex vient s'intercaler entre le PC et son clavier et offre par la même occasion une extension parfois bienvenue du câble du clavier. Le clavier hex offre les fonctions suivantes :

- * Génération des minuscules de a à f;
- * Génération des majuscules A à F en pressant la touche Shift du clavier du PC ou avec sa touche Caps Lock enfoncée à demeure. Si l'on veut éviter que les touches des chiffres ne produisent aussi la seconde affectation du clavier alphanumérique, il faut actionner la touche # du clavier hex;
- * Génération des chiffres de 0 à 9 conformément aux touches alphanumériques des chiffres;
- * Génération des chiffres de 0 à 9 conformément aux touches numériques des chiffres lorsque la touche # du clavier hex est active et lorsque la touche NUM du clavier du PC est active;
- * Génération de la commande du curseur conformément aux touches numériques des chiffres lorsque la touche # du clavier hex est active et lorsque la touche NUM du clavier du PC est inactive;

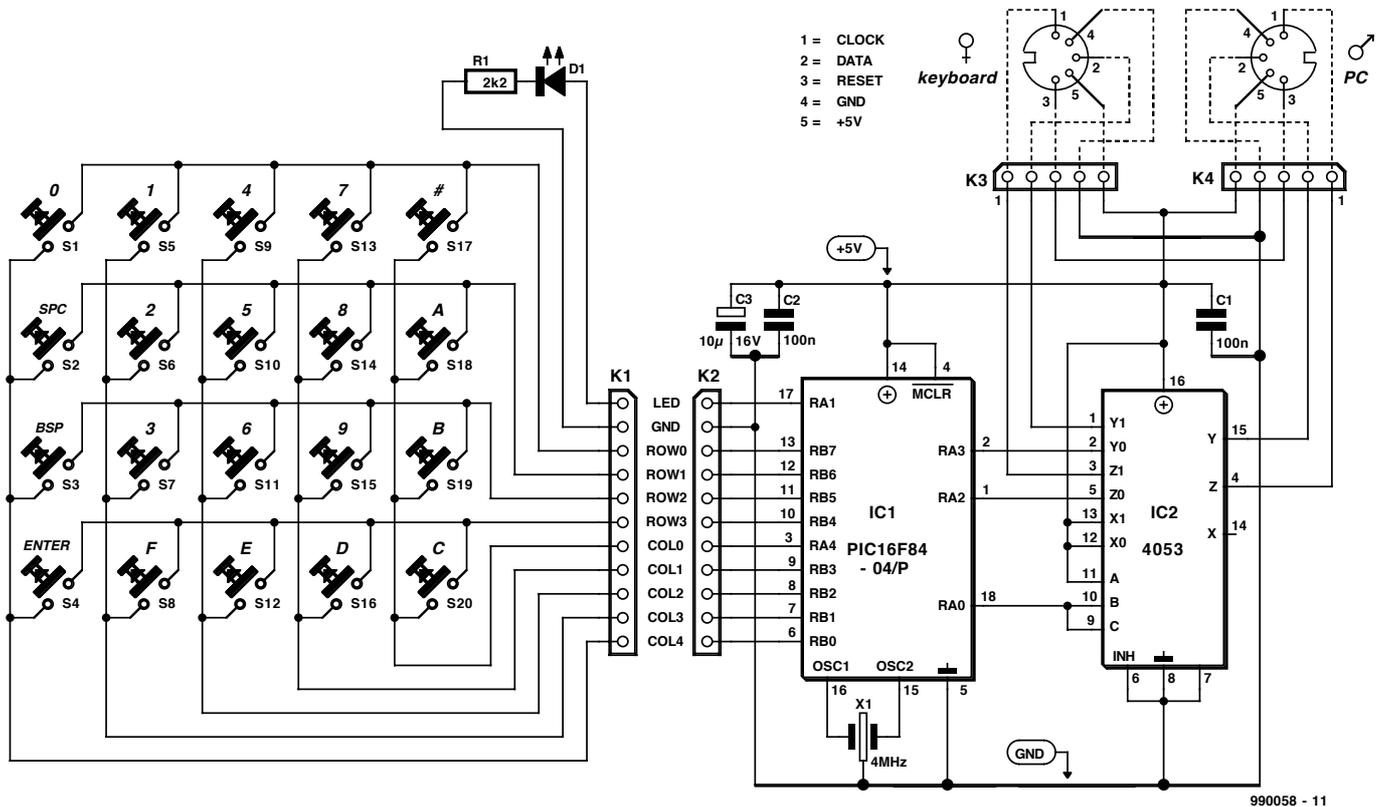


Figure 1. Schéma de connexion du clavier hex.

* Space, Backspace et Enter (possible dans toutes les combinaisons);

* Il est en outre possible d'utiliser les fonctions de commande du pavé numérique. Il faut alors que la touche NUM du PC soit inactive, ce que l'on peut déterminer en vérifiant que la LED NUM-LOCK du clavier du PC est éteinte.

Hormis les touches, le circuit du clavier hex représenté par la **figure 1** ne comporte que peu de composants, ce qui permet de réduire au maximum la description du matériel. Le contrôleur PIC 16F84 interroge continuellement les 20 touches et envoie le code de la touche pressée au PC. Le second circuit, un multiplexeur bidirectionnel, ne connecte que l'un ou l'autre des deux claviers au PC. Au repos, le clavier du PC est connecté à celui-ci. Ce n'est que lorsque le contrôleur détecte qu'une touche a été pressée que le 4053 commutue au clavier hex. Une fois le code de la touche envoyé, on revient immédiatement au clavier du PC. Il est possible, du moins théoriquement, d'interrompre le signal du clavier du PC à l'aide du clavier hex. La tension de fonctionnement de 5 V du clavier hex (et simultanément du clavier du PC) est fournie par le PC.

Le logiciel

La **figure 2** illustre le déroulement du programme que l'on peut aussi extraire du listage du code assembleur (sur la disquette EPS 996021-1 disponible auprès des adresses habituelles). La première partie du programme, INITIALISATION, permet de fixer les entrées et sorties des ports A et B ainsi que de rendre actives les résistances de charge pour le port B. Un auto-diagnostic lors de la mise sous tension fait clignoter brièvement deux fois les LED du clavier hex. La durée d'allumage, comme d'ailleurs celle d'extinction, est fixée à 109 ms par TIME4. Cette fonction indique à l'utilisateur la bonne marche des fonctions de base du processeur PIC. L'interrogation de la matrice du clavier (S1 à S20) marque le début du programme principal du contrôleur. En outre, une des 5 colonnes (COL??) est commutée de HAUT à BAS. Après un délai d'attente de 5 ms (BOUNCHE), toutes les 4 lignes sont interrogées pour déterminer si une commutation aurait transmis l'état BAS. L'arrangement des colonnes et des lignes permet de déterminer le numéro du commutateur et de s'en servir pour sauter au sous-programme approprié (SW??). Ce sous-programme attend lui aussi pendant 5 ms à l'aide de CALL BOUNCHE, puis répète l'interrogation du

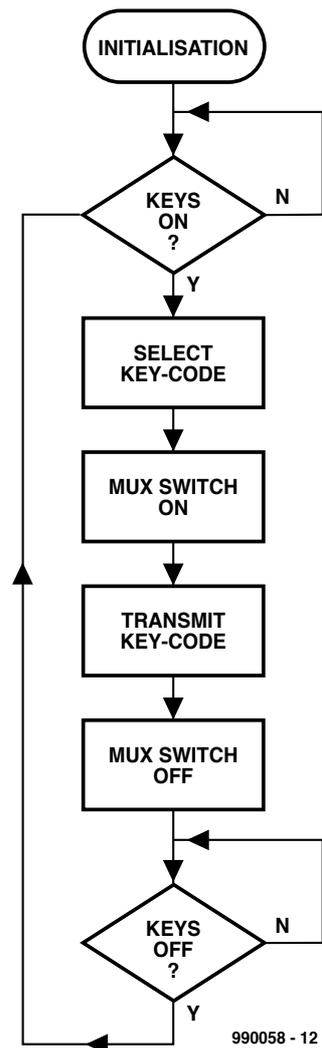
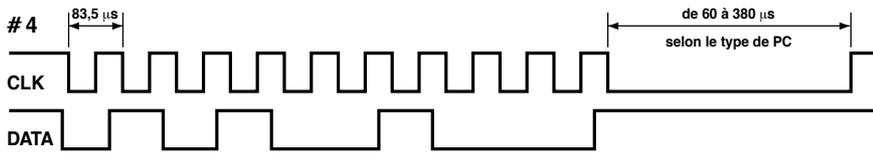
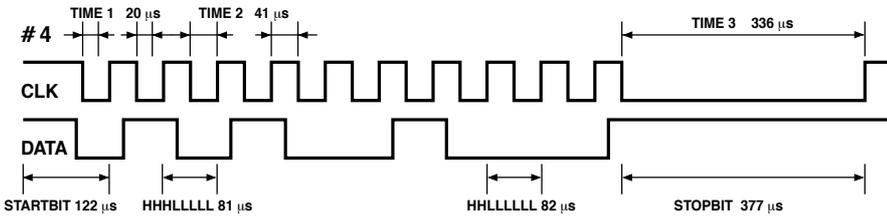


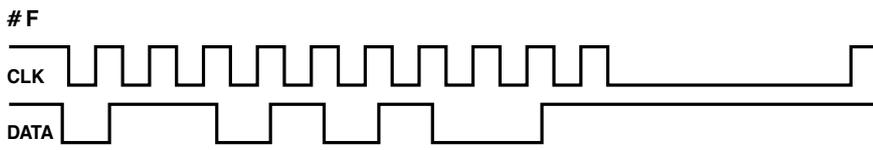
Figure 2. Organigramme.



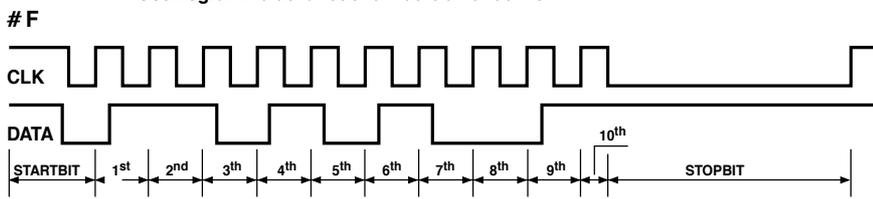
Oscillogramme de l'action sur la touche 4 du clavier numérique du PC



Oscillogramme de l'action sur la touche 4 du pavé hexadécimal



Oscillogramme de la touche F du clavier du PC



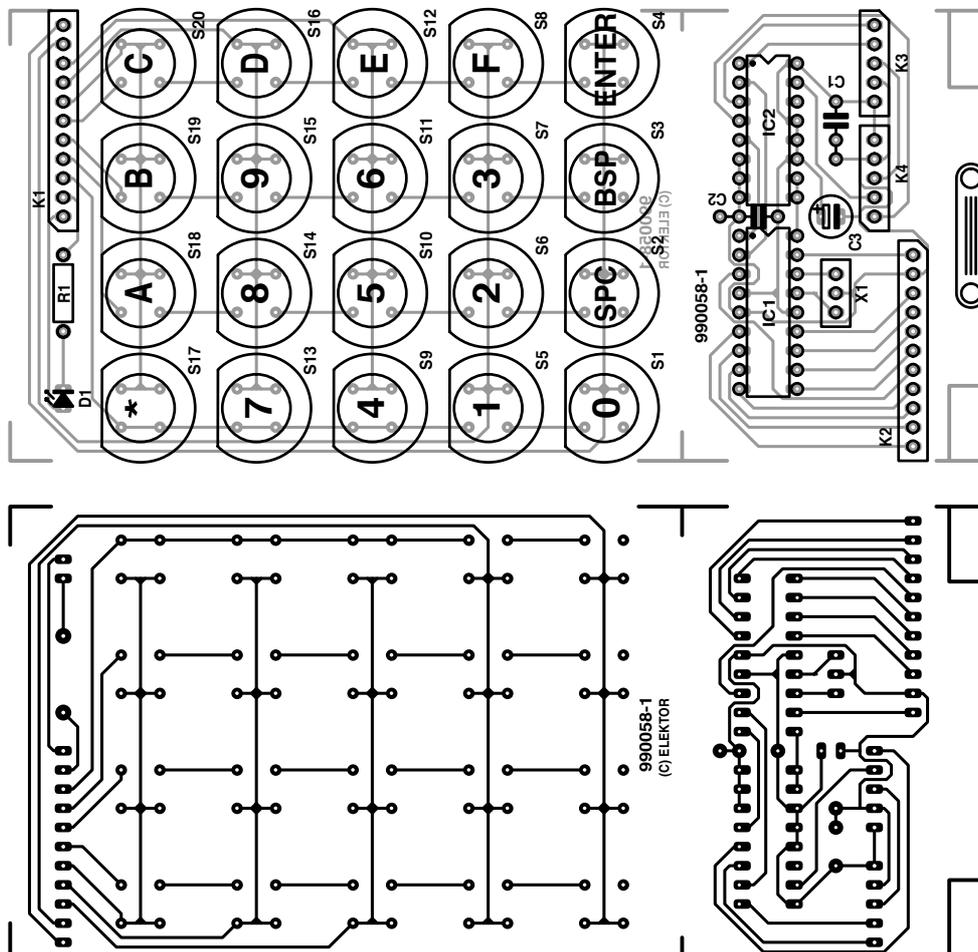
Oscillogramme de la touche F du pavé hexadécimal

990058 - 13

Figure 3. Série des impulsions des codes du clavier.

commutateur déjà identifié. Si le résultat est négatif, il faut supposer la présence d'une erreur (vraisemblablement le rebondissement du commutateur). GOTO COLUMN? répète l'interrogation de la même colonne. Si par contre le résultat de la commutation est confirmé, il s'ensuit une interrogation de la LED # des touches numériques; puis CALL RELAISSON positionne les entrées de commande Bet C du 4053 sur BAS, ce qui connecte le signal des données et le signal de synchronisation du clavier hex au PC. Pour laisser un délai de commutation suffisamment confortable à IC2, le déroulement du programme est momentanément retardé de 5 ms en appelant de nouveau la boucle d'attente BOUNCHE. La série des bits du code des touches est engendrée en appelant 12 sous-programmes. L'instruction BSF PORTA,REL provoque ensuite l'inversion du 4053. Pour éviter qu'un caractère soit émis plusieurs fois, GOTO KEYSOFF attend que toutes les touches soient libres. Les 5 colonnes sont toutes mises sur BAS, puis les 4 lignes sont interrogées (scrutées) suc-

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



cessivement. Une nouvelle interrogation du clavier au moyen de l'instruction GOTO COLUMN4 ne commence que lorsqu'aucun BAS n'est reçu.

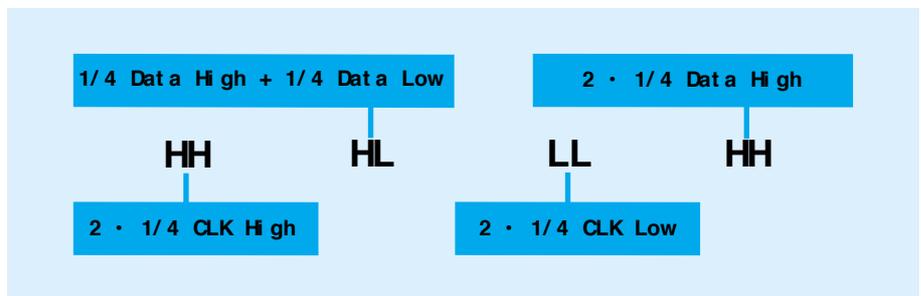
CODE DES TOUCHES

La **figure 3** montre deux séries d'impulsions représentant les signaux des liaisons de données et de synchronisation du clavier du PC pour les touches 4 et F. La fréquence de synchronisation est approximativement 12 kHz. Le changement de niveau de la liaison données, quand il est nécessaire, se produit dès la fin de la moitié de la durée haute du cycle. Comme le début et la fin des séries d'impulsions sont toujours identiques, ils sont repris comme STARTBIT et STOPBIT et transmis chacun à un sous-programme. Le signal des données et le signal de synchronisation sont engendrés en positionnant ou effaçant RA2 et RA3. La durée d'impulsion est fixée par les deux boucles d'attente TIME1 et TIME2. Si le niveau du signal des données change, TIME1 produit un quart de cycle de 20 μ s (HHB et HHH). Le changement des données suit et un autre quart de cycle termine la partie haute du signal de synchroni-

sation. Si les données ne changent pas (HHB et HHH), TIME2 (demi-cycle) produit la partie haute complète (41 μ s). Comme un changement de données ne se produit jamais pendant la durée de la partie basse du signal de synchronisation qui suit, ce dernier est toujours produit par TIME2. On obtient donc une durée d'impulsion de 81 à 82 μ s qui correspond à une fréquence de 12,345 kHz à 12,195 kHz. Comme les séries d'impulsions de chaque touche sont constituées de plusieurs informations partiellement identiques, elles sont attribuées à autant de sous-programmes (HHLLLLL, HHHLLHH, et ainsi de suite). L'ordre des lettres H et L (Low, c'est-à-dire Bas) dans les noms des sous-programmes a la signification suivante :

enfilera dans les rainures de guidage du boîtier au niveau des touches S1 à S4 une bande de platine de 65 mm de large et de 9 à 10 mm de haut (selon le type de touches) dépourvue de revêtement de cuivre et une autre sur la platine du processeur comme traverse transversale. Elles occupent l'espace des platines respectives jusqu'au bout des rainures. Les traverses transversales évitent que les touches ne s'enfoncent dans le boîtier lorsqu'on les presse. Percer un trou oblong de sortie des deux câbles dans la partie frontale du boîtier.

Il est superflu de procéder à des tests ou un ajustement si les deux platines ont été montées correctement. Étant donné que la tension d'alimentation provient du PC, il vaut toutefois mieux



Liste des composants

Résistances :

R1 = 390 Ω

Condensateurs :

C1 = 100 nF

Semi-conducteurs :

IC1 = PIC 16F84 (programmé EPS 996516-1)

IC2 = 4053

D1 = LED rouge

Divers :

S1 à S20 = touche D6-R-RD (Conrad RFA 70 84 88)

K1, K2 = embase autosécable mâle à 1 rangée de 11 contacts

K3, K4 = embase autosécable mâle à 1 rangée de 10 contacts

connecteur DIN (ou Mini-DIN) à 5 contacts mâle

connecteur DIN (ou Mini-DIN) à 5 contacts femelle

boîtier Pult-Box 25 x 109 70 50 (Conrad RFA 52 24 49)

bride anti-traction (Conrad RFA 52 46 20)

2 entretoises M3x10 mm

10 cm environ de câble plat à

11 conducteurs

TOUCHES DE LA PLATINE

Les dimensions de la platine (**figure 4**) et la disposition des composants sont adaptées à la place disponible dans le pupitre indiqué dans la liste de pièces. Pour achever le meilleur montage possible du circuit, les parties se rapportant au clavier (y compris LED et résistance série) et les parties se rapportant au processeur (PIC et multiplexeur) sont montées sur des platines séparées. Elles sont reliées par un câble plat à 11 conducteurs (K1, K2). Utilisez de préférence les touches mentionnées dans la liste de pièces car certaines de leurs pistes conductrices sont reliées entre elles. Si l'on utilise d'autres types de touches, il peut s'avérer nécessaire de placer des ponts reliant deux contacts de chaque touche, comme cela est déjà le cas pour S9, S13 et S17.

Superposez les deux platines en utilisant deux entretoises de 10 mm. La liaison avec le PC et le clavier du PC est effectuée à l'aide de K3/K4 et de deux connecteurs mâle/femelle DIN ou Mini-DIN. Une bride anti-traction maintient les deux câbles à 5 conducteurs sur la platine du processeur.

Effectuer 21 découpures dans le pupitre pour les touches et la LED. Puis monter les platines en sandwich. On

vérifier le circuit avant sa mise en service en le raccordant à une autre source de tension. La LED doit clignoter deux fois lorsque la tension de 5 V est appliquée, indiquant le bon fonctionnement du PIC. Un test simple du bloc de touches consiste à allumer et éteindre la LED en pressant la touche # S17. Si un commutateur reste fermé parce qu'il est coincé dans le boîtier ou mal monté, le circuit ne réagira plus du tout après les deux clignotements ! Il faut alors vérifier les signaux des sorties des données et de commande avec un oscilloscope ou une sonde logique. Lorsqu'on presse une touche, la série d'impulsions aux sorties synchronisation et données est similaire à celle indiquée et la sortie de commande (RA1) ne produit qu'une seule impulsion.

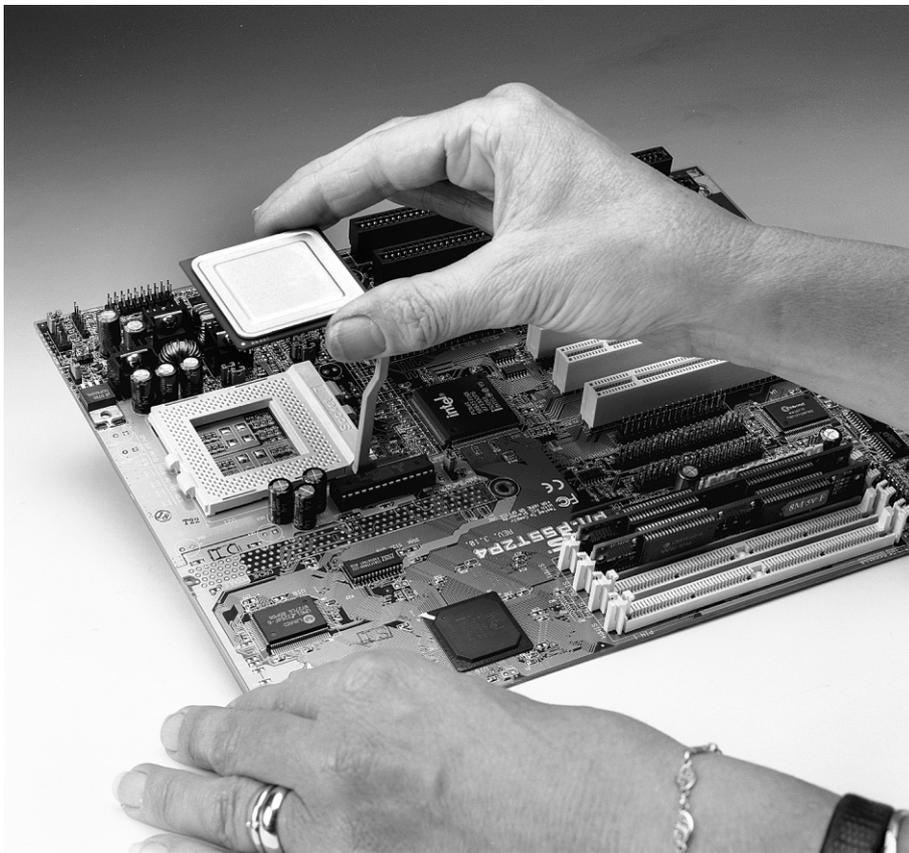
(990058)

Il est possible, par quelques modifications simples et peu coûteuses, de rendre capables, d'anciennes cartes-mères dotées d'un support dit Socket-7, de s'accommoder sans problème de processeurs modernes. Il arrive même dans certains cas qu'il suffise de tenter l'une ou l'autre combinaison de positions de cavaliers longtemps restée non documentée.

Rainer Schulze

PC-Upgrade

réactualisez votre vieille carte-mère dépassée



Le cycle d'innovation que connaît la micro-informatique est, aujourd'hui, devenu si bref qu'au bout d'un an au maximum, il faudrait, acheter un ordinateur tout neuf pour pouvoir faire tourner les logiciels les plus récents, si ce n'est pas, tout simplement, se sentir totalement dépassé.

Le présent article a pour but de montrer comment, à l'aide de quelques modifications simples, il est possible d'économiser une somme appréciable en tirant le maximum du son vieux matériel. Dans le cas le plus favorable il suffit de

moins d'un centimètre de conducteur et d'une configuration de cavaliers (*jumper*) « intelligente » pour pouvoir faire tourner sur la carte-mère ancienne génération, par exemple, un AMD K6-3 moderne à 500 MHz.

N'allons cependant pas tenter de décrocher les étoiles. Nous allons voir, en prenant comme exemple une carte-mère P/I-P55T2P4 de ASUS, comment, donner vie à un processeur AMD K6-2, au prix très intéressant actuellement, et ce à 350 MHz. Nombreux sont les PC à disposer d'une

carte-mère ASUS du type T2P4, vu les caractéristiques intéressantes et les qualités de ce produit.

L'apprentissage

C'est au plus tard lors de la première réactualisation du PC que le froment est séparé de l'ivraie et le fait d'avoir acheté son PC ailleurs que chez un spécialiste se revendra. Il se peut que l'on ait à regretter son achat dans le super-marché du coin, la durée de vie espérée du matériel que l'on y vend étant de l'ordre d'une douzaine de mois et les possibilités de réactualisation quasiment inexistantes.

Il faudra commencer, avant de passer à l'aspect matériel de la réactualisation, par mettre la main sur la version la plus récente du BIOS de sa carte-mère. Il n'est pas rare, en l'absence d'une telle mise à jour logicielle, qu'après avoir terminé le montage du matériel dans les règles de l'art, on se trouve confronté, à la première mise sous tension du système, à des événements très déroutants. Dans bien des cas, les versions de BIOS anciennes ne supportent pas, ou, au mieux, que partiellement, les fonctions les plus récentes de processeurs. Il faudra donc commencer par télécharger la dernière version du BIOS depuis Internet. En s'aidant de l'utilitaire de flashage (encadré 1) on transfère le nouveau fichier dans la ROM Flash du BIOS dont il écrase le précédent contenu. Il ne devrait plus avoir maintenant de problème à procéder à la transplantation d'un nouveau cerveau électronique.

Dans le cas de la carte-mère T2P4 utilisée ici comme cobaye, la complexité de l'opération de réactualisation varie selon la version de la carte. Les ver-



Horloge de bus et horloge de CPU interne

L'horloge de CPU interne présente un rapport direct avec l'horloge de bus. La carte-mère ASUS T2P4 à compter de la version Rev. 3xx supporte les fréquences de bus de 50 à 83 MHz. Une paire de broches, BF0 et BF1, permettent de définir des facteurs allant de 1,5 à 3,0. Il nous faut cependant, dans le cas d'une CPU travaillant à 350 MHz, un facteur de 5,5 pour une horloge de bus de 66 MHz. Un calcul montre que cela correspond à une horloge de CPU interne de 366 MHz. Pas de problème, une CPU 350 MHz travaille aussi à 366 MHz. La quasi-totalité des CPU vendues actuellement supportent sans broncher quelques MHz de plus. On peut également expérimenter avec une horloge de bus de 75 MHz. L'inconvénient majeur est le fonctionnement asynchrone à 33 MHz de l'horloge PCI. Cette fréquence de 75 MHz n'est partant intéressante que pour les applications n'utilisant que très peu le bus PCI ou ne communiquant que peu par son intermédiaire avec les périphériques. Dans le cas d'applications mathématiques qui n'utilisent, par exemple, que la RAM. Pour les jeux et autres applications graphiques, qui requièrent l'accès à la carte graphique PCI et qui partant, travaillent mieux à une horloge de bus de 66 MHz, parce que synchrones et plus efficaces.

Facteur horloge CPU/bus

Broche CPU :	BF1	BF0
Cavalier T2P4 :	JP11	JP12
x1,5 / 3,5 (5,5)	1-2	1-2
x2,0 / 4,0 (6,0)	2-3	1-2
x2,5 / 4,5 (6,5)	2-3	2-3
x3,0 / 5,0 (7,0)	1-2	2-3
Facteur sans BF2, avec BF2		

Le facteur d'horloge de bus est « généré » dans la CPU et ne peut être choisi, de l'extérieur, que par le biais des broches BF0, BF1 et BF2. L'implantation du pont BF2 se traduit par une multiplication par 2 du facteur d'horloge de bus dans le cas des CPU K6 et K6-2. On pourra obtenir une multiplication par ailleurs de ce facteur en cas d'utilisation des nouveaux modèles de CPU à noyau CTX tels que les nouvelles versions du K6-2 et tous les K6-3.

Tableau 1

Tension de noyau	contacts de JP20
2,0V	1-2, 3-4, 5-6, 7-8
2,1V	1-2, 3-4, 5-6
2,2V	1-2, 3-4
2,3V	1-2, 5-6
2,4V	3-4, 5-6
2,5V	1-2

de multiplication de l'horloge interne de la CPU par rapport à l'horloge de bus (cf. encadré 2).

Il ne reste plus qu'à régler la tension de noyau (*core*) du processeur. La tension de noyau (*V_{core}*) est mentionnée sur la plupart des CPU. Il nous faut, dans le cas de notre carte-mère T2P4

tournant à 350 MHz, une tension de noyau de 2,2 V ce qui implique la fermeture des paires de contacts 1-2 et 3-4 du bloc de cavaliers JP20. Ce faisant on interconnecte en parallèle les 2 résistances de 85 et 130 k Ω du réseau de résistances (figure 4) de la carte-mère, ce qui nous donne une

tension de noyau de l'ordre de 2,2 V à la sortie du régulateur. On consultera le tableau 1 si l'on a besoin d'une autre tension de noyau.

On vérifiera, avant d'enficher la CPU dans le support, la nouvelle tension de

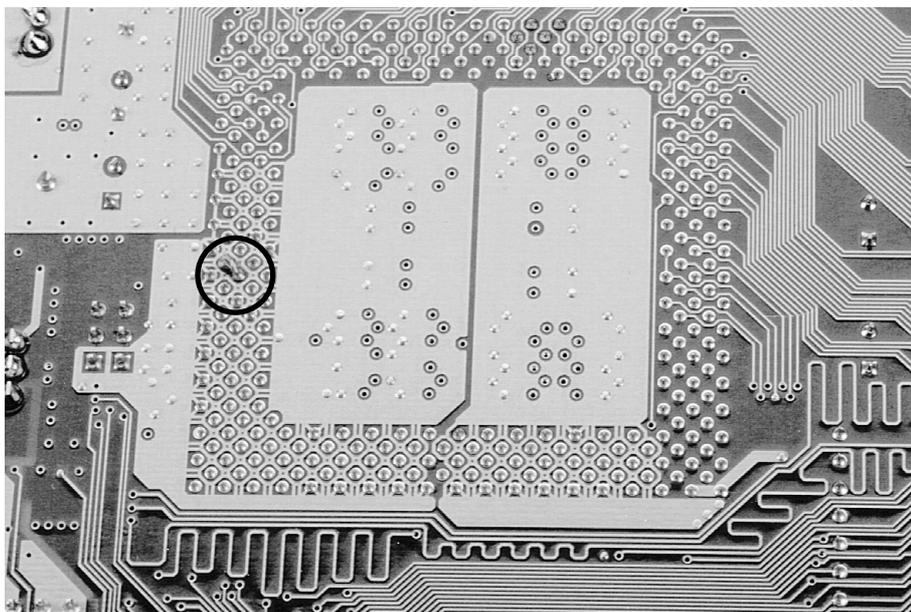


Figure 3. Support-7 vu du dessous et doté de la soudure pour le bit BF2.

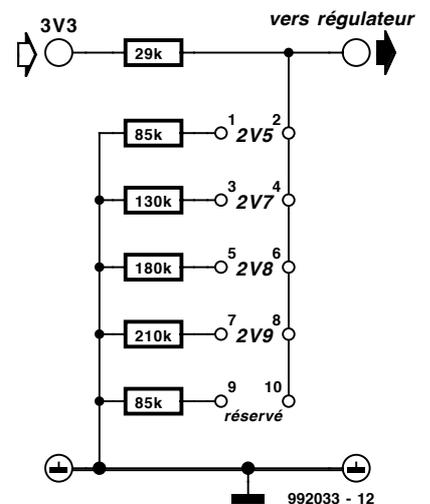


Figure 4. Sur la carte-mère ASUS la définition de la tension de noyau de toutes les CPU se fait par le biais du bloc de cavaliers JP20.

noyau à l'aide d'un multimètre. On a vite fait de trouver les lignes véhiculant la tension d'alimentation sur le support-7. Elles se trouvent pour la plupart sur la rangée extérieure de contacts. Notons la présence d'une ligne de séparation qui passe au milieu du brochage et qui est disposée parallèlement au levier de verrouillage des contacts. La moitié, dont fait partie ledit levier, véhicule la tension de noyau (Vcc2) de la CPU, l'autre moitié lui fournissant la tension d'ES (Vcc3). On commencera, pour effectuer une mesure, par implanter un conducteur fin et rigide dans, par exemple, le contact U1 (rangée de contacts disposée, vu du dessus, parallèlement et le plus près du levier, au milieu de la rangée, à mi-chemin environ de la longueur du levier et à proximité immédiate de celui-ci) et on abaissera le levier. On devrait mesurer, dès application de la tension d'alimentation, une tension de 2,15 à 2,25 V, valeur mesurée par rapport à la masse. Ces tolérances ne sont pas critiques, les CPU acceptant une plage de tolérance de 5% environ. Une tension plus élevée se traduit par une dissipation plus importante mais le fonctionnement de la CPU est plus stable qu'à une tension plus faible. On vérifiera également la tension d'ES de la CPU. On utilisera pour ce faire la broche U87 par exemple (contact médian de la rangée de contacts la plus éloignée du levier). On devrait mesurer entre 3,3 et 3,5 V.

Ceci termine le chapitre de l'alimentation de la CPU; intéressons-nous maintenant à son refroidissement. Il est impératif de toujours garantir un refroidissement correct de la CPU. En cas de doute on préférera refroidir plutôt trop que trop peu, les processeurs ayant tendance à arrêter de fonctionner, momentanément voire une fois pour toutes, lorsque leur température devient trop élevée (au-delà de 70 °C). D'éventuels dépassements de température se traduisent d'abord par des crashes imprévus du système, les cas de dépassements plus graves se traduisant par le trépas prématuré de la CPU. Les radiateurs recommandés dans le cas d'un support-7 ont des ailettes de 2 cm au moins sur lesquelles est placé un ventilateur de 50 mm. L'approche optimale consiste à veiller à ce qu'un ventilateur de 60 mm souffle latéralement sur le radiateur de la CPU, ce n'est que dans ces conditions que la dissipation passive est suffisante pour éviter que le processeur ne cesse immédiatement de fonctionner au cas où le ventilateur

devait cesser, pour une raison ou une autre, de tourner.

Si toutes les tensions se trouvent dans la plage verte et que les cavaliers ont trouvé leurs positions respectives, on pourra enficher la CPU dans le support-7. Attention au pont de court-circuit au cas où la CPU doit venir s'enficher par-dessus.

Il est temps maintenant de mettre son cœur et ses nerfs à l'épreuve de l'application de la tension d'alimentation. On devrait, en principe, constater une

accélération sensible des programmes tournant sur le PC. Au cours du processus de *boot* du système d'exploitation cela est moins sensible vu que le temps de chargement dépend principalement des caractéristiques du disque dur.

Profitez de l'augmentation de vitesse ainsi obtenue; il ne vous restera sans doute qu'à acheter un disque dur de capacité plus grande et l'un ou l'autre nouveau périphérique.

(992033)

Adresses Internet

ABIT	www.abit.com.tw (www.abit.nl)
American Megatrends	www.ami.com
AOpen	www.aopen.com.tw
ASUS	www.asus.com.tw
BMG Biostar	www.Biostar-usa.com
ChainTech	www.chaintech.com.tw
DFI	dfi.com.tw (taiwanais)
Diamond/Micronics	www.micronics.com/Micronics/support/index.html
EliteGroup	ecs.com.tw
EPOX	www.elito-epox.com
FIC	www.fic.com.tw /
Gigabyte	www.giga-byte.com.tw/giga-byte-web/newindex.htm
Intel	developer.intel.com/design/motherbd/index.htm
IWill	www.iwill.com.tw
MICROTRONICS	www.diamondmm.com/fr/products/sysbrds.htm
MSI	www.msi.com.tw
M-Technology	www.mtiusa.com
Octeck	oceanhk.com/index.html
QDI	www.qdi.nl
SOYO	www.soyousa.com
SuperMicro	tcweb.tccomputers.com/tctechweb/techinfo/Ddownloads/BIOS/SUPERMIC/bios%20index.htm
TYAN	www.tyan.com/support/bios_support.html

Adresses intéressantes pour des réactualisations en tous genres, BIOS et autres
tcweb.tccomputers.com/tctechweb/techinfo/Ddownloads/BIOS/
www.upgraders.de

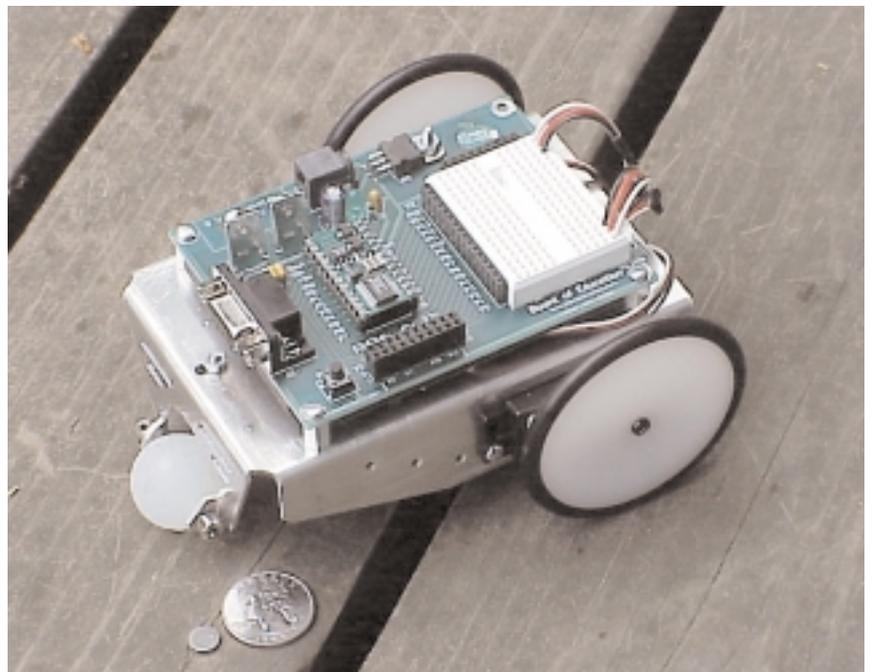


cours de programmation du BASIC Stamp (1)

1^{ère} partie : introduction & le matériel du BoE

Voici le premier article d'une série décrivant l'assemblage d'un robot rustique baptisé BoE-Bot, qui nous servira à vous initier aux bases de la programmation du BASIC Stamp. Le BoE du nom de baptême représente la notion de **Board of Education** (carte d'initiation), le Bot faisant référence au châssis du roBot. L'approche du projet est flexible et pourrait également prendre la forme d'une planche à couper le pain à roulettes. Si tant est que vous disposiez d'une plate-forme à servo pilotée par BASIC Stamp, vous n'aurez pas de problème à suivre ce cours.

par Chuck Schoeffler, Ph. D.
et Ken Gracey

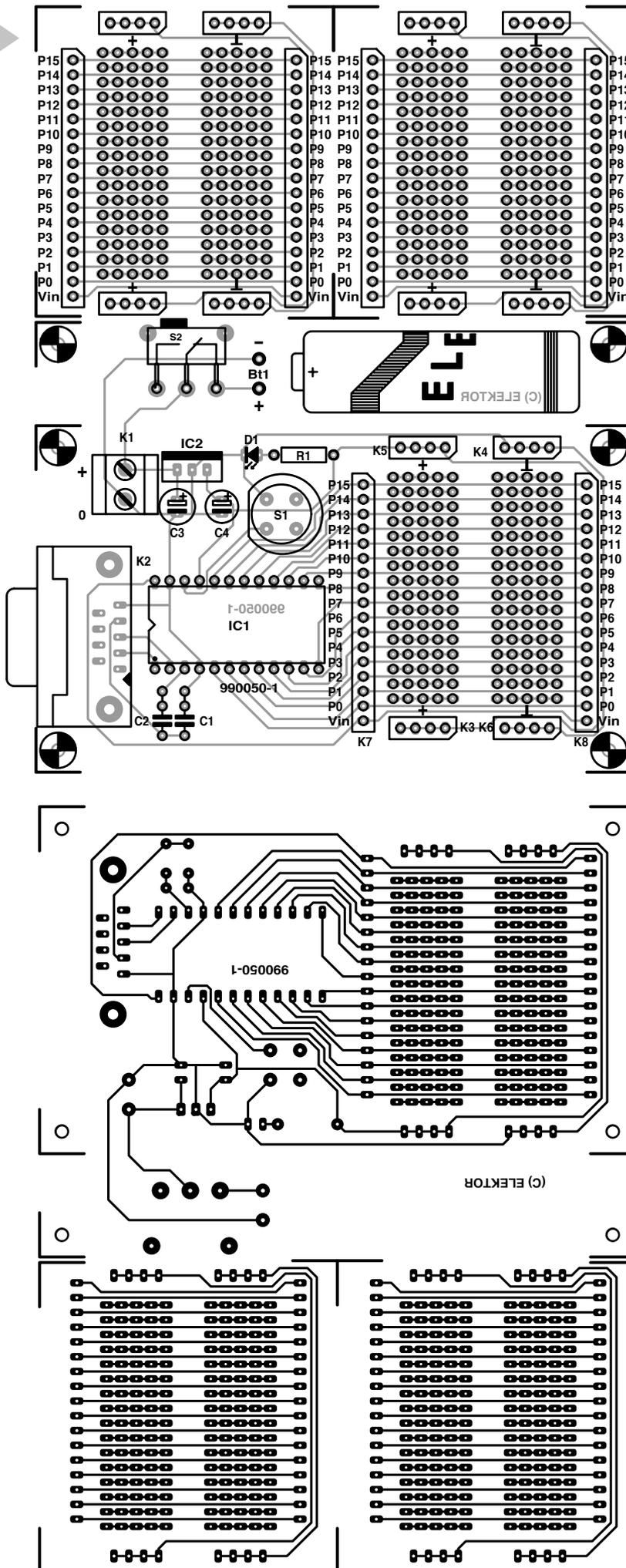


Photographie montrant le BoE de l'auteur. On notera que, sur certains points, il diffère légèrement du BoE décrit dans le présent article.

Les projets à base de robot constituent une approche idéale pour apprendre à interfacer un microcontrôleur à d'autres composants électroniques. Il est important, quel que soit l'âge de la personne concernée, de **voir** et d'**entendre** les signaux de sortie tout en pouvant les observer (les mesurer en fait) par le biais d'un multimètre ou d'un oscilloscope. Nombreux sont les enseignants à souligner l'importance de la lumière, des sons et des mouvements pour capter et garder l'attention d'un élève. Ce premier article est consacré à la réalisation d'une plate-forme robotisée reposant sur un *Board of Education* animé par un BASIC Stamp II, ensemble baptisé, nous le disions plus haut, BoE-Bot. Une fois cette

plate-forme réalisée, nous l'utiliserons pour vous donner les bases de la mise en oeuvre d'un microcontrôleur. Voici notre plan de bataille (sujet à d'éventuelles modifications) :

- **Introduction & réalisation du BoE**, septembre 1999 – l'article que vous êtes en train de lire en ce moment. La platine du BoE provient de la table de dessin d'Elektor et repose sur un modèle fourni par les auteurs. Présentation succincte du BASIC Stamp II.
- **Construction du BoE-Bot**, octobre 1999 – Processus de construction et schémas permettant la réalisation du BoE-Bot.
- **Programmation en BASIC**, novembre 1999 – Programmation



Liste des composants

Résistances :

R1 = 2k Ω

Condensateurs :

C1, C2 = 100 nF

C3 = 47 μ F/40 V radial

C4 = 1 μ F/25V radial

Semi-conducteurs :

D1 = LED rouge

IC2 = 7805

Divers :

Bt1 = pile compacte 9 V PP3 (6F22)

avec connecteur à pression

IC1 = module BASIC Stamp-2

K1 = bornier encartable à 2 contacts
au pas de 7,5 mm

K2 = embase sub D à 9 contacts

fémele en équerre encartable

K3 à K6 = embase SIL à 4 contacts
en tulipe

K7, K8 = embase SIL à 17 contacts
en tulipe

S1 = bouton-poussoir unipolaire à
contact travail encartable du type
D6-R-RD (ITC)

S2 = interrupteur à glissière encar-
table

support pour circuit intégré à
24 broches

2 boulons et écrous M3 pour fixation
de K2

Figure 2. Dessin des pistes et sérigraphie de la platine simple face dessinée à l'intention de cette réalisation.

signal DTR (*Data Terminal Ready*) présent sur la broche 4 de K2 avec la ligne d'entrée ATN du module Stamp. Une action sur le bouton-poussoir S1 permet de réinitialiser le module BASIC Stamp. On utilisera cet organe de commande lorsque l'on voudra transférer un programme vers le Stamp.

Toutes les lignes d'E/S du Stamp ainsi que sa ligne V_{in} (PWR, 9 à 15 V, cf. les fiches de caractéristiques du Stamp) sont câblées vers des embases SIL placées à proximité immédiate de la zone de prototypage de manière à permettre une connexion aisée du Stamp à vos propres circuits expérimentaux. Une alimentation embarquée simple permet l'alimentation du BoE soit par le biais d'une pile compacte soit par l'intermédiaire d'un adaptateur secteur fournissant une tension continue de 9 V sous un courant de 300 mA. Il se peut qu'il vous faille un adaptateur plus puissant si l'alimentation doit également alimenter une électronique gourmande présente sur la zone d'expérimentation à pastilles. Il se peut qu'il faille alors doter le régulateur IC2 d'un radiateur.

Attention. Il ne saurait être question de brancher l'adaptateur secteur lorsque l'alimentation se fait par le

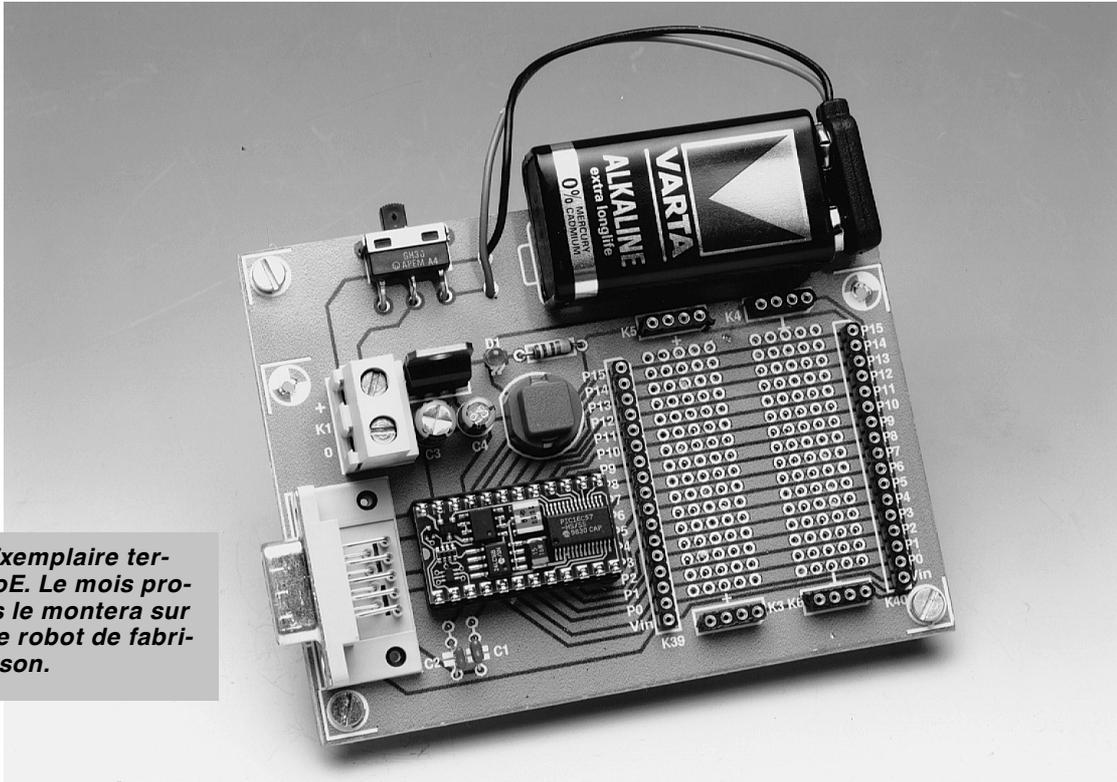


Figure 3. Exemple terminé du BoE. Le mois prochain nous le monterons sur un véhicule robot de fabrication-maison.

biais de la pile (interrupteur S2 fermé). La LED D1 s'allume pour signaler la présence de la tension d'alimentation de 5 V. Nous avons également câblé l'alimentation vers les embases SIL, vers K3 et K5 respectivement, les embases SIL K4 et K6 servant à la connexion de la masse.

LA CONSTRUCTION

La **figure 2** vous propose la platine dessinée à l'intention du BoE. Ce simple face est disponible auprès des adresses habituelles.

La mise en place des composants ne devrait pas présenter le moindre problème vu le faible nombre de composants, l'absence de composant exotique et la sérigraphie « aérée » adoptée. Assurez-vous de bien utiliser, pour K2, une **embase femelle**. Les condensateurs C3 et C4, le régulateur IC1 et (bien entendu) le support destiné au BASIC STAMP II sont des **composants** dont il faudra respecter la **polarité**. En cas de doute on se référera au schéma et à la sérigraphie.

Le module BASIC Stamp ne prendra pas place dans le support avant que vous n'ayez terminé toutes les soudures sur la platine du BoE.

Une fois terminé, le BoE, (cf. **figure 3**) met à disposition les broches d'E/S du BS2 à proximité immédiate du champ de prototypage. Il vous suffira, pour de futures expériences, d'une pince à dénuder et à couper. La platine comporte 2 autres champs de prototypage que l'on pourra séparer de la platine principale. Si vous vous en servez pour vos propres expériences il suffira de doter leur côté « pistes » de broches de longueur suf-

fisante pour pouvoir les enficher en gigogne sur la platine principale. Un exemple terminé du BoE vous offre les possibilités suivantes :

- connexion aisée de montages prototypes pouvant être pilotés par le module BASIC Stamp;
- embase sub-D à 9 contacts pour programmation du BS2 et communication sérielle en cours d'exécution (*run-time*);
- disponibilité des broches d'E/S P0 à P15, d'alimentation V_{dd} (+ 5 V) et V_{ss} (Masse) du BS-2 à proximité d'une zone de prototypage de 4 x 3 cm;
- les lignes de la sérigraphie rendent les connexions entre le port du BS-2 et les embases de la zone de prototypage;
- possibilité de découper la zone de connexion de la pile au cas où l'objet à commander dispose de sa propre alimentation (9 V_{CC} au minimum à appliquer à K1) ;
- 2 platines de zones d'expérimentation additionnelles.

ENVIE DE VOUS FAIRE LA MAIN ?

Vous pouvez, si vous aimeriez vous essayer aux microcontrôleurs, sans vous encombrer de la partie robot, télécharger le cours gratuit (en anglais) mis à disposition par Parallax à l'adresse Web suivante :

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

L'idée d'utiliser un BASIC Stamp sur une plate-forme robotique est celle de Chuck Schoeffler de l'Université de l'Idaho. La première approche du Dr. Schoeffler prenait la forme d'une configuration très bon marché basée sur le BASIC Stamp Rev. D et rien ne vous interdit de reprendre ce concept s'il vous faut une alternative aussi peu onéreuse que possible. Le BASIC Stamp Rev.D dispose d'un champ de pastilles pour expérimentations et peut être monté sur un châssis robot qui lui est propre. Cette approche pourrait intéresser les établissements scolaires.

(990050-1)

Le mois prochain :

Assemblage mécanique et calibration du BoE-Bot

Internet

<http://www.parallaxinc.com> –BASIC Stamp Manual Version 1.9, BASIC Stamp DOS et Windows Editor, exemples de programme. Distributeurs internationaux.

<http://www.stampsinclass.com> –documentation du BoE, cours de robotique (Robotics curriculum), dessins aux formats *.dxf et *.dwg, groupe de discussion concernant l'utilisation à des fins éducatives du BASIC Stamp.

chucks@turbonet.com –créateur du BoE-Bot et auteur de cette série. Assistance technique (en anglais).

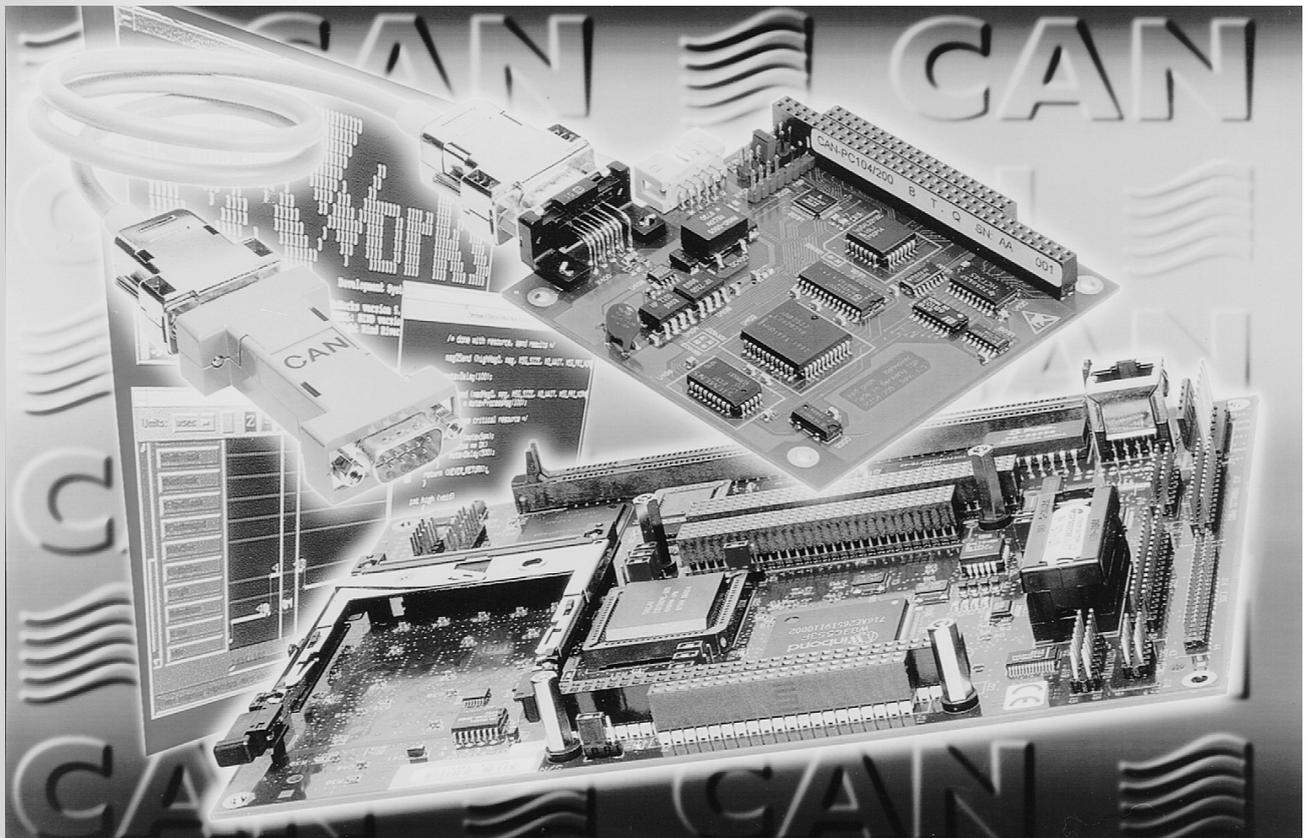
ggracey@parallaxinc.com –co-auteur de cet article. Assistance technique et questions concernant le programme éducatif.

le bus CAN

communication de données futée pour spécialiste (1^{ère} partie)

CAN, Profibus, LON, ASI, Interbus-S, Bitbus, FIP, EIB, eBUS, il n'est plus possible d'endiguer l'avancée spectaculaire des systèmes de communication en réseau, les bus industriels.

Une technique bien rodée, des protocoles aujourd'hui gravés dans le silicium des puces, des prix en baisse et une mise en oeuvre aisée, les bus industriels n'inspirent pas uniquement le milieu des fabricants, il y a des spécialistes de pointe dans bien d'autres domaines qui s'y intéressent. Nous vous proposons, dans cette petite série d'articles, de faire plus ample connaissance avec le bus CAN (*Control Area Network*, réseau local à contrôleur), en toute simplicité, à l'aide d'illustrations et selon une approche pratique. Après les bases théoriques exposées dans une première partie, nous en viendrons à une interface universelle pour un système à microprocesseur, puis des applications dans un réseau élémentaire basé sur le bus CAN.



Nous allons commencer par nous intéresser brièvement à l'histoire et à la normalisation du bus CAN, pour passer ensuite à l'exposé de ses caractéristiques principales et à sa connectivité, désignées communément par *physical layer*, le niveau matériel.

La deuxième partie traitera du protocole de transfert de données, *data link layer*, et vous fournira un tableau des composants à structure CAN proposés par quelques fabricants. Ce sera alors le moment de s'attaquer à l'étude de schémas pratiques : une interface CAN universelle, adaptable à la plupart des systèmes à microcontrôleur ou microprocesseur.

Dans la troisième partie de cette série, nous vous proposerons la construction, la programmation et la mise en service d'un petit réseau CAN à l'aide de platines pour PC et pour microcontrôleur.

LE DÉVELOPPEMENT DU CAN

Au début des années 90, l'industrie automobile, dans tous les pays, était confrontée à deux difficultés majeures dans la conception de nouveaux modèles de voitures aussi bien que de camions.

Le premier concernait l'accroissement des exigences de confort à bord des automobiles. Le client demandait des vitres électriques, des rétroviseurs orientables, des sièges chauffants et adaptables à la morphologie du conducteur, le conditionnement d'air électrique et finalement l'audiovisuel embarqué et la navigation assistée par GPS. En un mot, le marché réclamait un spectre plus large de confort et de commodité.

Le second problème, c'était naturellement le souci d'une meilleure sécurité, avec le développement indispensable de systèmes électroniques à verrouillage central, d'autres qui empêchent le démarrage, les composants ABS, en allant jusqu'à la gestion économique et écologique du moteur.

La conséquence inévitable de l'apport de toute cette électronique embarquée, c'est la nécessité soudaine de moyens de communication entre les unités dispersées sur le châssis du véhicule. On estime qu'en 2005, on comptera, au travail sous la carrosserie, jusqu'à cent microcontrôleurs séparés, obligés d'échanger entre eux des données. Les chemins de communication, en d'autres termes, l'artère ramifiée d'interconnexion, doit dès lors gonfler en proportion. Dans les voitures de luxe, on doit déjà installer 2 000 m de câble d'un poids total de 100 kg et sur différentes gammes de modèles d'un même constructeur, on peut totaliser jusqu'à 600 types différents de troncs de câbles. Il n'était désormais plus possible, commercialement parlant, de réaliser le câblage automobile selon les méthodes

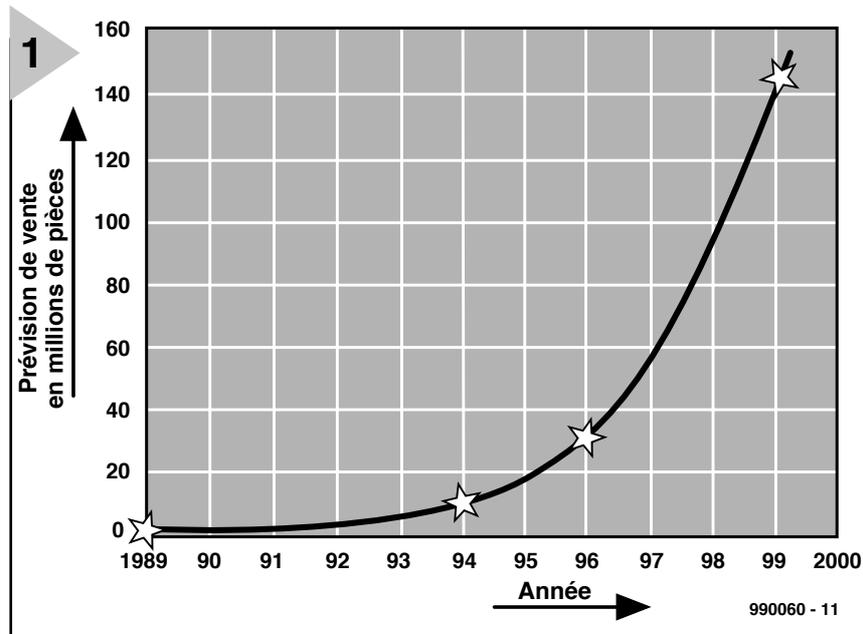


Figure 1. Progression de vente escomptée de puces pour le bus CAN [1].

traditionnelles.

L'industrie automobile, surtout en France, en Allemagne et aux États-Unis s'est donc mise à la recherche de concepts plus modernes de communication et se rallia à la technique des bus, tout en considérant les exigences particulières du transport routier :

- basse et haute vitesse de transfert des données dans la gamme de 5 kbit/s jusqu'à 1 Mbit/s pour les électroniques de confort et de sécurité ;
- très haute fiabilité du transfert de données pour une distance de Hamming (HD) supérieure à 4 ;
- optimisation pour l'échange d'informations très courtes, telles que celles des organes d'exécution ou des capteurs, d'une largeur de mot comprise entre 1 et 8 bits ;
- facilité de manipulation et faible coût de la mise en oeuvre des protocoles, compte tenu d'une production de masse ;
- construction simplifiée des bus (médiats et topologie) en vue de leur intégration au châssis.

Malheureusement, chaque grand groupe automobile s'est lancé dans la mise au point de son propre concept de bus (bien entendu incompatible avec celui des concurrents), après quoi chacun s'efforça de faire standardiser le sien à l'échelle mondiale, de manière à s'assurer d'un profit commercial maximal.

Ce n'est pas chose aisée que de formuler une norme internationale capable de satisfaire tout un chacun. On s'est ainsi retrouvé face à quatre standards restants, tous accrédités : le CAN (*Controller Area Network*) pour les versions à basse et haute vitesse, le VAN, le J1850 SCP et le J1850 DLC [1].

Le concept VAN, bien que déjà normalisé, ainsi que d'autres bus spécifiques à certains constructeurs, furent peu à peu abandonnés au milieu des années 90, au profit du bus CAN, qui devenait ainsi le leader du marché automobile mondial. C'est d'abord Mercedes qui, dès 1992, l'incorpore à ses voitures de haut de gamme, la classe S, en tant que système de transfert de données à basse et haute vitesse. Suivent alors BMW, Porsche et Jaguar et à présent, le bus CAN fait partie des voitures de milieu de gamme chez VW, Renault, Fiat, etc.

Presque au même moment, l'industrie en général, surtout dans les branches d'automatisation et de production, découvre les avantages des bus qui font leurs preuves dans l'automobile, le concept CAN en particulier, pour la mesure, la commande et le réglage, si bien qu'on le retrouve aujourd'hui comme épine dorsale des systèmes SPS, des commandes de robots et de moteurs, mais aussi dans la construction d'immeubles, les ascenseurs, l'automatisation des laboratoires, les systèmes de régulation à capteurs et organes de commande et ainsi de suite. Entre-temps, le protocole CAN a été « gravé dans le silicium », il est donc disponible sous forme de puces et l'utilisateur n'a plus à se préoccuper des finesses de diplomatie à respecter lors des communications, les circuits intégrés ont appris à s'entendre. Tout bénéfique pour la mise à niveau de systèmes à périphériques existants aussi bien que pour les nouveautés en matière d'ensembles à microcontrôleurs.

Cette maniabilité étonnante, couplée à la chute de prix des puces adaptées au CAN n'est pas un privilège réservé

ISO / OSI Layer model	CAN Bus Layer		
Layer 8 Application: "Device on Bus"	CANopen	DeviceNet	Smart Distributed System (SDS)
Layer 7 "Application Layer"	CAL: CAN Application layer for industrial Applications	DeviceNet Specifications	SDS Specifications
Layer 3 - 6	<i>Empty !!</i>		
Layer 2 "Data Link Layer"	LLC: Logical Link Control MAC: Medium Access Control acc. to ISO 11898 Result: CAN 2.0 A } Specifications CAN 2.0 B }		
Layer 1 "Physical Layer"	"Low-Speed CAN" ISO 11519-2	"High-Speed CAN" ISO 11898	

990060 - 12

Figure 2. Le bus CAN représenté selon le modèle par niveaux ISO/OSI.

à l'industrie, le particulier aussi y trouve avantage, par exemple lorsqu'il veut réaliser un réseau de communication décentralisé, un bus de terrain. La **figure 1** montre la progression attendue par les fabricants de semi-conducteurs, à l'échelle mondiale, de la mise en service de bus CAN, en regard du prix, exprimé en dollars US, d'un réseau complet fondé sur cette technologie.

LA NORMALISATION

Qui veut établir la structure d'un système de communication universel se doit de clarifier quatre points essentiels avant d'en fixer une norme.

- Comment les différents participants vont-ils se connecter au réseau, d'un point de vue électrique aussi bien que logique ? À quoi va ressembler la topologie du système ?
- Comment va-t-on introduire et transporter les données à travers le médium, qu'il s'agisse d'un câble, d'une fibre optique ou de l'air en radio ou en infrarouge ? Comment définir le niveau de signal, le raccordement des prises et toutes ces contingences ?
- Quelles règles fixer pour l'échange de données entre les intervenants ? Comment éviter les erreurs, les reconnaître, les corriger ? Quel genre

de protocole devra y présider ?

- Comment accède-t-on au canal de transfert de données, quand un des membres du réseau désire émettre et comment résoudre les conflits lorsque plusieurs stations demandent la parole en même temps ?

Il s'agit dans ce dernier cas, la solution des demandes d'accès simultanées, de ce qu'on nomme l'arbitrage du bus. Aussi longtemps que les membres du réseau restent en réception, on rencontre peu de difficultés. Dans le cadre des spécifications, n'importe quelle station réceptrice, raccordée au réseau, peut écouter toutes les informations qui passent sans causer le moindre souci. En règle générale, dans pareil système de communication, il ne peut y avoir, à un moment donné, qu'un seul poste qui émet, mais des récepteurs en service, on peut en compter autant qu'on veut.

Ces questions, et quelques autres encore, doivent recevoir une réponse explicite si l'on veut établir valablement un réseau de communication, *a fortiori* si l'on souhaite que le système obtienne une approbation internationale. C'est ainsi qu'au début des années 90, l'Organisation internationale de standardisation ISO se mit au travail pour établir une norme de bus automobile, alors que le bus CAN s'im-

posait de plus en plus.

La base de l'activité de standardisation dans le domaine colossal de la communication de données, dans un environnement ouvert et sans attache avec aucun constructeur, c'est le modèle de référence à sept niveaux ISO/OSI [2]. Pour les bus industriels, au nombre desquels on compte les bus automobiles, les niveaux 3 à 6 sont inoccupés, si bien que pour le bus CAN, il faut fournir des spécifications pour les niveaux 1, 2 et 7, comme l'indique la **figure 2** :

Niveau 1 : aspects physiques (canal de transfert de données)

Sous cette rubrique sont définies les caractéristiques du canal de transfert de données, le câble de bus, par exemple, la manière de raccorder les prises, le niveau d'émission et de réception du signal. Il existe à cet égard deux normes CAN :

ISO 11519-2 : Low-Speed-CAN. Elle repose sur les travaux entrepris au début des années 80 en Allemagne par la firme Bosch et qui obtint le soutien de INTEL, qui s'est alors chargé d'en traduire le protocole sur les puces. *Low speed* signifie ici des débits de données entre 5 kbit/s et 125 kbit/s.

ISO 11898 : High-Speed-CAN. Il s'agit dans ce cas de débits qui peuvent atteindre 1Mbit/s.

Niveau 2 : aspect liaison informatique (accès au bus et détection d'erreur)

Ce niveau-ci définit la manière d'entrer en communication avec le réseau pour y émettre des informations, le format sous lequel les données doivent être positionnées (les champs d'adresse, de données, de commandes et les binaires de contrôle d'erreur), la structure du protocole de transfert.

Les spécifications correspondantes se retrouvent également dans la norme ISO 11898.

En outre, les spécifications CAN 1991 ont encore été étendues, si bien qu'il y a aujourd'hui deux normes qui coexistent, les versions CAN 2.0A et CAN 2.0B, dont nous étudierons plus en détail les similitudes et les disparités dans la deuxième partie.

Le fait que le concept CAN se soit entre-temps répandu à de nombreux autres secteurs de l'industrie a rendu le **niveau 7**, celui des applications, particulièrement touffu, voire confus, parce qu'en fin de compte, il se traduit directement en termes d'interface pour des domaines spécifiques, on en arrive alors au niveau 8, sous forme de bus industriels. À signaler qu'au fil du temps, on a développé trois branches importantes du concept CAN : CANopen, DeviceNet et *Smart Distributed*

System (SDS). Mais comme les spécifications en sont très volumineuses, nous ne les aborderons pas dans cet article d'introduction. Toutes les variantes du concept de bus CAN, dans leur ensemble, se fondent toutefois sur les prescriptions décrites dans les niveaux 1 et 2.

Continuellement mis à jour et à niveau, le CAN ne s'est pas endormi sur ses lauriers. Dès 1982, en Allemagne, une association d'utilisateurs du concept, appelée CiA, de *CAN in Automation*, regroupe plus de 280 firmes de renom dans différents secteurs et travaille de concert à perfectionner sans cesse le bus CAN. Vous pouvez d'ailleurs obtenir des informations très fouillées sur CAN, CANopen, SDS et bien d'autres sur le site Internet de CiA [3] ou sous la référence citée en [4].

LES CARACTÉRISTIQUES

Examinons un peu plus en détail, à présent, les spécifications de la norme CAN.

Le raccordement du bus (aspects physiques) comprend la **topologie du réseau du bus CAN** et le **lien au canal de bus**.

La topologie du réseau décrit la manière dont on construit physiquement le système de communication, comment les stations se branchent sur le câble de transit.

Le bus CAN fait usage d'une topologie de bus conventionnelle, à savoir que tous les participants sont connectés à une paire unique de conducteurs, torsadée, blindée ou non, dont les extrémités sont bouclées sur des résistances équivalentes à l'impédance caractéristique de la ligne (**figure 3**).

Cette configuration assure à chaque station la faculté de communiquer sans entrave avec n'importe quelle autre station.

Pour assurer le *lien au canal de bus*, l'étage d'émission/réception d'une station CAN dispose de deux bornes, comme vous le montre la **figure 4**, l'une dite haute, CAN-High (et pas canaille, comme certains pourraient le suggérer) ou CANH, l'autre basse, CAN-Low ou CANL (cannelle, c'est déjà plus gentil).

Le transfert de données proprement dit met à profit la symétrie de la ligne pour passer le signal en mode différentiel, garantissant ainsi une haute immunité aux parasites. Dès lors, c'est la différence de tension entre les deux conducteurs du bus qui est prise en compte. Sous la norme ISO 11898, deux domaines de tension de différence sont définis pour l'interprétation des données sur le bus : l'état de bus caractérisé par un niveau récessif ou par un niveau dominant, à l'instar de la génétique, où un caractère dominant empêche un récessif, bien que présent, de s'exprimer.

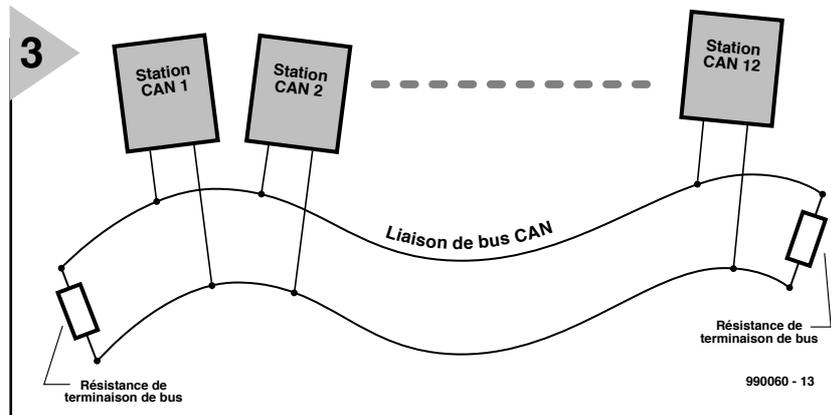


Figure 3. La topologie du bus CAN.

Tension sur... \ État du bus	récessif	dominant
CANH	2,5 V	3,5 V
CANL	2,5 V	1,5 V
Tolérance en tension différentielle : $U_D = \text{CANH} - \text{CANL}$	0 à 0,5 V	0,9 à 2,0 V

Tableau 1. Les niveaux absolus d'attaque du bus entre chaque conducteur et la masse, selon la norme ISO 11898.

On le voit, on ne travaille pas ici avec les habituels niveaux logiques « 0 » et « 1 », ce qui mérite d'évidence un mot d'explication.

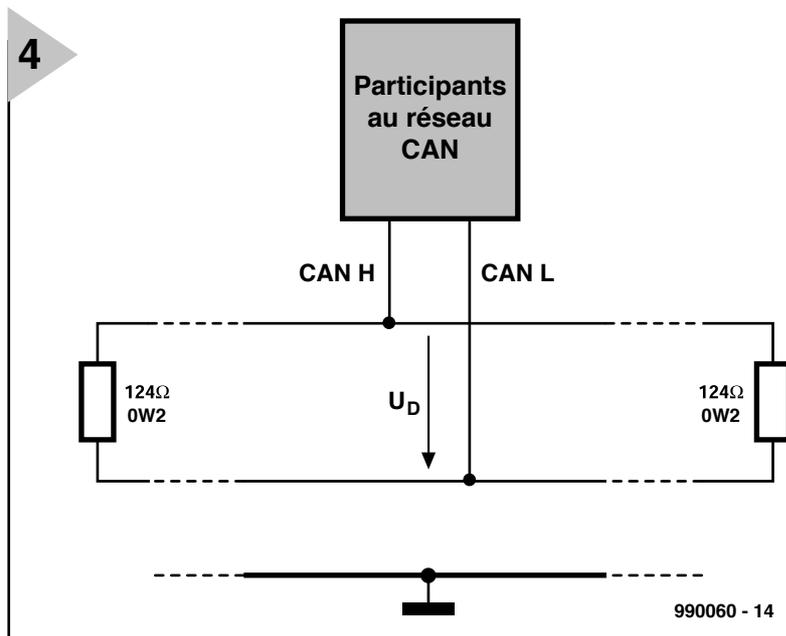
Si la différence de tension U_D entre CANH et CANL est tout au plus d'un demi volt ($\leq 0,5 \text{ V}$), le bus présente un état récessif.

Si elle atteint ou dépasse neuf dixièmes de volt ($\geq 0,9 \text{ V}$), on assiste à un état dominant.

Les niveaux absolus d'attaque du bus, c'est-à-dire ceux de chaque conducteur par rapport à la masse (locale) sont repris dans le **tableau 1**.

Naturellement, ces niveaux absolus

Figure 4. Comment raccorder une station participant au bus CAN.



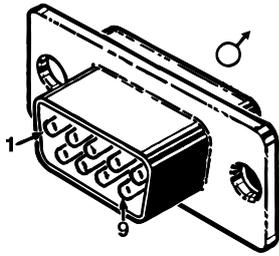


Figure 5. Brochage des connecteurs pour le bus CAN.

1 réservé	6 GND
2 CAN L	7 CAN H
3 CAN GND	8 réservé
4 réservé	9 CAN V+ (alimentation externe optionnelle)
5 optionnel: blindage	

990060 - 15

s'accommodent, en pratique, de tolérances, comme on peut s'en apercevoir à la dernière rangée du tableau.

Les prescriptions de la norme ISO 11519-2 (*Low-Speed CAN*) sont quelque peu différentes, mais actuellement, on s'aligne d'habitude sur celles de la norme ISO 11898, tant en haute qu'en basse vitesse.

Le praticien, en réalité, n'a que peu à se préoccuper de ces singularités, puisque les fabricants de semi-conducteurs lui fournissent des circuits d'émission/réception (*transceiver* ou encore *trancepteur*) tout faits, qui ne se différencient plus les uns des autres que par l'optimisation de leur comportement électromagnétique, de leur encombrement et de leur protection thermique (en cas de court-circuit entre CANH et CANL). Les niveaux de signal y sont fixés d'avance, il n'y a plus qu'à se raccorder aux conducteurs du bus et la liaison CAN est établie.

La seule chose dont il faut se soucier, c'est de savoir si ces composants fonc-

tionnent selon la norme ISO 11519-2 ou ISO 11898, avec une préférence pour cette dernière. Il faut aussi savoir que d'autres systèmes sont aptes à travailler sur la différence de tension et sont compatibles avec le transfert de données en CAN, c'est le cas par exemple de la norme RS485. D'autres informations à ce sujet sont disponibles dans l'ouvrage cité en référence [5].

Les questions fondamentales à se poser à propos du bus CAN, ce sont :

- quelle extension maximale peut-on atteindre selon un système de transfert donné,
- combien de stations peut-on raccorder au même bus ?

Les réponses à ces questions dépendent essentiellement du câble de bus utilisé. Le **tableau 2** dresse le topo pour un réseau CAN selon ISO 11898 lorsque le nombre de stations se limite à 32. Comme câble de liaison, une paire torsadée, blindée ou non, fait généralement l'affaire. Par exemple : un câble composé d'une paire torsadée de

conducteurs d'une section comprise entre 0,34 et 0,6 mm² avec des résistances de bouclage terminales de 127 Ω. En outre, il faut que la résistivité des conducteurs soit inférieure à 60 mΩ/m, mais cette condition est réalisée dès que la section dépasse 0,30 mm².

Un autre paramètre à prendre en considération, c'est la longueur de dérivation, lorsqu'une station n'est pas raccordée directement au bus CAN principal. Pour un débit inférieur à 250 kbit/s, pareille jonction ne devrait pas excéder une longueur de 2 m et 0,3 m aux débits supérieurs. Au total, il faut d'ailleurs se limiter à 30 m de bifurcations.

Pour en finir avec les considérations sur le niveau des aspects physiques, remarquons encore que les prises et leur brochage sont normalisés (**figure 5**).

Dans la deuxième partie de cette série consacrée au bus CAN, nous verrons comment se déroule le protocole de communication, quelques composants dignes d'intérêt et la réalisation pratique d'une liaison CAN.

(990060-1)

Littérature

[1] W. Lawrenz (éditeur), *CAN Controller Area Network*, Hüthig Verlag, 3^{ème} édition, 1999

[2] Phoenix Contact (éditeur), *Grundkurs Sensor/Aktor-Feldbustchnik*, Vogel Buchverlag, 1997

[3] Page d'accueil de l'association « CAN in Automation (CiA) », <http://www.can-cia.de>

[4] PRAXIS Profiline, *Controller Area Network*, Édition spéciale de Vogel Verlag, 1998

[5] K. Etschberger (éditeur), *CAN Controller-Area-Network*, Carl Hanser Verlag München, 1994

[6] A. Raab, *le bus CAN - réseau local à contrôleur*, Elektor 11/1992, pp. 40 sqq.

Longueur du bus	Câble constitutif du bus		Résistance de terminaison du bus	Taux de transfert maximale
	Résistance	Caractéristiques		
0 ... 40 m	70 mΩ/m	0,25 ... 0,34 mm ² AWG23, AWG22	124 Ω (1%)	1 Mbit/s à 40 m
40 ... 300 m	< 60 mΩ/m	0,34 ... 0,6 mm ² AWG22, AWG20	127 Ω (1%)	500 Kbit/s à 100 m
300 ... 600 m	< 40 mΩ/m	0,5 ... 0,6 mm ² AWG20	de 150 Ω à 300 Ω	100 Kbit/s à 500 m
600 m ... 1 km	< 26 mΩ/m	0,75 ... 0,8 mm ² AWG18	de 150 Ω à 300 Ω	50 Kbit/s à 1 km

Tableau 2. Débit de données, longueur de bus, type de câble et impédance caractéristique [3].

EDiTS Pro

booster & Cie

Cet article remplit une double fonction. Il a pour but, d'une part, de proposer une interface simple, à l'intention de ceux qui voudraient, avec un système EDiTS Pro, utiliser un booster Märklin qu'ils posséderaient déjà. Il comporte en outre 2 trucs destinés à ceux d'entre nos lecteurs qui redouteraient de coupler la puissance respectable d'un booster d'EDiTS aux rails de leur réseau ferroviaire.

Dans l'univers du modélisme ferroviaire la compatibilité est un aspect important, raison pour laquelle nous annonçons, dans la première partie de « EDiTS Pro » déjà, que nous avons l'intention de permettre, au travers d'un adaptateur, l'utilisation, entre autres, du booster Märklin. Nous tenons ici notre promesse. Nous avons en effet développé spécialement dans ce but une interface simple qui viendra se brancher sur l'une des embases à 5 contacts présentes sur la face arrière du booster Märklin.

La **figure 1** vous la montre cette face

face se trouvera alimentée par le biais de la broche 4 de K8. Signalons, à titre d'information complémentaire, la présence d'une signalisation de court-circuit du booster; ce dispositif présente normalement (hors court-circuit donc) un niveau bas en sortie, signal transmis à l'interface par le biais de la broche 5.

SCHEMA DE L'INTERFACE

Intéressons-nous maintenant à l'électronique de l'interface proprement dite. Un rapide examen de la **figure 2** a vite fait de vous convaincre de sa simplicité. Une triplette d'inverseurs à

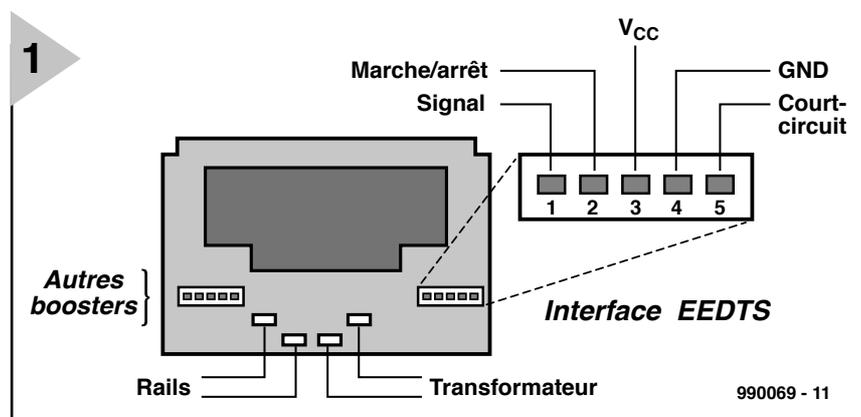


Figure 1. Brochage du booster Märklin.

arrière. Les embases à 5 contacts sont identiques, de sorte que l'on pourra indifféremment en utiliser l'une ou l'autre. L'unité de commande d'EDiTS Pro comporte une sortie de signal à niveau TTL spécialement prévue pour être connectée au dit booster. Ce signal, disponible sur la broche 3 de K8 (cf. la 1^{ère} partie publiée en juin) est appliqué directement à la broche 1 du booster Märklin; il n'est pas nécessaire, à l'inverse de ce qui est le cas avec le booster EDiTS, de le couper en cas de court-circuit.

La borne de masse (GND) de l'unité de commande (broche 2 de K8) est interconnectée à la broche GND (broche 4) du booster. En cas de mise en fonction de l'unité de commande notre inter-

trigger de Schmitt, quelques composants passifs, il n'en faut pas

plus. L'embase K1 est reliée à l'embase K8 de l'unité de commande et K2 attaque, par le biais d'un câble à 4 conducteurs, le booster Märklin.

À l'instant de l'application de la tension d'alimentation le condensateur C1 est encore déchargé. Cette situation force la bascule bistable (flipflop) RS que constituent les portes IC1b et IC1c à prendre un état parfaitement défini (la sortie de IC1b présente un niveau haut, celle de IC1c un niveau bas). Au bout d'un certain temps, C1 se sera chargé au travers de R3, mais la présence de la diode D2 lui enlève tout effet sur le fonctionnement de l'électronique.

La résistance R4 a pour fonction d'as-

sur une décharge de C1 après la disparition de la tension d'alimentation, de sorte que C1 pourra remplir à nouveau sa fonction lors de la mise sous tension suivante.

Nous savons donc qu'à la mise sous tension la bascule RS applique un niveau bas à la broche 2 du booster, ce qui se traduit par la mise du booster en mode veille (*idle*) de sorte qu'il n'y a pas de signal de sortie. Dès l'action sur la touche S2 (GO), la bascule change d'état, ce qui se traduit par l'allumage de la LED D1 et le passage au niveau haut de la broche 2 de l'embase du booster; celui-ci est, partant, activé, ce que confirme l'allumage d'une LED présente sur le booster proprement dit. En cas d'apparition d'un court-circuit la broche 5 de l'embase du booster (et de K2 bien entendu aussi) passe d'un niveau bas vers un niveau haut.

IC1a convertit ce changement d'état en une impulsion descendante qui a le même résultat qu'une action sur la touche S1 (STOP) : le booster est mis hors-fonction. Dès l'élimination du court-circuit (ou dès que le convoi responsable du court-circuit est remis sur les rails), l'ensemble du système peut être réactivé par action sur la touche GO. La signalisation de court-circuit étant câblée sous la forme d'une fonction OU (OR), un court-circuit se traduit par la mise hors-fonction des autres boosters éventuellement pris dans le circuit.

RÉALISATION DE L'INTERFACE

Vu le petit nombre de composants mis en oeuvre, la réalisation de cette interface à l'aide de la platine représentée en **figure 3** ne devrait pas poser de problème. On peut fort bien envisager l'utilisation d'un morceau de platine d'expérimentation à pastilles pour construire ce montage. On vérifiera avec soin la réalisation de la platine et

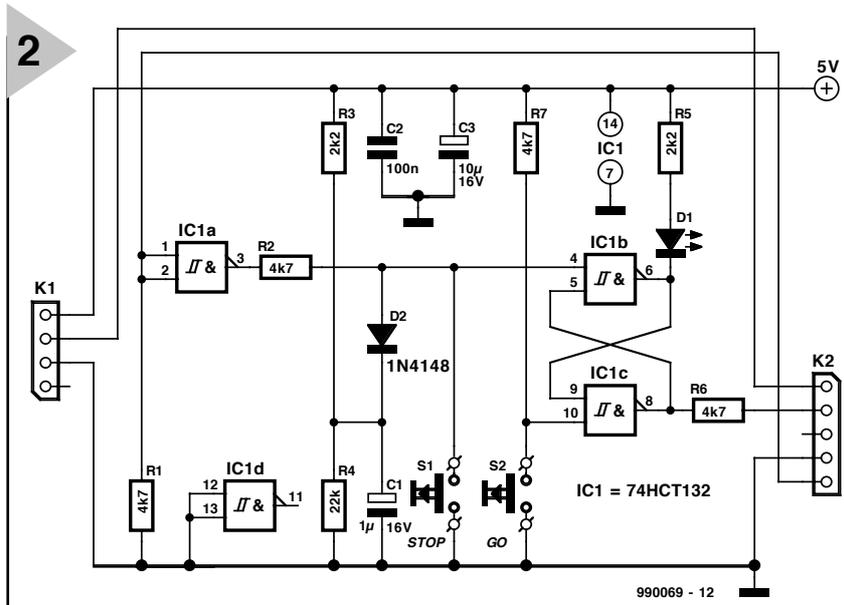


Figure 2. Schéma de l'interface requise.

le câble des connecteurs avant d'utiliser le montage.

Dernière remarque concernant la mise sous tension : il est impératif de toujours commencer par mettre l'unité de commande sous tension avant les boosters Märklin.

PROTECTION DU BOOSTER

Certains ferro-modélistes ont des sueurs froides à la seule 'idée d'appliquer aux rails de leur réseau la totalité de la puissance d'un booster de forte puissance. Ils ont des cauchemars de roues soudées aux rails, de départs de feux et autres catastrophes latentes.

Bien que la protection contre un court-circuit dont est doté le booster EDiTS minimise, dans la pratique, ce genre de dangers, il peut se faire, dans la situation la plus défavorable, qu'il y ait circulation, par les rails, d'un courant de court-circuit de quelque 10 A.

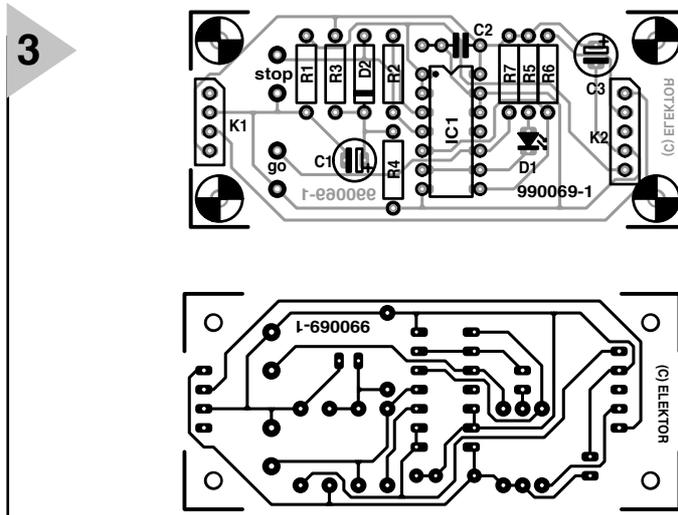
Figure 3. Dessin de platine conçue à l'intention de l'interface pour booster Märklin.

Une solution simple et bon marché à ce problème consiste à réparer le signal de sortie du booster sur, disons, 5 sections distinctes protégées chacune par un fusible de 2 A. C'est très précisément ce qu'illustre le schéma de la **figure 4**. On trouve, dans les magasins de composants électroniques, de beaux fusibles thermiques à bouton de réactivation intégré. L'alimentation du réseau ferroviaire se fait alors par le biais de ces 5 sections qui se doivent d'être isolées les unes par rapport aux autres. Il suffira pour cela d'isoler tout simplement le rail central, voire l'un des deux rails si l'on travaille en mode bi-rail (2 rails seulement).

La **figure 5** vous propose une solution bien plus élégante à ce problème de gestion de courant. Il est en effet fait appel ici à une paire de boosters distincts chargés de fournir chacun la moitié du courant.

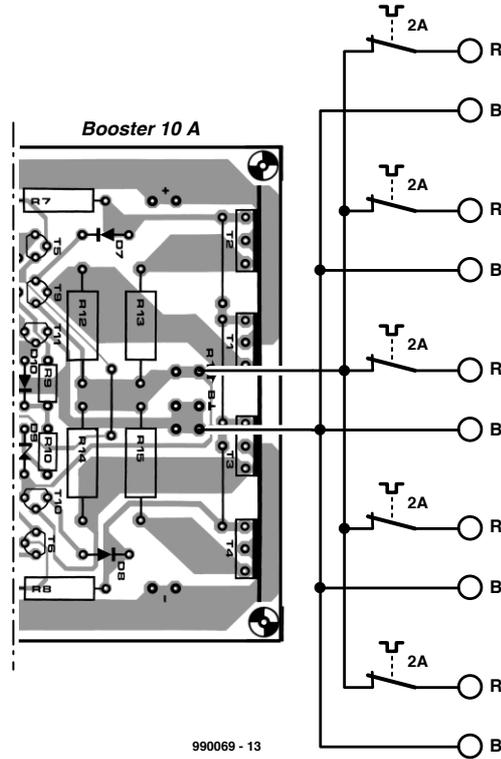
Il est facile, dans le cas du booster EDiTS, de

- Liste des composants de l'interface du booster Märklin**
- Résistances :
R1, R2, R6, R7 = 4kΩ7
R3, R5 = 2kΩ2
R4 = 22 kΩ
- Condensateurs :
C1 = 1 μF/16 V radial
C2 = 100 nF
C3 = 10 μF/16 V radial
- Semi-conducteurs :
D1 = LED faible courant
D2 = 1N4148
IC1 = 74HCT132
- Divers :
K1 = embase SIL mâle à 4 contacts
S1, S2 = bouton-poussoir unipolaire
K2 = embase SIL mâle à 5 contacts



4

Figure 4. Distribution de la sortie du booster en 5 sections dotées chacune de sa propre protection.



990069 - 13

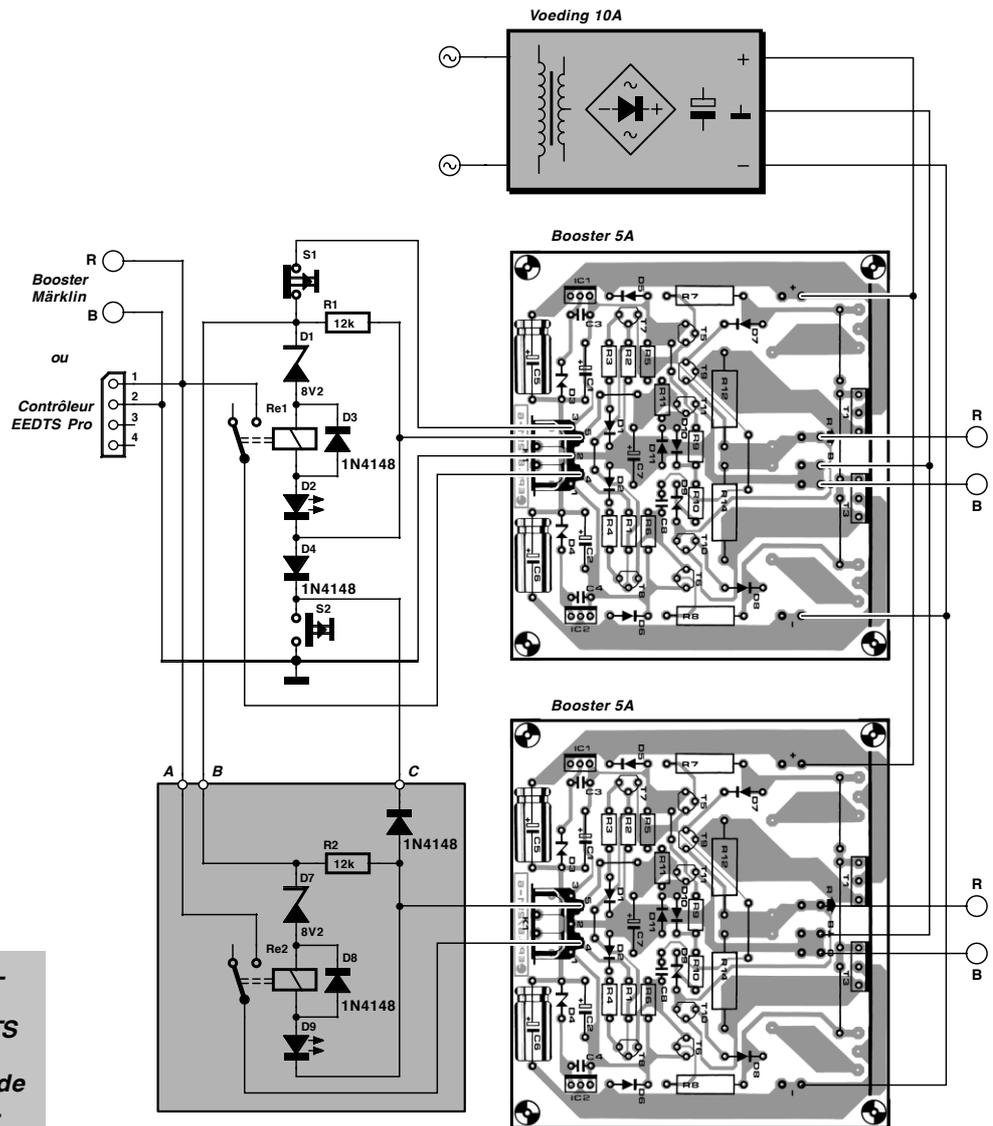
diviser par 2 le courant de sortie en n'implantant ni T2, ni T4 ni R13 ni R15. La limitation de courant entre en fonction à un courant de l'ordre de 5 A.

Comme l'illustre la figure 5, chaque booster sera doté de sa propre interface pour booster. On pourra, sur le second exemplaire, ne pas implanter les boutons-poussoirs S1 et S2. On pourra, si l'on veut répartir le courant sur un nombre de boosters supérieur à 2, interconnecter les interfaces pour booster additionnelles par le biais des points A, B et C. On pourra ainsi interconnecter un nombre pratiquement illimité de booster.

L'alimentation des différents boosters pourra se faire, comme l'illustre la figure 5, par le biais d'une unique alimentation de forte puissance mais rien n'interdit de doter chacun d'entre eux de sa propre alimentation. Une version 5 A requiert un transformateur de 150 VA et une paire de condensateurs de filtrage de 10 000 μ F. Les lignes de masse des deux alimentations devront être interconnectées ! (990069)

5

Figure 5. Utilisation d'une paire de booster EDiTS pour une consommation de courant réduite.

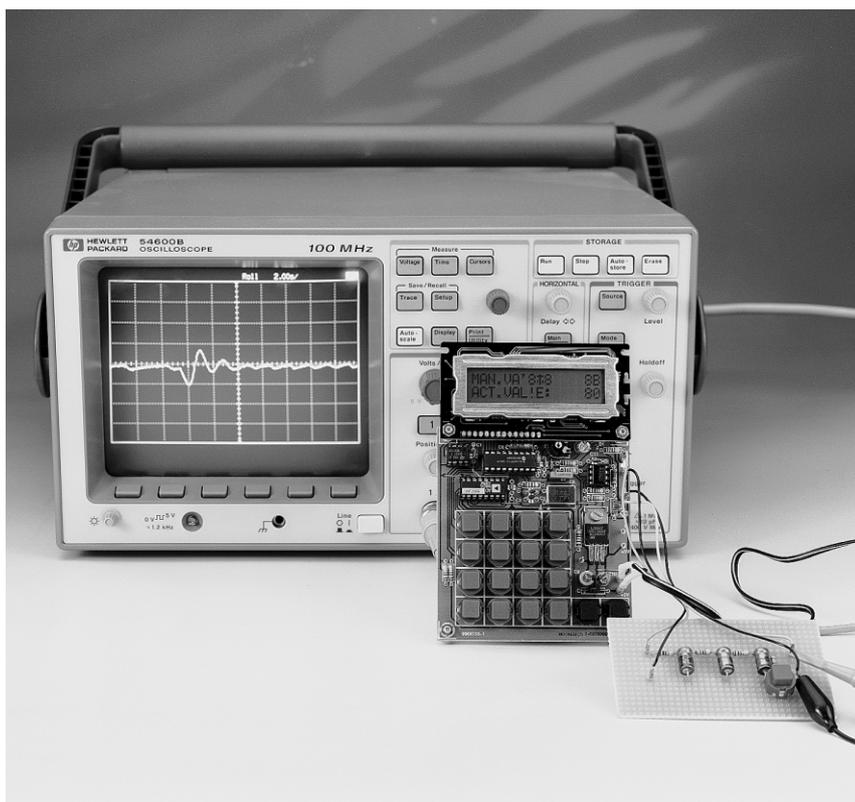


990069 - 14

régulateur P.I.D. numérique

à PIC dans le système réglé

On se représente normalement un régulateur P.I.D. sous la forme d'un circuit de régulation analogique basé sur quelques amplificateurs opérationnels. Nous indiquons ici comment réaliser un régulateur numérique de ce type. Ce circuit de régulation universel à PIC et convertisseur numérique-analogique permet de spécifier par clavier tous les paramètres de régulation en fonction de chaque application.



Le régulateur numérique P.I.D. à PIC se prête particulièrement bien à l'étude de l'influence des paramètres de systèmes réglés simples. Mais rien n'empêche de l'incorporer dans une application concrète. La régulation complète d'une application requiert, outre le régulateur P.I.D., un actionneur et un capteur appropriés. Un microcontrôleur économique PIC16C71 constitue le cœur du circuit ; son convertisseur analogique-numérique numérise la valeur mesurée (valeur effective) et la transmet au programme. Ce dernier détermine la déviation de chaque valeur mesurée par rapport à la valeur de consigne et

en tire la réponse transmise au convertisseur (externe) numérique-analogique. Les paramètres de régulation introduits par pavé de touches sont mémorisés à demeure dans une EEPROM.

Pour ne pas dépasser les 1 k de mémoire de programmation du PIC, il faut positionner les paramètres en hexadécimal. Leur représentation sur l'afficheur LC fournit toutefois un peu plus de confort.

INFORMATIONS DE BASE

Les circuits de régulation sont communs dans l'électronique, par exemple le régulateur de tension présent dans chaque circuit ou presque. Il est un peu plus compliqué d'asservir d'autres grandeurs physiques (comme le nombre de tours, la vitesse, la température, la pression, le débit, etc.). Les éléments de la technique de réglage,

Caractéristiques :

- Comportement de régulation P.I.D.
- Introduction des paramètres par clavier (hexadécimal) et affichage LC (alphanumérique)
- Paramètres réglables :
 - valeur de consigne
 - grandeur réglante initiale
 - limite supérieure
 - limite inférieure
 - intervalle d'échantillonnage (100 ms – 25 s)
- Mémoire EEPROM des paramètres
- Conversion analogique-numérique 8 bits et numérique analogique (valeur mesurée et grandeur réglante)
- Microcontrôleur 16C71

consacrée à ce genre de problème, sont brièvement récapitulés ici.

La technique de réglage désigne par *processus* l'objet de la régulation. Le but consiste à maintenir la *grandeur de consigne* d'un processus à la *valeur de consigne* désirée, même sous l'influence de *perturbations* contraires. Il faut pour cela que le régulateur mesure la *valeur effective* y , calcule la *déviaton de réglage* e par comparaison avec la valeur de consigne w et, selon le résultat de cette comparaison, produise une *grandeur réglante* u ayant le résultat désiré. La caractéristique de toute régulation est son caractère de *boucle fermée* (**figure 1**). Le signal d'entrée du processus asservi et du régulateur produit un signal de sortie bien déterminé qui, dans la plupart des cas, *dépend du temps*. La caractéristique principale du processus et du régulateur se dénomme *réponse à un échelon*. Elle indique la réaction à un signal d'entrée u qui effectue un saut brusque puis reste constant. Le comportement reproduit dans le **figure 2** est celui de nombreux processus : si on applique par exemple une tension constante à une résistance de chauffage, la température résultante suit un comportement temporel similaire.

Le régulateur peut être aussi considéré comme un élément de transfert qui réagit au signal d'entrée « déviation de réglage » par un signal de sortie « grandeur réglante ».

La forme que prend cette réaction différencie les régulateurs proportionnels, intégraux et différentiels. L'élément de transfert d'un régulateur proportionnel (régulateur P) n'est en fait rien d'autre qu'un amplificateur. La modification du signal de sortie est proportionnelle au signal d'entrée, l'amplification est le seul paramètre ajustable. Comme le laisse déjà supposer leur nom, l'élément de transfert d'un régulateur intégral (régulateur I) est un élément d'intégration (intégrateur) et celui d'un régulateur différentiel (régulateur D) un élément de dérivation (différentiateur). Les constantes de temps forment des paramètres supplémentaires à l'amplification. Un type de régulateur « tout-terrain » réunit ces trois types en un : le régulateur P.I.D. Comme l'indique le **figure 3**, la grandeur réglante u est formée de la somme des sorties des trois types de régulateurs ; la contribution de chaque régulateur peut être « pondérée » différemment selon le comportement de régulation désiré. La relation mathématique entre la grandeur réglante et la déviation de réglage se nomme *algorithme de régulation* et celle d'un régulateur P.I.D. analogique prend la forme d'une équation différentielle.

L'incapacité d'un régulateur analogique à saisir continuellement la valeur effective le distingue principalement d'un régulateur numérique. La saisie

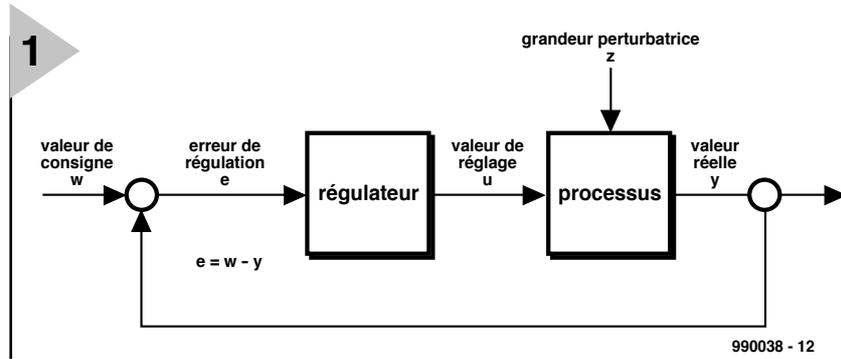


Figure 1. Boucle de régulation fermée d'un processus.

de chaque mesure est effectuée au rythme d'un intervalle d'échantillonnage (T_0) donné au cours duquel le régulateur évalue la déviation de réglage et s'en sert pour calculer la grandeur réglante au moyen de l'algorithme de régulation. L'algorithme de régulation est basé sur l'équation suivante (dérivation dans le texte encadré « **Bases d'un régulateur P.I.D. numérique** ») :

$$u_k = u_{(k-1)} + q_0 * e_k + q_1 * e_{(k-1)} + q_2 * e_{(k-2)}$$

L'indice k est le numéro de séquence des mesures.

La grandeur $e_{(k-1)}$ représente donc la déviation de réglage de la mesure précédente. Les paramètres de régulation q_0 bis q_2 peuvent être approchés au moyen des équations ci-dessous dans lesquelles on introduit les valeurs fournies par la réponse du processus à un échelon :

$$q_0 = [1,5 T_G / (K_p * T_u)] * (1 + T_u / 2T_0)$$

$$q_1 = [1,5 T_G / (K_p * T_u)] * (T_0 / 2T_u - T_u / T_0 - 1)$$

$$q_2 = 3T_G / 4K_p T_0$$

La réponse du processus à un échelon est déterminée expérimentalement. Il faut aussi que les paramètres remplis-

sent les conditions suivantes :

$$q_0 > 0$$

$$q_1 < 0, |q_1| > q_0$$

$$|q_0 + q_1| < q_2 < q_0$$

L'intervalle d'échantillonnage T_0 devrait être de l'ordre d'un dixième du temps de réponse du processus à un échelon pour parvenir à 95 % de la valeur définitive.

MATÉRIEL ET LOGICIEL

L'entrée du régulateur est prévue pour une tension de 0 à 5 V qui peut provenir de n'importe quel capteur et est transmise au convertisseur analogique-numérique du PIC 16C71 par l'amplificateur opérationnel IC5a servant de tampon. Le filtre passe-bas R8/C8 à la sortie de l'amplificateur opérationnel retient les perturbations de fréquence élevée à l'entrée A/D (RA0 du PIC).

La sortie de régulation fournit aussi une tension de 0 à 5 V. Le courant de sortie fourni par l'amplificateur opérationnel IC5b suffit largement pour le circuit de simulation du processus asservi mentionné au début de cet article. Il faut naturellement prévoir un étage de puissance commandé par

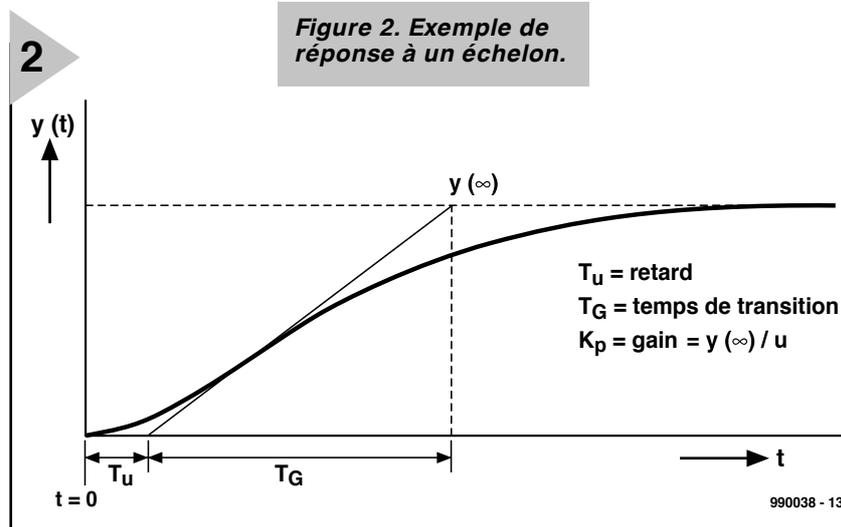


Figure 2. Exemple de réponse à un échelon.

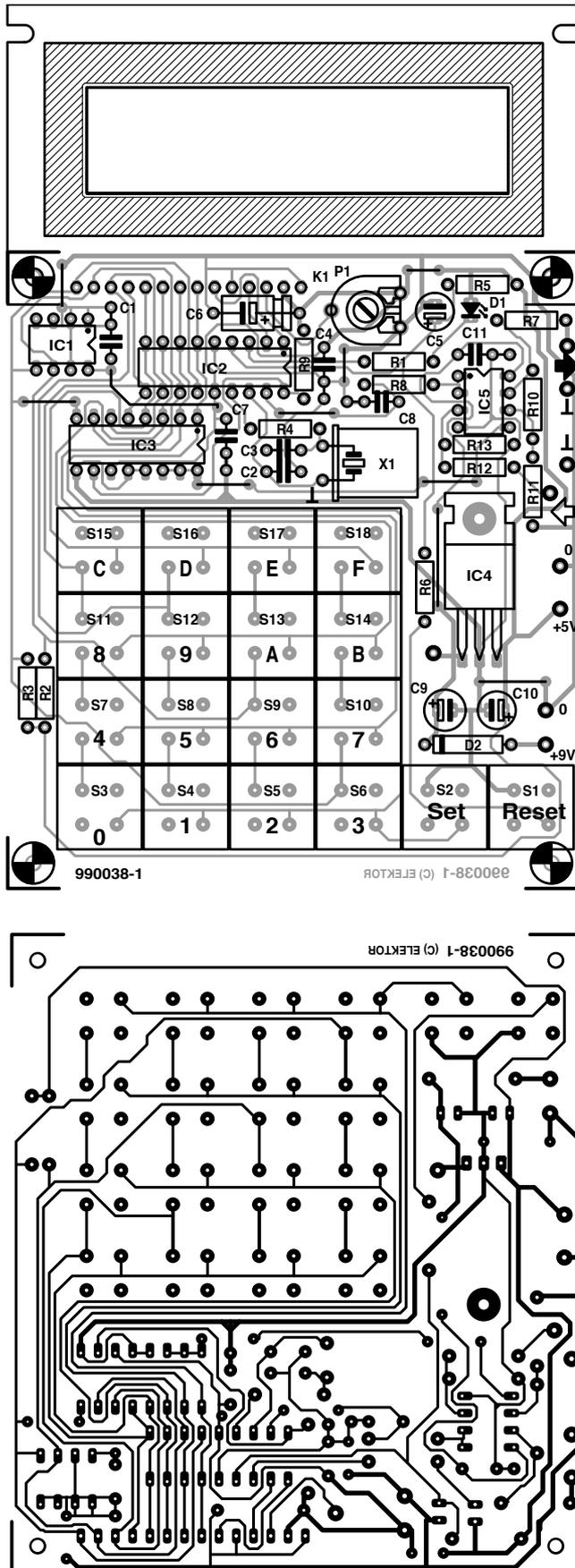


Figure 5. Disposition et plan de montage de la platine.

problème fondamental car les registres utilisés par exemple pour le calcul ne le sont jamais pendant l'introduction des paramètres. Il est par contre peu élégant que les noms des registres ne reflètent

pas toujours exactement leur rôle. Pour qui veut examiner le programme de plus près ou le modifier, il existe une disquette contenant le code source (disponible auprès de sources habituelles).

Liste des composants

Résistances :

R1 = 1k Ω
 R2,R3,R5,R9,R12 = 10 k Ω
 R4 = 270 Ω
 R6 = 100 Ω
 R7 = 10 Ω
 R8 = 1 k Ω
 R10 = 9k Ω 53 1%
 R11 = 10 k Ω /1%
 R13 = 1 M Ω
 P1 = 10 k Ω

Condensateurs :

C1,C4,C7,C11 = 100 nF céramique
 C2,C3 = 22 pF céramique
 C5,C10 = 10 μ F/10 V vertical
 C6 = 10 μ F/10 V
 C8 = 10 nF MKT
 C9 = 100 μ F/25 V vertical

Semi-conducteurs :

D1 = LED rouge à haut rendement
 D2 = 1N4001
 IC1 = 24C01
 IC2 = PIC16C71(programmé :EPS 996504-1)
 IC3 = AD557JN (Analog Devices)
 IC4 = 7805
 IC5 = TLC272 CP

Divers :

X1 = quartz 4 MHz
 K1 = affichage LCD à 2 rangées de 16 caractères tel que, par exemple, LM16A211 (Sharp)
 S1 à S18 = bouton-poussoir
 D6-R-RD (ITC)

CONSTRUCTION ET UTILISATION

Le montage du régulateur sur la platine ne requiert aucune dextérité particulière. Comme il s'agit d'une platine simple face, il faut parfois recourir à des fils de liaison. Ne pas oublier de les monter !

Les connexions internes ne permettent d'utiliser que le type de boutons poussoirs d'introduction indiqué dans la liste des pièces détachées. Un adaptateur secteur 9 V sans régulation suffit comme alimentation pourvu qu'il puisse fournir environ 200 mA au minimum. Pour mettre la platine de régulation en service, régler tout d'abord le contraste de l'affichage avec P1 pour obtenir la meilleure lisibilité possible.

Le circuit simple proposé ici (figure 7) permet de tester le régulateur et de « bidouiller » en simulant le processus asservi. La réponse du processus à un échelon est déterminée par les constantes de temps des trois filtres passe-bas RC couplés en série. Il est facile de modifier les constantes de temps de ces processus « électroniques » et aussi de simuler les perturbations en raccordant la résistance de charge à la sortie.

L'entrée du circuit de simulation est raccordé à la sortie du régulateur, la

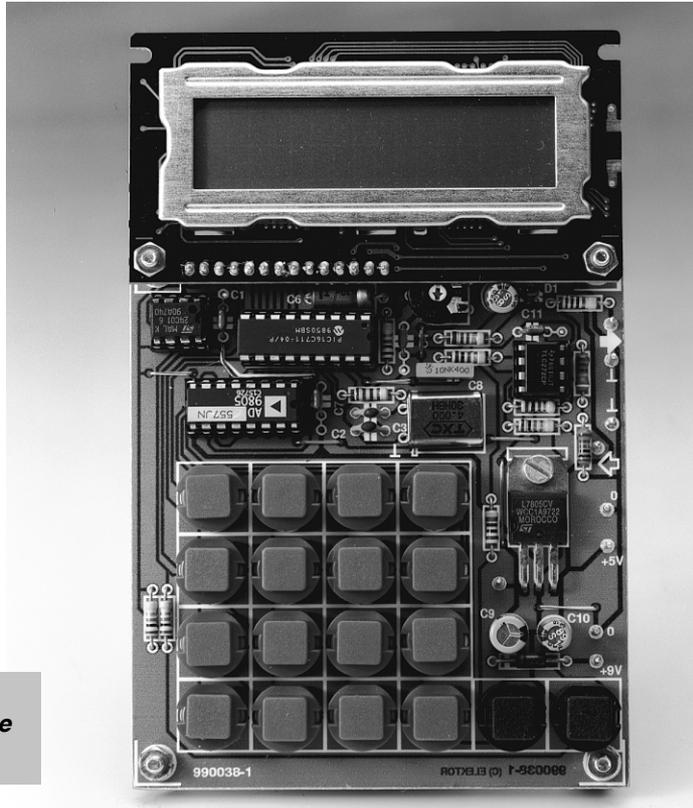


Figure 6. Platine de régulateur numérique prête à l'emploi.

sortie du circuit de simulation à l'entrée du régulateur. Il n'est possible d'introduire les paramètres qu'immédiatement après la mise en marche du régulateur ou après sa réinitialisation. Le clavier sert à introduire les paramètres sous forme hexadécimale. Il faut presser deux touches avant que les deux chiffres correspondants soient affichés. Les données introduites ne sont acceptées et stockées dans l'EEPROM que lorsque la touche SET a été pressée. La valeur de consigne et la grandeur réglante peuvent être positionnées entre 00 et FFh, ce qui correspond à une plage de tension de 0 V à 5 V. Il est aussi possible de choisir la limite supérieure et la limite inférieure entre 00 et FFh. L'intervalle d'échantillonnage (affichage « INTERVAL ») se compose d'un intervalle de base de 0,1 s multiplié par la valeur hexadécimale introduite.

Figure 7. Ce petit circuit sert à tester le régulateur en simulant le « processus » asservi.

Les paramètres q_0 , q_1 et q_2 , qui forment une partie importante de l'algorithme de régulation, appa-

raissent sous la forme de nombres à virgule fixe d'un octet (deux chiffres HEX) avant et d'un autre après la virgule. Le poids des bits après la virgule est de :

$1/2^n$, où n est la position du bit (MSB = 1, LSB = 8). Le nombre HEX 80 après la virgule exprime donc $1/2 = 0,5$ décimal. Comme q_1 est toujours négatif, on récapitulera brièvement ici la représentation des nombres binaires négatifs : un nombre binaire négatif est formé par l'inversion des bits suivie de l'addition de 1. Exemple :

$$\begin{aligned} 13d &= 0000\ 1101b = 0Dh \\ \text{inverse} &: 1111\ 0010b = F2h \\ + 1 &: 1111\ 0011b = F3h \end{aligned}$$

Cela signifie que les valeurs des nombres de 8 bits précédés d'un signe s'étendent de $+127d = 7Fh$ à $-128d = 80h$. On reconnaît les nombres de 8 bits négatifs à ce que le MSB = 1.

Lorsqu'on introduit q_0 à q_2 , il faut introduire séparément l'octet entier et l'octet fractionnaire (par exemple Q0H1 et Q0LO, au moyen de deux touches chacun, comme pour l'intro-

duction des autres quantités numériques, et terminer chacun des deux octets par un SET).

On a obtenu des résultats satisfaisants avec le circuit de simulation de la figure 7 en utilisant les paramètres suivants :

$$q_0 = 13,5d = 0D80h$$

$$q_1 = -18,5d = ED80h$$

$$q_2 = 6,5d = 0680h$$

Intervalle d'échantillonnage = 0,4 s (introduire : 04)

Valeur de consigne = 80h (= 2,5 V environ)

Grandeur réglante initiale = A0h (= 3,1 V environ)

Comme les tolérances des condensateurs électrolytiques du circuit de simulation ne sont pas des plus rigoureuses, il peut arriver que la régulation ne fonctionne pas encore parfaitement avec ces paramètres. Il faut alors ajuster empiriquement q_0 à q_2 .

La sortie du régulateur se trouve à zéro volts lors de l'introduction des paramètres. À la fin de l'introduction des paramètres confirmée par SET, la valeur initiale choisie pour la grandeur réglante (MAN.VAR. = *MANipulated VARiable*) est appliquée jusqu'à ce que la valeur de consigne (SETVALUE) soit atteinte. Le processus de régulation proprement dit entre alors en action et maintient la valeur de consigne même en présence de perturbations avec plus ou moins de succès selon les paramètres de régulation adoptés. La valeur instantanée de la grandeur réglante (MAN.VAR.) et de la valeur effective (ACT.VAL. = *ACTUAL VALUE*) sont continuellement affichées pendant la régulation.

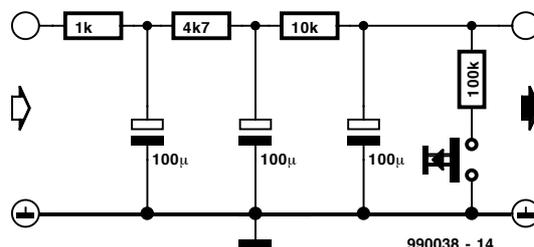
Si la limite supérieure (UPPER LIMIT) est dépassée ou la limite inférieure (LOWER LIMIT) n'est pas atteinte, la LED d'alarme D1 clignote aussi longtemps que dure l'écart en plus ou en moins. Pendant ce temps, les nouvelles données ne sont pas affichées. La sortie du régulateur devient nulle quand la limite supérieure est dépassée et FFh = 5 volts quand la limite inférieure n'est pas atteinte.

Si le régulateur est remis en marche par Reset, il faut effectuer la procédure d'introduction en confirmant chaque nouvelle valeur affichée par SET. L'afficheur contient tout d'abord SETVALUE avec la dernière valeur mémorisée. Suivent (en pressant SET) MAN.VAR., UPPER LIMIT, LOWER LIMIT et INTERVAL. Chaque donnée mémorisée est affichée et peut être modifiée si nécessaire.

(990038)

Référence :

Hoffmann, N. : *Digitale Regelung mit Mikroprozessoren*, Vieweg Verlag 1983



Bases d'un régulateur P.I.D numérique

L'approche la plus simple consiste à prendre comme base la structure connue d'un régulateur P.I.D. analogique en mode continu pour ensuite en dériver une représentation discrète. En domaine continu l'algorithme de régulation du régulateur P.I.D. répond à l'équation différentielle suivante :

$$u(t) = K_{PR} \cdot e(t) + K_{DR} \cdot de(t)/dt + K_{IR} \cdot \int_0^t e(\tau) d\tau$$

Dans la présente équation les symboles K_{PR} , K_{DR} et K_{IR} représentent le gain (ajustable) des 3 régulateurs (P, D et I respectivement) de la figure 3. Il est possible, par des transformées de Laplace de rendre cette équation pour le domaine des fréquences :

$$u(s) = K_{PR} \cdot e(s) + K_{DR} \cdot s \cdot e(s) + (K_{ID} / s) e(s)$$

formule qui suppose que les conditions de départ soient nulles. L'ingénieur de développement aura à définir les gains des 3 régulateurs partiels (souvent baptisés valeurs partielles de proportionnalité, d'intégration et de différentiation du régulateur P.I.D.) en fonction du comportement de régulation souhaité.

L'étape suivante consiste à rechercher une représentation discrète du régulateur décrit par les équations [1] et [2]. La différence rétroactrice est définie comme étant l'équivalent temporel discret de la représentation chronologique en continu d'une fonc-

tion. Elle se traduit par :

$$\Delta f(t) = [f(t) - f(t - T)] / T$$

formule dans laquelle T est la période du signal d'échantillonnage (= intervalle d'échantillonnage).

La valeur de somme est définie comme l'équivalent chronologique discret de l'intégrale temporelle continue d'une fonction. Elle s'obtient par la formule suivante :

$$\int_a^b f(\tau) = T [f(a + T) + f(a + 2T) + \dots + f(b)]$$

On obtient, si l'on applique les équations données en [3] et [4] à celles des 2 premières formules, l'équation de l'algorithme de régulation du régulateur P.I.D. numérique donnée dans l'article :

$$u_k = u_{(k-1)} + q_0 \cdot e_k + q_1 \cdot e_{(k-1)} + q_2 \cdot e_{(k-2)}$$

L'indice k est à nouveau le numéro de séquence des différentes mesures :
le facteur $e_{(k-2)}$ représente lui la déviation de réglage de la mesure précédente.

L'HISTOIRE DE L'ÉLECTRONIQUE (7)

Ce n'est peut-être qu'une illusion, mais il semble, qu'à la fin du 19^{ème} siècle, les découvertes se suivent à intervalle de plus en plus rapproché.

En 1875, Alexander Graham Bell découvrit le téléphone. Il utilisa, pour la capture du son, un capteur (micro) de forme plus ou moins conique. Tout près de la fine membrane métallique montée à l'extrémité de l'écouteur se trouvait un barreau magnétique sur lequel était montée une petite bobine. Lorsque l'on parlait dans le capteur la membrane entraînait en oscillation de sorte que la distance entre la membrane et le barreau magnétique ne cessait de varier.



**Alexander Graham Bell
(1847 - 1922)**

On avait de ce fait induction, dans la petite bobine, d'une tension alternative qui variait en fonction du rythme des mots émis. Un correspondant pouvait, par la prise en parallèle de 2 de ces appareils interconnectés par un long câble, faire vibrer la membrane de l'autre appareil. On parlait dans, et on écoutait aussi, le même « appareil » que l'on mettait alternativement devant la bouche (parole) et contre l'oreille (écoute). Le prototype fut rapidement amélioré par l'adjonction d'un écouteur distinct. Dès 1878 on ouvrit aux États-Unis le premier réseau de téléphone public. La découverte de Bell est un indiscutable repère dans le domaine de la communication à longue distance.

Une découverte pour le moins aussi révolutionnaire vint, en 1877, le jour dans le laboratoire de Thomas Alva Edison. Il s'agissait d'un appareil servant à enregistrer mécaniquement le son. Ledit appareil prenait la forme d'un rouleau de bois tourné manuellement, pièce sur laquelle était monté une feuille d'étain de faible épaisseur. À proximité immédiate du rouleau on avait disposé une membrane dotée d'une pointe métallique qui appuyait légèrement sur la feuille d'étain. Les ondes sonores faisaient

entrer la membrane en oscillation de sorte que la pointe gravait un sillon en forme de spirale dans la feuille d'étain, piste dont la profondeur variait en fonction du rythme de la modulation. Lors de la reproduction le sillon ainsi gravé faisait, à son tour, entrer la membrane en oscillation.

Bien que la qualité de reproduction de ce « phonographe » ne soit pas des meilleures, le système était parfaitement fonctionnel. En 1888 le rouleau recouvert d'une feuille d'étain fut remplacé par un rouleau recouvert de cire et l'entraînement du moteur devint électrique; ces 2 modifications améliorèrent très sensiblement la qualité. Plus tard, Emile Berliner eut l'idée de ne pas moduler le sillon en profondeur mais en largeur. En 1888 Berliner remplaça le rouleau par un disque métallique dans lequel le sillon avait été écrit par un processus de gravure. Le disque microsillon qui fut son apparition plus tard n'est en fait qu'un développement de cette approche. Évoquons, en guise de conclusion, un événement important qui a pris place à la même époque. Vu la faible durée de vie de leurs filaments, les ampoules à incandescence restaient une sorte de « jouet ». En 1879, Josef Wilson Swan démontra une ampoule à incandescence dont le ballon avait été vidé de son air ce qui empêchait la combustion du filament. Malheureusement pour lui il s'avéra que, quelques mois plus tôt, Edison avait déposé un brevet concernant une ampoule aux caractéristiques similaires. En 1882, la société « Edison Lamp Company » lança la production en série d'ampoules à incandescence dotées, en guise de filament, d'un morceau de bambou japonais. Ceci n'empêcha pas Swan de poursuivre le développement de sa propre ampoule à incandescence. Il obtint, en 1880, un brevet anglais pour un exemplaire doté, en guise de filament, d'un morceau de fil de coton carbonisé. Aux Pays-Bas non plus on ne resta pas les bras croisés; 1884 vit la naissance, dans ce pays, de la première fabrique d'ampoules à incandescence, la société « Electriciteits Maatschappij Systeem de Khotinskijj ». (995073)



Phonographe antique

potentiomètre numérique

la puce Xicor ne manque pas d'R

Même si le (micro)processeur accroît son emprise sur l'électronique, il reste des domaines où l'analogique conserve la préférence. Le potentiomètre en est un bon exemple. Cependant, comme composant numérique, celui-ci n'a pas son pareil. Les potentiomètres numériques Xicor X9440 et X9448 allient la souplesse de l'analogique à la puissance de la communication de données (numériques).

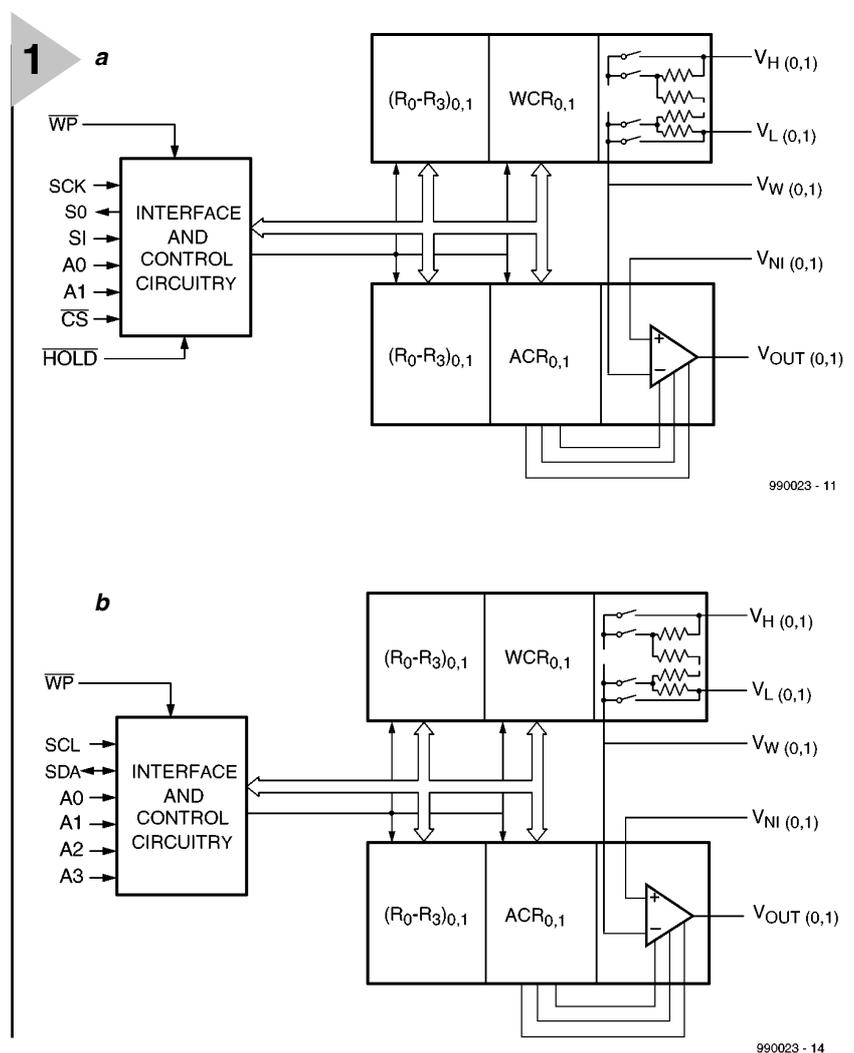


Figure 1. Structure interne du X9440 et du X9448. Les différences se situent au niveau de l'interface sérielle uniquement.

Les X9440 et X9448 sont des potentiomètres doubles, à commande numérique (EEPOT), équipés de deux comparateurs de tension. Les puces se fabriquent en technologie CMOS et disposent, l'une de l'interface SPI (Serial Programming Interface = interface de programmation sérielle), l'autre

d'une I²C. La figure 1 reprend le diagramme fonctionnel du potentiomètre double, basé sur deux bancs de chacun 63 résistances en série. Tous les noeuds peuvent accéder à la sortie qui correspond au curseur, si bien que 64 prises (désignées par 6 bits) sont ainsi disponibles. Le curseur est en réa-

lité le rail commun d'une batterie d'interrupteurs et c'est par logiciel que sera déterminée sa position sur l'échelle. Sur le X9440, les commandes de dépla-

cement du curseur passent par une interface sérielle SPI alors que le X9448 se sert du bus I²C. Voilà pourquoi l'un et l'autre excellent à communiquer avec

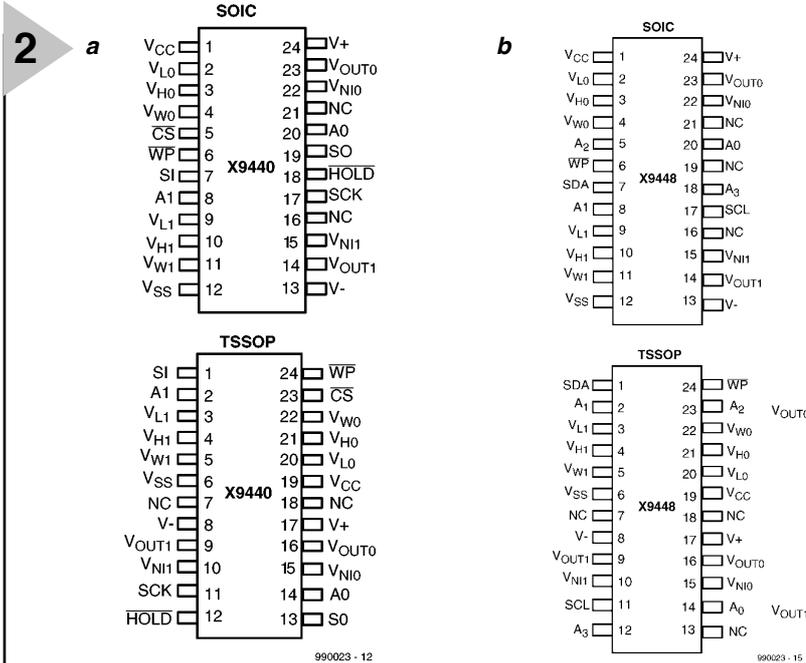
un microcontrôleur.

La valeur nominale de résistance du potentiomètre est, au choix, de 10 k Ω ou 2,5 k Ω . Chaque potentiomètre est en outre doté d'un circuit qui compare la tension V_W présente sur le curseur à V_{NI} , la tension d'entrée.

Deux registres sont prévus pour assurer la commande du fonctionnement du potentiomètre et du comparateur de tension : on trouve ainsi WCR, *Wiper Counter Register* pour positionner le curseur et ACR, *Analog Control Register* pour le comparateur. Mais il y a encore huit autres registres de 6 bits que l'on peut utiliser à sa guise pour inscrire des données et les relire.

Le contenu de WCR détermine la position du curseur, celui de ACR définit le comportement du comparateur et de son signal de sortie.

Figure 2. Brochage des 2 circuits intégrés décrits dans le présent article. Il existe, pour chacun d'eux, une version en boîtier SOIC et une autre en TSSOP.



Le brochage des potentiomètres

$V_H(V_{H0}-V_{H1})$, $V_L(V_{L0}-V_{L1})$

Les broches V_H et V_L s'identifient aux extrémités de la piste d'un potentiomètre classique.

$V_W(V_{W0}-V_{W1})$

Cette sortie est similaire au curseur d'un potentiomètre ordinaire. Elle est branchée à l'entrée inverseuse du comparateur de tension.

Comparateurs et autres raccordements

Tensions d'entrée V_{NI0} , V_{NI1}

V_{NI0} et V_{NI1} sont les entrées homophases (non inverseuses) des deux comparateurs.

Sorties tamponnées, V_{OUT0} et V_{OUT1}

V_{OUT0} et V_{OUT1} sont deux sorties où se manifestent les signaux issus des comparateurs, après remise en forme. Ces sorties sont activées par le bit de commande correspondant de l'ACR.

Entrée de protection matérielle en écriture

Un niveau bas sur la broche WP interdit toute modification de l'ACR et du WCR.

Tensions analogiques V_+ en V_-

Les tensions V_+ en V_- constituent les alimentations de la partie analogique, celle qui comprend les potentiomètres électroniques et les comparateurs.

V_{CC} et V_{SS}

V_{CC} et V_{SS} sont les tensions d'alimentation des fonctions d'interface et de commande.

L'interface du X9440

Sortie sérielle (SO)

SO est une sortie de données sérielle symétrique, configurée en *push pull*. Lors d'un cycle de lecture, la donnée devient accessible au flanc négatif du signal d'horloge.

Entrée sérielle (SI)

SI est l'entrée sérielle de données. Tous les codes d'opération, les octets d'adresse et les données transitent par cette broche. Les informations sont valides au flanc montant du signal d'horloge.

Horloge sérielle (SCK)

Cette broche scande le transfert d'information, en entrée comme en sortie.

Validation de puce (Chip Select)

Aussi longtemps que cette broche porte un niveau haut, le X9440 est inactif, SO est à haute impédance et le circuit intégré en veille. Dès qu'elle passe au niveau bas, la puce s'active.

Maintien (Hold)

Hold, de concert avec \overline{CS} , s'emploie pour valider la puce. Dès qu'elle est activée, la transmission sérielle commence. Hold permet aussi d'introduire une pause dans la communication sans la couper effectivement. Il suffit pour cela de passer Hold au niveau bas pendant que SCK est bas. Si l'on n'utilise pas cette fonction, Hold doit rester au niveau haut.

Lignes d'adresse (A0 et A1)

Ces entrées d'adresse concernent les deux bits de poids fort de l'octet d'adresse. Seuls quatre composants peuvent se servir d'une même interface SPI.

L'interface du X9448

Horloge sérielle (SCL)

Cette ligne scande le transfert d'information, en entrée comme en sortie.

Donnée sérielle (SDA)

SDA est une connexion bidirectionnelle, elle sert aussi bien d'entrée que de sortie.

Lignes d'adresse (A0 à A3)

Ces entrées d'adresse définissent les quatre bits de poids fort de l'octet d'adresse. Il en découle que seize composants peuvent disposer en commun du bus bifilaire I²C.

LE RÉSEAU DE RÉISTANCES

Les puces comptent deux réseaux qui totalisent chacun 63 résistances montées en série, chaînes dont les extrémités sont comparables aux bornes d'un potentiomètre classique, V_H et V_L . On y a branché, ainsi qu'en chacun des noeuds communs aux résistances des réseaux, un interrupteur capable de relier n'importe lequel de ces 64 points au rail commun de sortie. Ces interrupteurs sont commandés par le *Wiper Counter Register (WCR)*. Le code qu'il contient, sur six bits, désigne, après décodage, l'interrupteur à mettre en action.

Il est possible d'écrire directement dans le WCR, mais on peut aussi lire le contenu d'un des quatre registres qui lui sont associés et le copier dans le WCR. Les registres de données aussi bien que le WCR peuvent être lus ou modifiés par un ordinateur ou un microprocesseur via l'interface sérielle.

DRAPEAUX ET REGISTRES

Le comparateur mesure l'écart de tension entre le curseur (V_W) et l'entrée (V_{ND}). La logique qui lui est associée répond aux bits de commande *Shutdown* (couper), *Latch* (memoriser) et *Enable* (valider) présents dans le registre de commande analogique ACR. *Enable* établit la liaison entre sortie du comparateur et borne V_{OUT} . *Latch* retient la combinaison logique de sortie et *Shutdown* coupe l'alimentation de la partie analogique, réduisant du même coup la consommation.

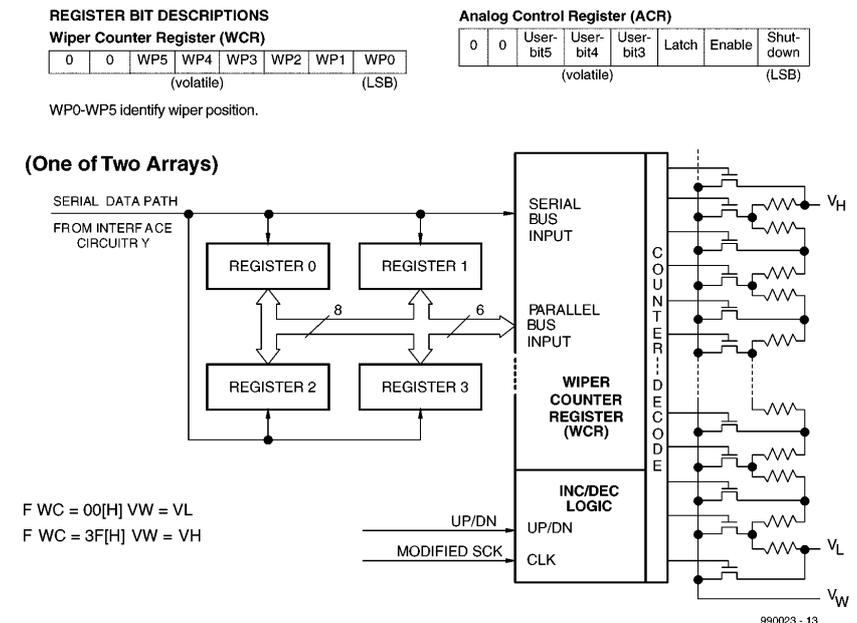
Le registre ACR, comme ses congénères, se programme par l'interface sérielle.

Les deux potentiomètres et les comparateurs de tension font usage d'interfaces standardisées pour les communications avec le processeur. Mais chacun d'eux peut aussi entrer en liaison avec les registres WCR, ACR et ceux de données. Il s'agit là d'une partie importante de la puce que nous allons analyser à présent.

LES REGISTRES WCR ET ACR

Le X9440, comme le X9448, contient deux WCR, un par potentiomètre électronique, et, comme on le voit à la figure 3, deux ACR, puisqu'il y a deux comparateurs à commander. Le WCR, on peut le comparer à un convertisseur à entrée sérielle et sortie parallèle, laquelle régit la sélection du commutateur choisi parmi les soixante-quatre de l'échelle potentiométrique.

Il est possible de modifier le contenu des deux registres de quatre manières différentes. Le processeur central peut y imprimer la valeur à l'aide de l'instruction *Write WCR* (transfert sériel), mais on peut aussi la charger indirectement



tement par l'intermédiaire d'un des registres de données (DR) associés. C'est le rôle de l'instruction *XFR Data Register*, qui transfère l'information en parallèle. Et puis il est aussi loisible de modifier le WCR, et lui seul, par décrémentation ou incrémentation. Finalement, on peut encore, à l'allumage, y charger le contenu du registre 0.

Il convient en effet de se rappeler que WCR et ACR sont des mémoires volatiles, qui oublient les données lors de toute coupure de tension. Mais à la mise en marche, leur contenu n'est pas indéterminé, parce qu'ils copient celui du registre de données R0. La programmation de l'ACR et du WCR est identique. Encore faut-il bien saisir que les six bits du WCR déterminent la position du curseur, tandis que les trois bits de l'ACR influencent le fonctionnement et la sortie du comparateur.

LES REGISTRES DE DONNÉES

Chaque potentiomètre et chaque comparateur dispose de quatre registres de données non volatiles. De quoi se faire un pense-bête, puisque les données qui y sont consignées peuvent passer directement dans l'ACR ou le WCR. L'enregistrement dans les registres de données est assez lent, quelque 10 ms, du fait qu'il s'agit de mémoires non volatiles.

Si, dans une application donnée, ces registres ne sont pas utiles pour y sauvegarder des configurations de rechange, ils peuvent trouver à s'employer au profit d'autres variables du système. La durée de rétention est garantie pour au moins cent ans et à

Figure 3. Lesdits circuits intégrés comportent 4 registres à usage général, un compteur (WCR) et une chaîne de résistances prises en série, résistances reliées à une borne commune au travers de transistors FET.

l'inverse, ils sont capables de subir plus de 100 000 modifications.

CONDENSÉ DU BROCHAGE

La figure 2 vous révèle la fonction de chacune des broches des circuits intégrés. Tous deux sont disponibles en boîtier CMS modifiés (SOIC et TSSOP) exclusivement. Les différences entre X9440 et X9448 se limitent aux fonctions de certaines broches. On peut les répartir en trois catégories : celles qui assurent l'interface avec le processeur central, celles des potentiomètres et des comparateurs et celles d'alimentation. La différence réside dans les signaux d'interface, selon le cas, un bus SPI ou un I²C.

Les feuilles de caractéristiques des puces X9440 et X9448, vous pouvez les trouver en formant l'adresse du site : <http://www.xicor.com>.

(990023)

electronique en ligne

le transfo de Tesla produire ses propres eclairs

Nikola Tesla, né en 1856, ne s'est sans doute jamais imaginé, au cours de sa vie, que sa découverte jouirait quelque 100 ans plus tard d'un intérêt aussi marqué. Internet regorge de nombre de sites consacrés à ce qui a été appelé le transfo de Tesla.

L'idée est en fait extrêmement simple. On réalise un transformateur caractérisé par un rapport de spires très élevé entre le primaire et le secondaire et on fait en sorte que l'accord entre les 2 boucles soit tel qu'elles résonnent à la même fréquence.

Si, dans ces conditions, on applique une impulsion sur l'enroulement du primaire (par le biais, par exemple, d'une étincelle), la tension générée ainsi au secondaire atteint une valeur telle qu'il peut se produire des décharges importantes qui se manifestent sous la forme de flashes de lumière.

Nombreux sont aujourd'hui les amateurs d'électronique à trouver captivant de réaliser un tel transfo de Tesla et de l'utiliser pour produire de magnifiques pluies d'étincelles. L'esprit de compétition s'en mêlant, le but de la manoeuvre est de produire les éclairs les plus grands possible. On n'a pas encore battu le record de Tesla qui, en 1899, produisit, avec son système expérimental monté à Colorado Springs, un flash de pas moins de 41 mètres, qui se traduisit d'ailleurs par la mise hors-service de l'ensemble du réseau électrique de Colorado Springs !

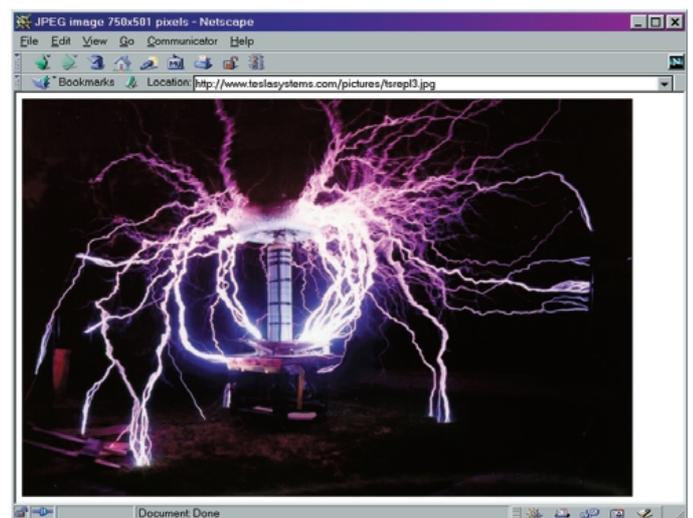
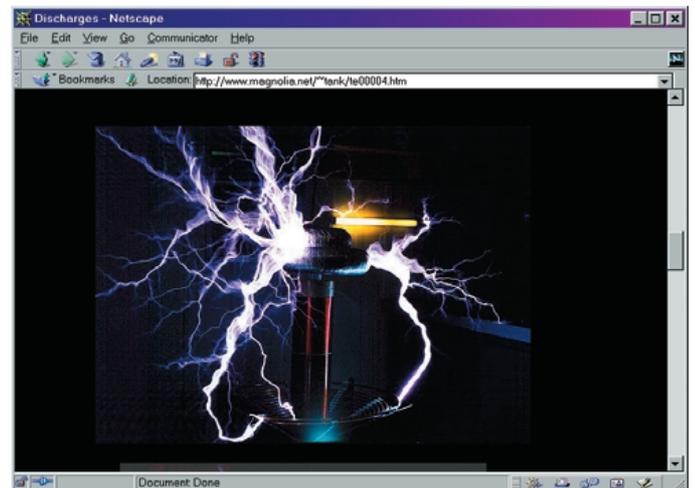
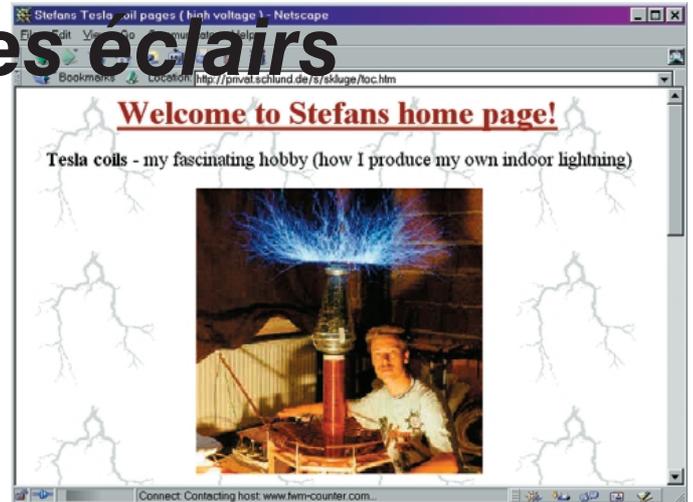
Le transfo de Tesla est devenu tellement populaire sur le Web que pas moins de 120 sites déjà sont associés au *Tesla Coil Webring* (<http://www.webring.org/cgi-bin/webring?ring=Tesla-Ring&list>); on peut également faire un tour sur <http://www.webring.org> et entrer « tesla » comme mot-repère pour le moteur de recherches.

Le site <http://www.pupman.com> fournit une longue liste de sites Tesla. Nous en avons extrait quelques-uns que pourrions visiter ceux d'entre nos lecteurs qui aimeraient en savoir plus sur ce phénomène. Les pages du site de Stefan,

adresse (<http://privat.schlund.de/s/skluge/toc.htm>) consacrées à Tesla donnent une description complète du concept d'un transfo de Tesla, et offrent nombre de tableaux, diagrammes et autres formules. Vous pourrez, si vous préférez une approche plus compréhensible, trouver votre bonheur sur les pages Tesla de Guy, à l'adresse : (<http://pages.vosnet.de/wilson/tesla1.htm>).

Le Web vous propose également l'un ou l'autre outil facilitant le calcul d'un transfo de Tesla, tel que la page *Tesla Coil Calculator* écrite en JavaScript (<http://www.geocities.com/CapeCanaveral/Hangar/3108/calculator.html>). Une fois les données requises entrées, le programme calcule tous les éléments. Une promenade sur les différents sites de ceux que passionne la réalisation d'un transfo de Tesla permet également de trouver des photos des transformateurs et des résultats qu'ils ont permis. 2 exemples intéressants

parmi bien d'autres : *Inonized Ether* (<http://www.magnolia.net/~tank/tesla.htm>) et la page *Tesla Coil* de Chuck Curran (<http://www.execpc.com/~ccurran/>). Il y a même, depuis peu, des sociétés qui se sont lancées dans la construction et/ou la démonstration de transfos de Tesla telle que, par exemple, Tesla System



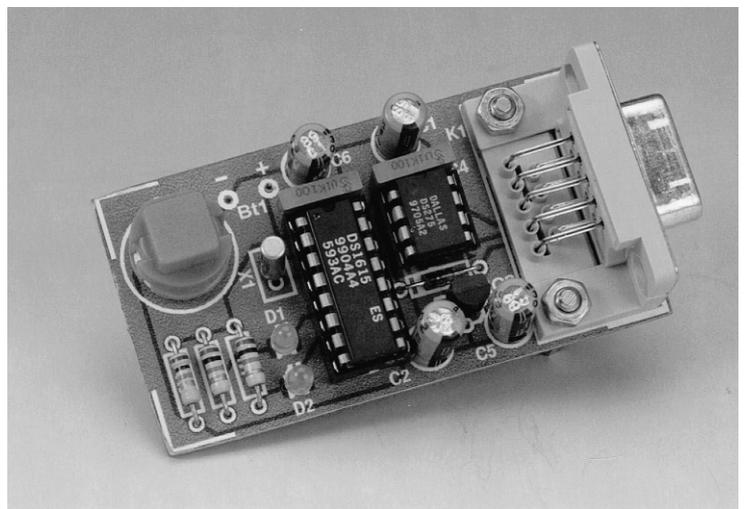
Research à l'adresse : <http://www.teslasystems.com/>. Très spectaculaire ! Si vous voulez en savoir plus au sujet de ce grand inventeur, vous pouvez faire un tour à l'une des adresses suivantes : <http://members.xoom.com/XOOM/tastraum/tesla.html> et <http://www.bena.com/lucidcafe/library/96jul/tesla.html>. (995070)

enregistreur de température DS1615

kit d'évaluation avec puce Dallas

Le programme de fabrication de l'entreprise Dallas Semiconductor comporte une série de capteurs qui, non contents de saisir les grandeurs de mesure, peuvent les stocker, les trier et les compter. Le nouvel enregistreur de température DS1615 que nous vous présentons constitue un

bon exemple. Montons le DS1615 sur une petite platine, ajoutons un logiciel (gratuit) pour PC et nous pourrons « bidouiller » à cœur joie.



Le DS1615 est un enregistreur de température intégré qui peut aussi bien servir d'enregistreur de données (température en fonction du temps) que créer des histogrammes (nombre des valeurs de température mesurées) entre -40 et $+85$ °C.

L'horloge en temps réel intégrée (RTC) compte les secondes, minutes, heures, jours, mois, jours de la semaine et années en format BCD. Cette horloge en temps réel tient compte des années bissextiles et est prête à affronter le prochain millénaire.

Il est possible de programmer l'enregistreur de température pour un terme d'alarme particulier au cours d'une semaine. Le DS1615 produit une interruption lorsque le délai vient à échéance.

Ce circuit intégré permet de programmer des intervalles de mesure de 1 mn à 255 mn. Ceux-ci valent pour les deux modes. La mémoire non volatile des valeurs de mesure peut en contenir 2 048, ci-inclus l'heure de départ de la

fonction enregistreur ou les 63 valeurs de température à 2°C d'intervalle lorsque le CIRCUIT INTÉGRÉ doit établir un histogramme. Chacun de ces 63 compteurs peut contenir 25 535 mesures. Les compteurs ne bouclent pas une fois le maximum atteint.

Il est en outre possible de positionner un seuil minimum et un seuil maximum de température qui produiront une alarme (autrement dit : une interruption ou le changement d'état d'une broche).

La liaison avec un PC ou un microcontrôleur est effectuée par interface série en mode synchrone ou asynchrone. Il n'est pas indispensable que le DS1615 reçoive l'ordre de commencer les mesures par cette interface série ; on parvient au même résultat en pressant un bouton.

TOPOGRAPHIE MÉMOIRE

Vu la complexité de ce circuit intégré, il est impossible d'en décrire toutes les fonctions en détail en quelques pages.

Pour qui envisage sérieusement de s'attaquer au DS1615, il est toujours possible de se référer aux 20 pages de la fiche de données www.dalsemi.com/DocControl/PDFs/1615.pdf.

Le schéma fonctionnel de la **figure 1** indique comment est conçu l'enregistreur de température. Les liaisons avec le monde extérieur passent par la logique de commande et l'interface sérielle (sans oublier le capteur). La cadence de 32,768 kHz est donnée par une horloge à quartz.

Le reste est l'objet du chapitre « mémoire », celle-ci étant segmentée en pages de 32 octets. Les pages 0 et 1 contiennent l'horloge en temps réel et les registres de commande, la RAM non volatile de l'utilisateuse se trouve à la page 2. La page 16 recèle le numéro individuel de série, les pages 17 à 19 sont dévolues aux repères de temps des alarmes et à leur durée. Les pages 64 à 67 sont réservées à l'histogramme de température, les pages de 128 à 191 à la mémoire de l'enregistreur de température. Les pages passées sous silence sont prévues pour des applications futures. Comme dans un bon quotidien, les faits les plus importants se trouvent en première page (*RTC and Control Page*, **figure 2**), et c'est sur celle-ci que nous nous baserons pour expliquer les divers modes de fonctionnement et le jeu de commandes.

SAISIE DE TEMPÉRATURE

Le capteur de température mesure la température dans la plage -40 à + 85 °C. La résolution est de 0,5 °C, l'erreur maximale sur la température de ± 2 °C. La valeur de la température est exprimée par un octet de 0000000_b pour -40 °C à 11111010_b pour + 85 °C. La température, calculée par la formule $T [^{\circ}\text{C}] = 0,5 (\text{octet}) - 40$ °C

est placée dans le *Current Temperature Register* situé à l'adresse 11 sur la première page mémoire où elle est disponible pour tout traitement supplémentaire ou pour un affichage direct hors de la période d'acquisition des données. Les données de température sont valides lorsque le bit *Temperature Ready (TR)* est à 1 ; ce bit est à zéro lors du calcul d'une valeur de mesure et de son écriture en mémoire.

ENREGISTREMENT DES DONNÉES

En mode *Data logging*, les températures détectées sont reportées successivement à partir de l'adresse 1000_h, jusqu'à ce que 2 048 registres soient remplis.

Comme nous l'avons déjà mentionné, le lancement d'une série de mesures est effectué à partir du PC par l'interface sérielle ou en pressant un bouton de l'enregistreur de température. Dans le premier cas, il faut que le bit SE du registre d'instruction soit à zéro pour que la broche d'entrée *ST* soit bloquée.

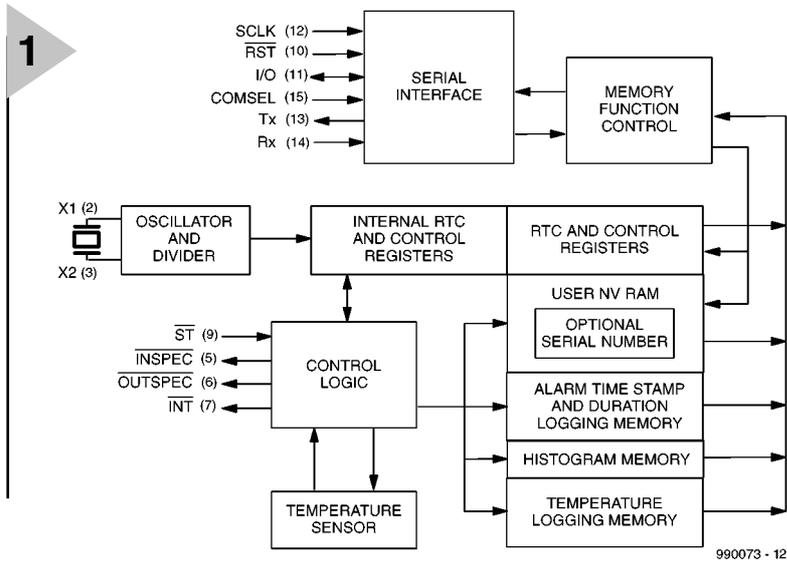


Figure 1. Schéma fonctionnel de l'enregistreur de température DS1615.

Figure 2. RTC et Control Page.

DS1615 RTC AND CONTROL PAGE

ADDR.	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION
00	0	10 Seconds			Single Seconds				Real Time Clock Registers
01	0	10 Minutes			Single Minutes				
02	0	12/24	10 h A/P	10 h	Single Hours				
03	0	0	0	0	0	Day Of Week			
04	0	0	10 Date		Single Date				
05	0	0	0	10 m.	Single Months				
06	10 Years			Single Years					
07	MS	10 Seconds Alarm			Single Seconds Alarm				Real Time Clock Alarm
08	MM	10 Minutes Alarm			Single Minutes Alarm				
09	MH	12/24	10 ha. A/P	10 h. alm.	Single Hours Alarm				
0A	MD	0	0	0	0	Day Of Week Alarm			
0B	Low Temperature Threshold								Temperature Alarm
0C	High Temperature Threshold								
0D	Number Of Minutes Between Temperature Conversions								Sample Rate
0E	EOSC	CLR	0	SE	RO	TLIE	THIE	AIE	Control
0F	(reads 00h)								Reserved
10	(reads 00h)								Reserved
11	Current Temperature								Temperature
12	Start Delay Register (LSB)								Start Delay
13	Start Delay Register (MSB)								Start Delay
14	TR	MEM CLR	MIP	SIP	LOBAT	TLF	THF	ALMF	Status
15	Minutes								Start Time Stamp
16	Hours								
17	Date								
18	Month								
19	Year								
1A	Low Byte								Current Samples Counter
1B	Medium Byte								
1C	High Byte								
1D	Low Byte								Total Samples Counter
1E	Medium Byte								
1F	High Byte								
20-3F	(Read 00H)								Reserved

990073 - 13

3

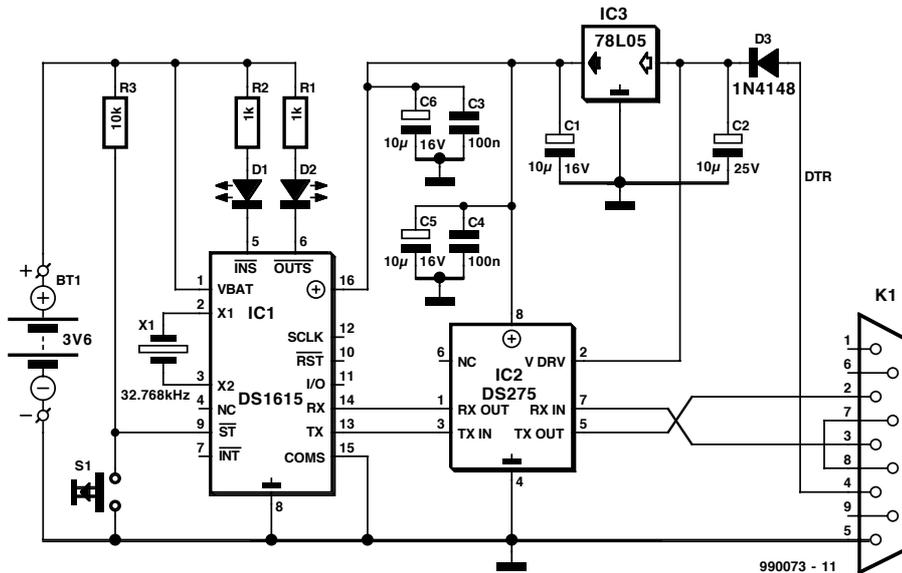


Figure 3. Matériel du kit d'évaluation DS1615.

La première valeur différente de zéro écrite dans le registre de taux d'échantillonnage (*sample rate*) met la mesure en marche. Mais si l'on met le bit SE au niveau logique haut (1) et que l'on écrit une valeur différente de zéro dans le registre de taux d'échantillonnage, il faut par contre forcer la connexion *ST* (par un bouton) au niveau bas (0) pendant au moins 0,5 s pour faire démarrer l'enregistreur.

Quelles que soient les formalités de départ, les sorties *INSPEC* et *OUTS-PEC* engendrent simultanément 4 impulsions de niveau bas au début d'une série de mesures. En outre, le bit dénommé *MIP (Mission (sic!) In-Progress)* du registre d'état est positionné (mis au niveau logique 1). Pour mettre fin aux mesures, il suffit d'écrire zéro dans le bit *MIP*.

L'intervalle qui sépare le début des mesures de la première saisie de température est contenu dans le *Start Delay Register* (adresses 12 et 13). Ce registre est synchronisé chaque minute par le

registre des secondes de l'horloge en temps réel.

Il existe deux possibilités de venir à bout du dépassement de capacité du compteur de valeurs de mesure (c'est-à-dire plus de 2 048 mesures). On peut faire appel au retour au point de départ (*Rollover Feature*) en positionnant le bit 3 du registre d'instruction à 1. Les nouvelles données écrasent alors les anciennes à partir de l'adresse 1000_h. La deuxième solution consiste à arrêter la saisie des données lorsque le nombre de 2 048 est atteint. On MET le bit *Rollover RO* à zéro.

Le DS1615 enregistre bien l'état de l'horloge en temps réel lors de la première mesure, mais les repères de temps des mesures suivantes doivent être déterminés à partir de la valeur du compteur *Current Sample* et de l'adresse d'échantillonnage. Cela ne constitue pas un problème si le dépassement de

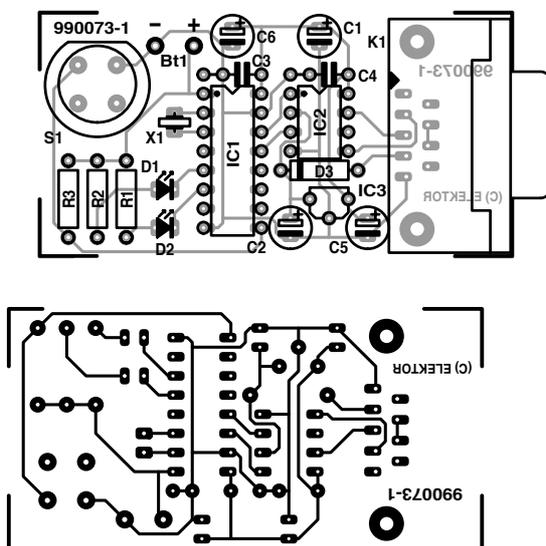
capacité est inactif, mais en mode *rollover* l'utilisateur doit aussi tenir compte du registre *Session Sample*. La valeur de celui-ci divisée par 07FF_h, donne le nombre de dépassements de capacité. Pour des raisons de sécurité, il n'existe aucune possibilité d'écrire dans la mémoire des mesures de température.

HISTOGRAMME

Le DS1615 calcule aussi l'histogramme des températures mesurées et le place dans les 4 pages *Temperature Histogram Memory* (adresses 0800_h à 087F_h). Ces pages contiennent 63 compteurs de 16 bits chacun, qui peuvent donc compter jusqu'à 65 535. Chaque compteur est incrémenté de 1 lorsque la température mesurée se trouve dans sa plage. Le compteur 0 est associé aux températures de -40 °C à -38,5 °C (valeurs 0_b, 1_b et 10_b), le compteur 1 à celle de -38 °C à -36,5 °C (valeurs 11_b, 100_b et 101_b), etc., jusqu'au compteur 62 qui contient le nombre de mesures de + 84 °C à + 85 °C (valeurs

Figure 4. Disposition et plan de montage de la platine.

4



Liste des composants

Résistances :

R1, R2 = 1 kΩ

R3 = 10 kΩ

Condensateurs :

C1, C2, C5, C6 = 10 μF/16 V

C3, C4 = 100 nF

Semi-conducteurs :

D1, D2 = LED à faible courant

D3 = 1N4148

IC1 = DS1615 (Dallas Semiconductor)

IC2 = DS275 (Dallas Semiconductor)

IC3 = 78L05

Divers :

X1 = quartz 32,768 kHz

BT1 = cellule au lithium 3V6

S1 = bouton-poussoir unipolaire à contact travail

11111000_b, 1111001_b et 1111010_b). Les compteurs restent par ailleurs au maximum de 65 535 mesures lorsque ce nombre est atteint.

NON CONFORME

Dans de nombreuses applications, il est important de savoir avec quelle fréquence et pour combien de temps une plage de température « permise » est dépassée. Les limites de la plage sont placées dans le registre *Temperature Alarm* (adresses 000B_h et 000C_h). Aussi longtemps que les mesures restent dans la plage, le DS1615 n'enregistre pas d'alarme de température. Sinon, le circuit intégré donne l'alarme en positionnant l'indicateur THF (*Temperature High Flag*) ou TLF (*Temperature Low Flag*) (TLF) du registre d'état (adresse 0014_h). L'enregistreur de température produit en outre un repère de temps qui signale le début de l'alarme et enregistre la durée du dépassement de la plage par le haut ou par le bas. Une interruption est engendrée à la broche **INT** lorsque le bit *Temperature High Interrupt Enable* (THIE) est positionné par un dépassement vers le haut ou le bit *Temperature Low Interrupt Enable* est positionné par un dépassement vers le bas.

Les repères de temps et la durée sont placés dans les domaines d'adressage 0220_h à 024F_h de 48 octets chacun dans le cas des dépassements vers le bas et dans ceux de 0250_h à 027F_h pour les dépassements vers le haut. Il est donc possible d'enregistrer 12 débuts et durées d'alarmes de dépassement vers le haut et le même nombre dans le cas de dépassements vers le bas. Le DS1615 détermine la date et l'heure des périodes de dépassement à partir du *Start Time Stamp* et de l'intervalle de mesure.

Les deux sorties *INSPEC* et *OUTSPEC* donnent des informations sur l'état du DS1615. Lorsque l'utilisateur lance une série de mesures, les deux sorties produisent simultanément 4 impulsions actives au niveau bas. Si l'on demande par contre des informations sur les températures mémorisées et si ces valeurs se trouvent dans la plage permise, la broche *INSPEC* envoie 4 impulsions basses. Si une valeur, ou plus, sort de la plage, la broche *OUTSPEC* produit des impulsions. Si la demande est effectuée après le début de la série de mesures mais avant la première mesure, les 4 impulsions apparaissent alternativement sur les 2 broches.

HEURE, DATE, ALARME

Pour avoir accès à l'heure et à la date stockées en format BCD, il suffit d'écrire dans/de lire les octets des registres correspondants. Le DS1615 fonctionne en mode 12 h ou 24 h (bit 6 de l'adresse 09 de la page *RTC AND CONTROL*). Le bit précédent départage, selon le cas, entre le matin et

l'après-midi ou indique la première ou la seconde dizaine d'heures (décade). Les registres d'alarme sont situés dans le domaine d'adressage 0007_h à 000A_h. Les 4 bits 7 (MS, MM, MH et MD) constituent un masque qui permet de spécifier dans quelle mesure le registre d'alarme doit coïncider avec celui de la RTC (0000_h à 0003_h) pour qu'une alarme se produise. Une modification des réglages d'alarme interrompt la saisie des données (pour des raisons de sécurité) et remet le bit MIP à zéro (*Reset*).

« SPECIAL PURPOSE »

Les fonctions du DS1615 décrites plus haut, aussi étonnantes soient-elles, ne constituent pas le mot de la fin. Les bits de la page *RTC/CONTROL* qui n'ont pas encore été mentionnés sont :

◆ **EOSC** (adresse 0E_h, bit 7)

Enable OSCillator commande l'oscillateur en mode de secours par batterie. L'oscillateur est actif si le bit est 0, sinon l'oscillateur s'arrête et l'enregistreur de température passe en mode *stand-by* de consommation au repos (environ 100 nA).

CLR (adresse 0E_h, bit 6)

Clear Memory exécute la commande *Clear Memory* (voir sous Commandes). Ce bit est mis à 0 à la fin du processus d'effacement.

AIE (adresse 0E_h, bit 0)

Alarm Interrupt Enable permet à *ALArM Flag* (ALMF) d'engendrer un signal **INT**.

MEM CLR (adresse 14_h, bit 6)

MEMory Clear indique que les mémoires Datalog, Histogram, Temperature Alarm, Current Samples, Start Time Stamp, Start Delay et Sample Rate Register sont vides. MEM CLR passe à 0 au début d'une série de mesures.

SIP (adresse 14_h, bit 4)

Sample In Progress est actif lors du traitement d'une mesure. La durée est limitée à 750 ms.

LOBAT (adresse 14_h, bit 3)

L'indicateur *LOW BATTERY Flag* devient actif lorsque les réserves de la batterie (Lithium) tirent à leur fin.

ALMF (adresse 14_h, bit 0)

Cet *ALArM Flag* indique que l'heure courante correspond à celle placée dans le registre d'alarme « Alarm ». Si le bit AIE est actif, **INT** passe à BAS. Si l'on positionne AIE sur BAS, ALMF est aussi remis à l'état initial.

COMMUNICATION

Le DS1615 peut faire appel à deux méthodes de communication. Toutes deux transmettent le bit de poids faible

en premier et le bit de poids fort en dernier. La liaison *COMMunication SElect Input* (COMSEL) permet de choisir le mode de communication synchrone ou asynchrone. Dans le premier cas, la communication se déroule (avec **RST** = 1) par la liaison sérielle du signal de rythme SCLK et la liaison des données I/O. Une commande de lecture ou d'écriture (voir sous Commandes) est suivie de l'adresse, puis des données. Le taux de transmission maximum est de 2 millions de bits/seconde.

Si COMSEL est à la masse (ou ouverte) la transmission est asynchrone ; le DS1615 communique par une interface de données semi-duplex à la vitesse de 9 600 bauds. Comme pour les UART, le protocole comporte les bits de départ et d'arrêt usuels. L'entrée des données du DS1615 est désignée par RX, la sortie par TX. La communication en mode asynchrone est contrôlée par un CRC (*Cyclic Redundancy Check*). Si la communication ne s'est pas effectuée correctement pour une raison ou autre, le DS1615, qui s'en aperçoit grâce à l'absence de bit d'arrêt ou à des délais de plus de 10 bits pendant les interruptions de la transmission, réinitialise automatiquement le processus de communication. Pour réinitialiser la communication en mode synchrone, il suffit d'utiliser la liaison **RST**.

COMMANDES

Pour communiquer avec le DS1615, on se sert de commandes suivies si nécessaire d'octets de paramètres. Il existe 5 commandes.

WRITE BYTE (22_h)

L'hôte écrit un octet dans la RTC, le registre d'instruction et la RAM non volatile de l'utilisateur. La commande d'écriture est suivie de l'adresse, le troisième octet contient la donnée. Le DS1615 ne donne pas de réponse.

READ BYTE (33_h)

L'hôte transmet un ordre de lecture suivi de 16 bits d'adressage (bit de poids fort en premier). Le DS1615 répond alors en envoyant par octets le contenu de la page choisie (plus 2 octets de contrôle par redondance CRC). En mode asynchrone, TX est désactivé après l'envoi du dernier registre, alors qu'en mode synchrone l'envoi est répété jusqu'à ce qu'un signal d'horloge ne soit plus transmis à l'interface sérielle.

SPECIFICATION TEST (44_h)

L'hôte envoie la commande, le DS1615 répond par 4 impulsions de niveau BAS à la broche *INSPEC* ou *OUTSPEC*, selon que les mesures indiquées sortent ou ne sortent pas de la plage de température définie.

READ TEMPERATURE (55_h)

Lorsque le DS1615 reçoit cette com-

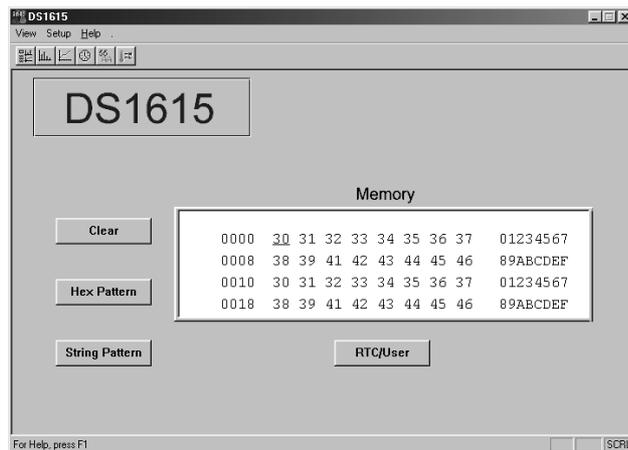
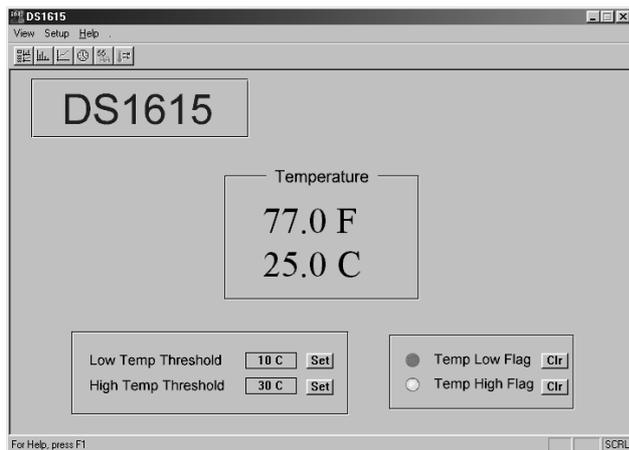
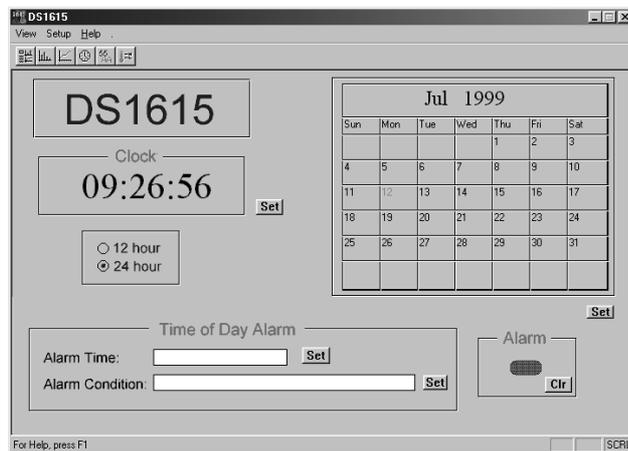
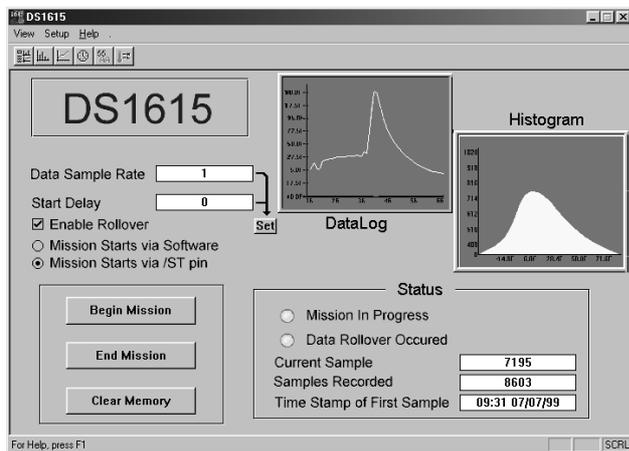


Figure 5. Ces saisies d'écran prouvent la convivialité du logiciel.

mande de l'hôte, il lit la valeur actuelle de la température et la mémorise dans le registre Current Temperature (si MIP est 0). Cette valeur de la température n'est toutefois pas incluse dans la mémoire de l'enregistrement ou de l'histogramme. Il faut tester le bit TR pour déterminer la fin de la conversion.

CLEAR MEMORY (A5_h)

Cet ordre, une fois donné par CLR et transmis, efface les mémoires Datalog, Histogram, Temperature Alarm, Current Samples, Start Time Stamp, Start Delay et Sample Rate Register.

KIT D'ÉVALUATION DS1615

Pour démontrer la puissance de son enregistreur de température, Dallas fournit un kit d'évaluation qui permet de programmer les registres, la mémoire, la date, les alarmes et les valeurs de seuil du DS1615. Ce kit comporte une petite platine sur laquelle est monté un DS1615 raccordé à un connecteur Sub-D, quelques composants supplémentaires, un câble RS232 ainsi qu'un logiciel sous Windows 95/98 qui sera disponible auprès des adresses habituelles (**EPS 996030-1a et 1b**) et (très prochainement selon Dallas) également sur Internet. On trouve en outre sur les 2 disquettes toute la documentation du DS1615 ainsi que le code source du logiciel.

Le plan de câblage de la **figure 3** com-

porte, outre l'enregistreur de température, un circuit intégré supplémentaire, le DS275. Ce circuit intégré gère la communication série avec le PC. Le rôle principal du DS275 consiste à convertir les niveaux symétriques du PC en niveaux TTL/CMOS et réciproquement, de façon similaire au MAX232 bien connu, mais avec une consommation bien moindre. Ce circuit intégré « vole » en effet l'énergie de la partie négative du signal RX_IN et la met à disposition du signal TX_OUT.

MENUS 5 ÉTOILES

Après une installation sans problème du logiciel, on peut lancer le programme au moyen de l'icône qui se trouve sur le bureau ou de l'élément du sous-menu Programmes du menu Démarrer de Windows. Le **menu principal (figure 5a)** qui apparaît ne donne pas seulement une rapide impression visuelle de la saisie des données et de l'histogramme, mais permet aussi de régler la fréquence d'acquisition, le délai initial ainsi que le mode de mise en marche (bouton/PC). Les deux diagrammes grossiers sont convertis en mode plein écran et haute résolution en pressant les boutons appropriés de la barre de menu. Les mesures sont lancées et arrêtées à l'aide du menu principal. La fenêtre d'état donne le nombre de valeurs collectées, le repère de temps de la première mesure, l'état des bits

MIP et RO. Il est aussi possible d'effacer toutes les mémoires.

Le **menu Heure et calendrier (figure 5b)** permet d'ajuster et de lire l'heure (en 12/24 h), le calendrier et l'alarme. L'année n'est indiquée qu'avec deux chiffres (compatible an 2000!). L'indicateur d'alarme signale les alarmes horaires et de données.

Le **menu Current Temperature/Threshold Set (figure 5c)** se passerait presque d'explications. La zone d'état permet de régler les deux seuils entre -40 °C et + 85 °C par pas de 0,5 °C. La fenêtre d'état indique aussi si toutes les mesures acquises se trouvent dans cette plage.

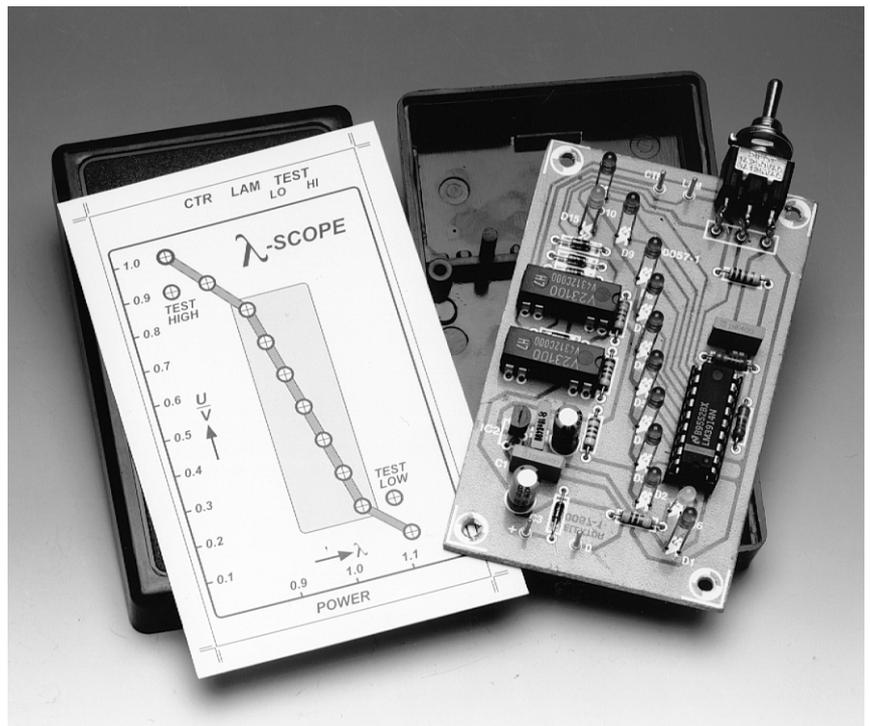
Le **menu User Memory (figure 5d)**, enfin, permet à l'utilisateur de programmer et d'éditer la mémoire non volatile. La mémoire est complètement effacée lorsqu'on presse sur CLEAR. La programmation est effectuée en hexadécimal ou par chaînes de caractères. La première ligne de la fenêtre « User NV Memory » indique le domaine d'adressage, la deuxième les 8 octets de données et leur équivalent ASCII à droite.

(990073)

testeur de sonde lambda

comment contrôler l'exactitude du mélange

De quoi servirait le meilleur pot catalytique si ses constituants ne fonctionnent pas correctement ? Parce qu'à côté du catalyseur proprement dit, dont la durée de vie est limitée, le capteur peut, lui aussi, être défaillant ou en panne. Mais un montage simple vous dira tout à son propos.



Le système actuel le plus efficace pour réduire considérablement l'émission de gaz nocif par les automobiles, c'est le catalyseur à trois voies équipé d'une sonde lambda. Le système joue sur la relation entre la richesse du mélange air-carburant qui alimente le moteur et la composition des gaz d'échappement. Le catalyseur G, à réglage, se compose de plusieurs parties. Le pot catalytique inséré dans l'échappement modifie la composition des gaz par un genre de postcombustion, une oxydation, mais aussi par rupture de certaines liaisons chimiques, celles d'avec l'oxygène, produisant donc une réduction, sans pour autant consommer sa propre substance. Le catalyseur G transforme ainsi les émanations nocives en gaz non polluants pour l'environnement.

La combustion des matières fossiles s'accompagne d'émission de radicaux HC (du carburant non brûlé), NO_x (les oxydes d'azote) et CO (le monoxyde

de carbone). Aussi longtemps que le rapport de 14 à 1 entre air et carburant est maintenu dans le mélange, le catalyseur est en mesure de convertir la plus grosse part des gaz nocifs. Il réduit NO_x en N_2 , (l'azote), séparé de O_2 (l'oxygène) ; à l'inverse, il oxyde HC et CO en H_2O (l'eau) et CO_2 (le dioxyde de carbone). Ce sont des constituants naturels de notre environnement, en principe dénués de toxicité, sauf l'éventuel effet de serre que peut causer l'excès de CO_2 dans l'atmosphère.

UN CAPTEUR CHAUD

La sonde lambda (**figure 1**), le coeur du dispositif régulateur, analyse la composition des gaz d'échappement de manière à optimiser le réglage électronique du mélange d'air et de carburant à l'admission. Le rapport entre le mélange gazeux issu de la combustion et celui qui a été consommé s'appelle, comme la onzième lettre de l'alphabet grec, **lambda**, à ne pas confondre avec la

Le *Lambda-scope* est le fruit d'un projet étudié par la classe TEV12 de l'école d'électrotechnique professionnelle Ludwig Geissler à Hanau (RFA). Nos remerciements aux étudiants et à leur professeur, Reinhart Weber.

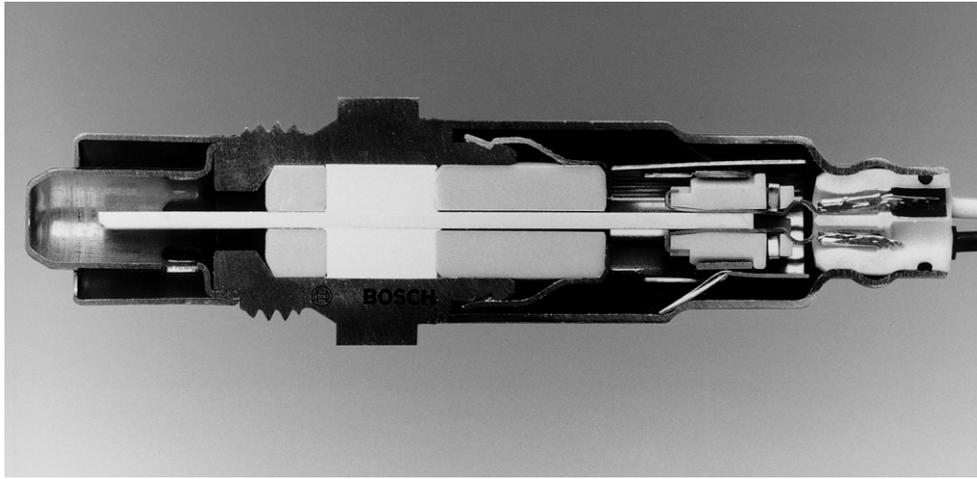


Figure 1. La sonde lambda planar à large bande de Bosch.

lambda, il y aurait de la java dans l'air ! À la proportion optimale de 14 à 1 correspond une valeur lambda de 1.

Une sonde lambda se compose habituellement d'un récipient en céramique au dioxyde de zirconium (ZrO_2), disposé dans l'échappement, garni d'une couche de platine perméable aux gaz et d'une céramique poreuse inusable. L'intérieur du récipient est mis en communication avec l'air ambiant. Le fonctionnement de la sonde repose sur le fait qu'à une température de la céramique supérieure à 600 °C, elle laisse passer les ions oxygène. Il lui faut dès lors un élément chauffant interne supplémentaire pour l'amener à cette température. Si la teneur en O_2 des gaz brûlés diffère

sensiblement de celle de l'air, d'une paroi à l'autre de la céramique, une tension électrique apparaît aux électrodes, proportionnelle à la valeur de lambda.

NI TROP PAUVRE, NI TROP RICHE

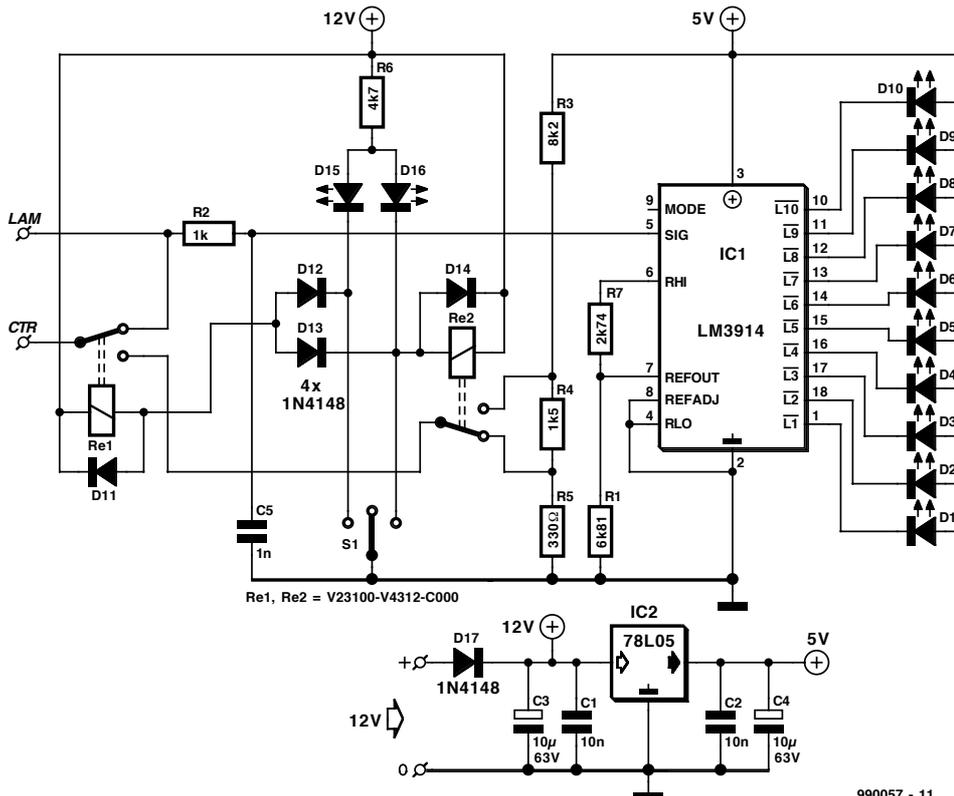
Pour un rapport idéal du mélange correspondant à un lambda de 14 à 1, on observe une tension pulsée de 0,5 V, la fréquence dépend de la constante de temps du circuit de réglage. Si la tension se maintient à 0,5 V, sans fluctuation, c'est le signe d'une panne dans le circuit de réglage, ou encore que la sonde est inactive ou froide. Une tension qui se

maintient durablement à une valeur plus grande indique un mélange trop riche, c'est-à-dire qu'il contient trop peu d'air, une situation qui n'est tolérable que si le moteur est froid ou qu'il est à pleine charge, dans certains cas. Sinon, il faudra contrôler le dispositif d'injection ou encore la température du capteur.

Pour un mélange trop pauvre, la tension chute sous les 0,5 V. Il peut s'agir également d'une situation acceptable, lors de la décélération active qui se produit quand on lève le pied à haut régime pour profiter du frein moteur.

Si tel n'est pas le cas, il y a lieu d'envisager un problème d'alimentation, une admission trop réduite de carburant.

Figure 2. Le testeur de sonde lambda se résume à un voltmètre et une source de tension de référence commutable.



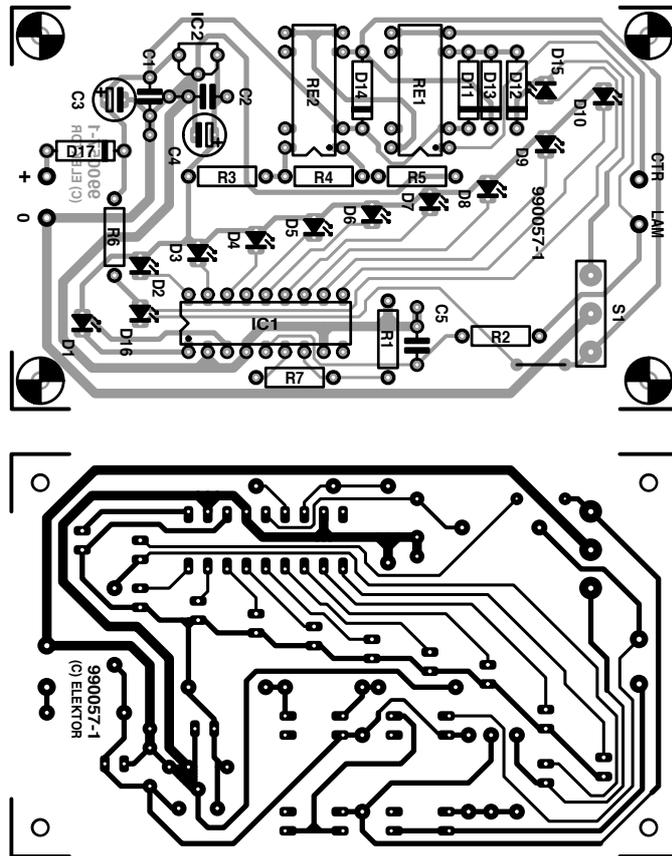


Figure 3. Les LED, sur la platine, tracent une courbe de réglage.

Liste des composants

Résistances :

R1 = 6kΩ81
 R2 = 1 kΩ
 R3 = 8kΩ2
 R4 = 1kΩ5
 R5 = 330 Ω
 R6,R7 = 4kΩ7
 R8 = 2kΩ49

Condensateurs :

C1,C2 = 10 nF
 C3,C4 = 10 μF/63 V vertical
 C5 = 1 nF

Semi-conducteurs :

D1 à D10 = LED 3 mm rouge
 D11,D13 à D16,D19 = 1N4148
 D17 = LED 3 mm jaune
 D18 = LED 3 mm verte
 IC1 = LM3914 (National Semiconductor)
 IC2 = 78L05

Divers :

S1 = inverseur unipolaire à position médiane encartable en équerre
 Re1 à Re3 = relais LS (Reed) unipolaire tel que, par exemple, Siemens V23100-V4312C000

UNE RANGÉE DE LED

Ainsi donc, d'un point de vue d'électronicien, tout ce dont nous avons besoin pour contrôler le fonctionnement de la sonde lambda, c'est d'un simple voltmètre et très peu d'accessoires, comme le laisse présager la figure 2. Le petit instrument de vérification, outre les raccords d'alimenta-

tion, présente deux points de contact avec le monde extérieur, pour le capteur repéré ici LAM et l'entrée de l'électronique de réglage, la borne CTR. Le capteur se connecte à l'entrée signal d'une puce d'attaque d'une colonne de LED LM3914, bien connue de tout un chacun.

Le réseau R2 et C5 constitue un filtre passe-bas destiné à éliminer les pics de tension parasites à l'entrée du signal. La tension de référence du LM3914 est déterminée par R1 et R8 de manière telle que la fenêtre de tension observable se situe entre 0 V (D1) et 1 V (D10), par échelons de 0,1 V, et c'est D6 qui y figure la valeur idéale de 0,5 V. La broche Mode reste en l'air, si bien que l'afficheur travaille à point unique, une seule LED s'éclaire à la fois.

Le câble du véhicule qui assure le lien entre capteur et régulateur doit évidemment être débranché avant de mettre en service l'appareil de test. On peut éventuellement le retirer de la prise pour le connecter sur le testeur.

SUIVI ET CONTRÔLE

Le commutateur S1 autorise divers modes de fonctionnement. Dans la position médiane où il est représenté, Re1 établit la liaison entre CTR et LAM. La boucle de régulation est fermée, l'appareil travaille en surveillance, il ne fait qu'observer la sortie de la sonde et en affiche la tension instantanée. Il peut donc rester en service au cours du fonctionnement normal de la voiture. Il faut toutefois men-

tionner qu'alors le véhicule peut ne plus être conforme aux prescriptions concernant la garantie.

Dans les deux autres positions du commutateur, l'appareil fonctionne comme testeur, la boucle de régulation est ouverte, de sorte que le mélange d'alimentation peut être rendu trop riche si le commutateur est en position *Test-Hi*, ou trop pauvre, en face de *Test-Lo*. La tension pulsée provient du diviseur de tension R3 à R5, commutée par Re2. Les LED D15 et D16 indiquent la tension d'essai choisie, D15 pour la plus haute, D16 pour la plus basse. La commande de mélange d'injection s'efforce alors d'optimiser le rapport, tantôt de l'enrichir pour une basse tension, de l'appauvrir s'il est soumis à une tension plus grande.

L'alimentation stabilisée s'obtient de la manière habituelle, à l'aide d'un 78L05. La diode D17 empêche que le montage ne soit endommagé lors d'une éventuelle inversion de polarité. En sortie, on trouve une tension continue stable et bien filtrée de + 5 V. Les relais se branchent directement sur le + 12 V.

LA COURBE DE RÉGLAGE EN POINT DE MIRE

Ainsi que vous pouvez le voir à la figure 3, les LED sur la platine du testeur de sonde lambda sont disposées en forme de courbe de réglage. Si les diodes s'éclairent dans la zone médiane délimitée, le catalyseur G a toutes les chances de fonctionner correctement. Si le point lumineux se déplace du côté des LED D9 et D10, le mélange est trop riche, il est en revanche trop pauvre vers D1.

Les LED dans le mode de test se trouvent en regard de la courbe de réglage, de manière telle que le contrôle du mélange puisse être corrigé précisément pour la valeur de lambda correspondante.

L'assemblage des composants sur la platine et la mise en boîte ne devrait pas poser problème. La puce IC1 et les deux relais à lames souples doivent se loger dans des supports. Mieux vaut souder les LED en premier, question de s'assurer d'une parfaite coïncidence avec les trous dans le boîtier.

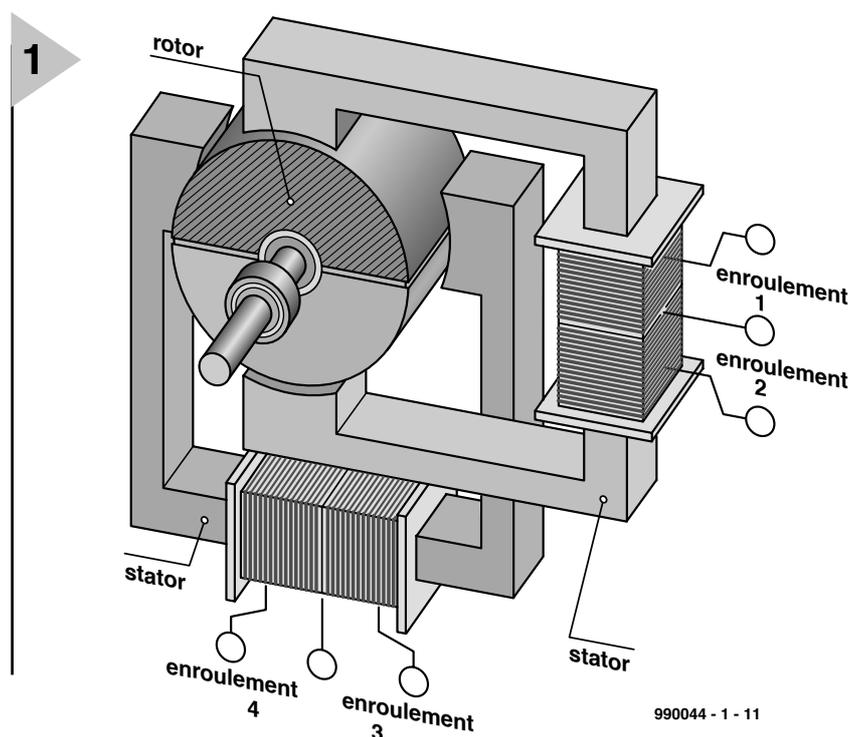
Mais c'est au câblage qu'il faut accorder toute son attention. Il convient de prélever la tension du capteur sur un connecteur situé aussi près que possible de la sonde et transmettre le signal impérativement sous câble blindé au testeur. La liaison au régulateur se réalise, elle aussi, en câble blindé.

(990057)

moteurs pas à pas

construction, fonctionnement et commande

Les moteurs pas à pas sont connus depuis le début des années 60, mais leur importance a crû fortement au cours des dernières années. Ils servent d'entraînement dans les montres et dans d'autres instruments à aiguilles, dans les imprimantes et les tables traçantes, les machines-outils et autres robots.



990044 - 1 - 11

Figure 1. Modèle simplifié de moteur pas à pas composé d'un rotor à deux pôles et de deux enroulements décalés de 90°.

Les moteurs pas à pas se composent d'un stator fixe et d'un rotor tournant, ne différant aucunement en cela de n'importe quel autre moteur. Mais, alors que le rotor d'un moteur à courant continu (et parfois aussi son stator) est équipé d'électro-aimants, un moteur pas à pas ne possède que des électro-aimants stationnaires. Le rotor, constitué par du fer doux aimanté ou un aimant permanent, pivote sous l'action des champs diversement orientés des électro-aimants du stator. Même sans savoir exactement comment fonctionne un moteur pas à pas, un avantage important saute aux yeux : le rotor n'est pas alimenté en courant, finie l'usure des balais et des bagues ! Les premiers moteurs pas à pas à **réluctance**, fort coûteux, comportaient un noyau de fer doux denté en guise de

rotor. Outre la faiblesse de leur couple, ces moteurs à réluctance ne possédaient pas de « position de crantage » puisque le noyau de fer doux était dépourvu de pôles magnétiques intrinsèques. C'est la raison de l'introduction de moteurs pas à pas permanents. Si leur rotor magnétisé à demeure offrait bien des positions de crantage, l'angle entre celles-ci était considérable. La raison : il n'est possible de disposer qu'un nombre restreint de **pôles magnétiques** sur un rotor cylindrique magnétisé radialement. Nous nous servirons toutefois de ce type de moteur pas à pas pour introduire le sujet. La **figure 1** représente la version la plus simple d'un moteur pas à pas. Le noyau n'est magnétisé qu'une fois (deux pôles) et le moteur comporte deux enroulements (**phases**) disposés à

90° l'un de l'autre. Un courant qui passe par une des phases engendre un champ magnétique. Le rotor, qui comporte une paire de pôles magnétiques, se place dans une position telle que l'entrefer entre le rotor et la phase est le plus petit et le flux magnétique le plus élevé. (Les pôles contraires s'attirent, les pôles semblables se repoussent !)

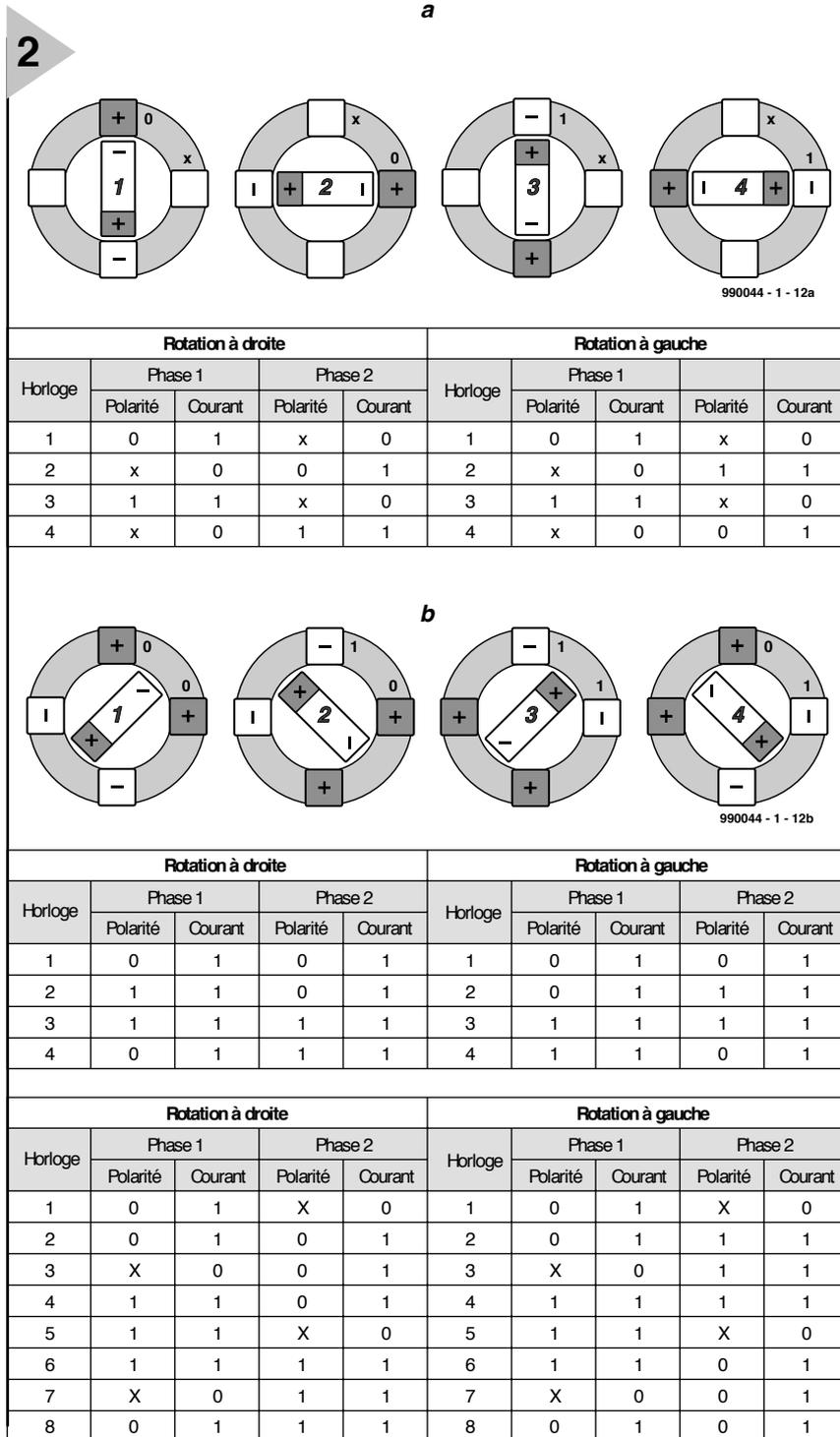
En inversant la direction du courant, il est possible de placer le rotor dans quatre positions différentes ; le sens de rotation du rotor suit les inversions de polarité. Ce mode de fonctionnement en **Wavedrive** est représenté schématiquement dans la **figure 2a**.

Une autre méthode pour faire pivoter le moteur pas à pas consiste à faire passer le courant par les deux phases en mode de **fonctionnement normal**. Dans ce cas, deux phases identiques se trouvent toujours l'une à côté de l'autre. La **figure 2b** donne toute la lumière sur le comportement du rotor. Une **séquence** est la rotation « électrique » de 360° (pas angulaire électrique) requise pour accomplir un pas angulaire mécanique complet. Les deux modes en **pas entier** mentionnés ci-dessus comportent une séquence de quatre cycles. Un pas angulaire mécanique de l'exemple est égal à une rotation complète du moteur de 360°.

Rien ne s'oppose à combiner le mode Wavedrive au mode de fonctionnement normal pour former ce qu'on appelle une commande **demi-pas**. Cela permet d'introduire des quasi demi-pas qui doublent « gratis » la résolution du moteur. Dans ce mode de fonctionnement, on alimente alternativement une et deux bobines. Une séquence du mode demi-pas se compose donc de huit cycles.

On peut encore effectuer une autre distinction entre deux techniques de commande additionnelles selon la conception des enroulements. Moyennement une modification élémentaire (un simple commutateur), on obtient le **mode unipolaire** autrefois très en faveur (**figure 3a**). Il faut toutefois que les enroulements soient munis d'une prise médiane, faute de quoi le courant limité des bobines ne permet d'atteindre qu'un faible couple et qu'un nombre de tours peu élevé. Seule l'arrivée de composants intégrés peu coûteux pour la commande des moteurs pas à pas a permis à la **technique bipolaire** (**figure 3b**) de s'imposer toujours davantage. Les enroulements y sont inversés des deux côtés. La commande doit donc disposer de deux commutateurs au lieu d'un seul.

Les moteurs pas à pas à nombre restreint de phases ont tendance à fonctionner par à-coups, même en mode demi-pas, particulièrement aux basses fréquences. Il est toutefois possible d'améliorer la situation en ne coupant et ne rétablissant pas simplement le



courant, mais en effectuant ces opérations graduellement. Ce mode **micro-pas** permet d'effectuer des mouvements de rotation fluides. Le mode micro-pas réduit toutefois le **couple** et la précision du positionnement, ce qui constitue avant tout un désavantage dans le cas des moteurs pas à pas ne comportant que peu de positions du rotor.

EN PRATIQUE

Nous n'avons parlé, jusqu'à présent, que d'un modèle de moteur pas à pas

Figure 2. Courant dans le Wavedrive (a) et en mode de fonctionnement normal (b). On obtient une commande demi-pas en combinant ces deux modes de fonctionnement en pas entier.

qui n'a pas sa place dans la pratique. Dans les **moteurs pas à pas hybrides**, on se sert de disques aimantés axialement de

façon permanente comme noyau. Ces disques sont munis de couronnes dentées décalées d'une demi-dent l'une par rapport à l'autre, de sorte que pôles nord et pôles sud alternent. La photo (**figure 4**) montre l'intérieur d'un moteur pas à pas hybride de ce type. On aperçoit clairement la denture du rotor.

Le **pas angulaire**, la **résolution** du

3

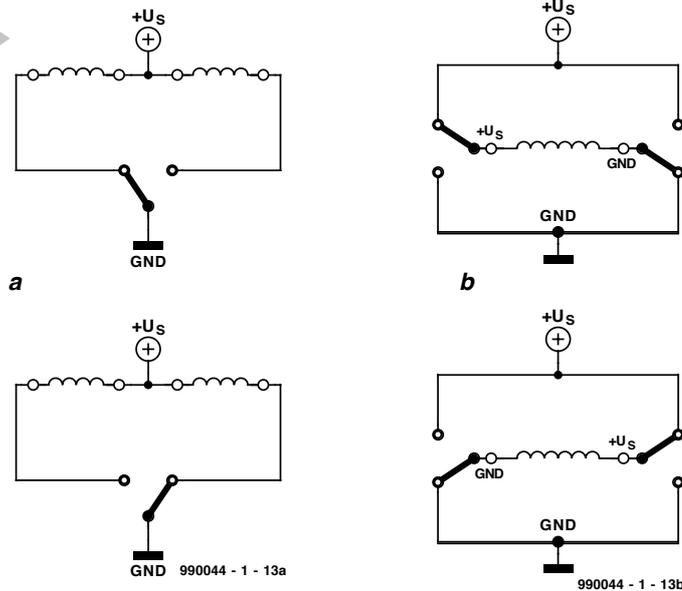


Figure 3. Les enroulements du moteur en mode unipolaire (a) doivent comporter une prise médiane. Les moteurs bipolaires (b) doivent être équipés d'une commande comportant deux commutateurs par enroulement.

4

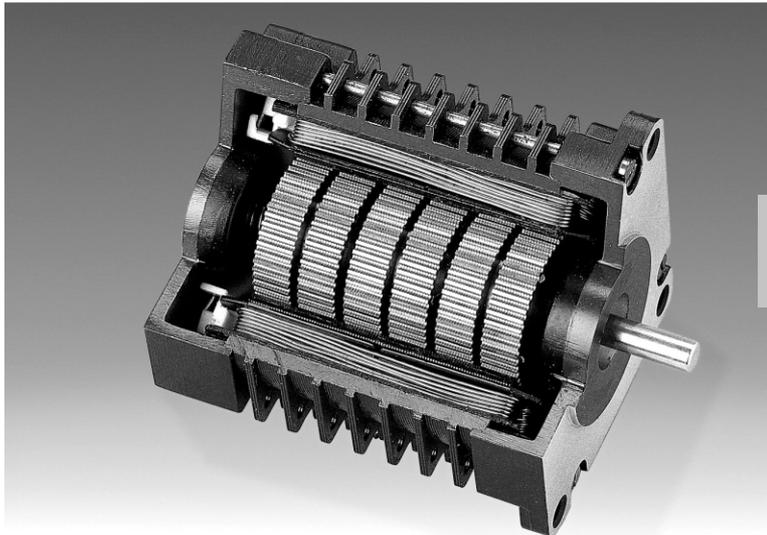
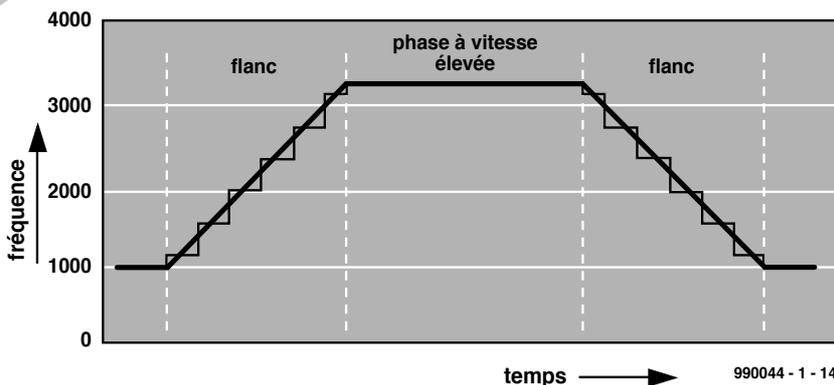


Figure 4. On aperçoit clairement la denture du rotor.

Figure 5. Accélération et freinage au moyen d'une fonction de rampe.

5



moteur ne dépend pas que du nombre de paires de pôles, donc des dents nord et sud du rotor, mais aussi du nombre de phases qui peuvent être commandées séparément. On se limite en pratique à des moteurs de deux à cinq pôles pour garder le câblage et la complexité de commutation dans des limites raisonnables. Si l'on attache de l'importance à une valeur élevée du couple, il faut donner la préférence aux moteurs pas à pas à deux pôles, tandis qu'un fonctionnement sans à-coups n'est atteint qu'avec un moteur à cinq pôles (en mode micro-pas). Un moteur trois pôles constitue un bon compromis.

La rotation des moteurs pas à pas est en général divisée en 24 pas (15°) ou 48 pas ($7,5^\circ$) au moins. Les anciennes unités de disques durs utilisaient dans la plupart des cas des moteurs de 200 ($1,8^\circ$) ou même de 400 ($0,9^\circ$) pas pour positionner la tête avec exactitude.

Si l'on applique une cadence de plusieurs kilohertz au moteur, celui-ci ne se meut pas du tout, car l'inertie du rotor est bien trop élevée pour suivre les rapides variations du champ tournant du stator. Une **fréquence marche/arrêt** du moteur entre 50 Hz et 2 000 Hz, selon sa construction, assurera à coup sûr son démarrage. On peut accroître la cadence après le démarrage. L'**accélération** peut être aussi faible que l'on veut, mais malheureusement pas aussi forte : le moteur pas à pas resterait aussi immobile que lors d'une tentative de dépasser la cadence la plus élevée.

La variation de la fréquence marche/arrêt lors de la phase d'accélération jusqu'à l'obtention de la vitesse de consigne ressemble à une rampe montante et est en effet désignée comme « fonction rampe » (figure 5). L'arrêt est aussi produit à l'aide d'une fonction rampe (descendante), car la masse du rotor continuerait à entraîner celui-ci après une coupure brutale. Les informations sur la position exacte du rotor seraient perdues. Ce n'est que pour la recherche du point zéro et pour des déplacements lents que l'on choisit une cadence pour laquelle le moteur pas à pas peut pivoter fiablement dans les deux sens sans fonction rampe. Pour faire pivoter le moteur pas à pas, principalement en mode demi-pas ou micro-pas, et pour créer en outre des rampes d'accélération supplémentaires, il faut remplacer la commande par un ordinateur. Ce type de commande de moteur pas à pas pour carte 80C166 sera décrit dans le prochain numéro.

(990044)

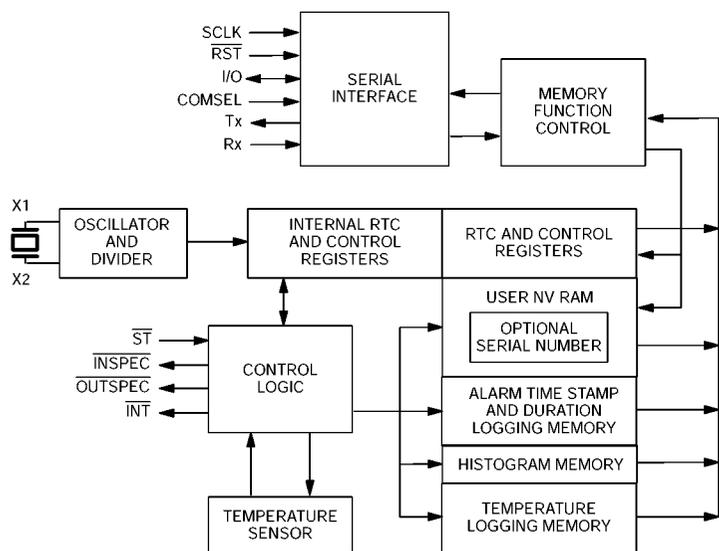
DS1615

Circuits Intégrés
Composants spéciaux

ELEKTOR

INFOCARTE

9 / 9 9



993007 - 12

Cartographie du DS1615.

INT (Interrupt)

La broche INT est une broche de sortie à drain ouvert, active au niveau bas qui pourra attaquer l'entrée d'interruption d'un microprocesseur. La sortie INT reste au niveau bas tant que le bit d'état auquel est dû l'interruption est présent et que le bit correspondant, *interrupt-enable*, est positionné.

INSPEC (IN-SPECification)

Cette broche de sortie à drain ouvert sert, en combinaison avec la broche OUTSPEC, à rendre l'état de fonctionnement et la situation au niveau des données du DS1615.

OUTSPEC (OUT-of-SPECification)

Cette broche de sortie à drain ouvert sert, en association avec la broche INSPEC, à rendre l'état de fonctionnement et la situation au niveau des données du DS1615.

ST (STart/Status)

L'entrée du bouton ST remplit une double fonction. La première, lorsqu'il est validé pour la fonction de

lancement de saisie de données (bit SE du registre de commande est à 1), la broche ST sert à donner l'instruction au DS1615 de démarrer la saisie des données de température à partir des informations programmées de temporisation de capture et de taux d'échantillonnage. La broche ST doit être maintenue au niveau bas pendant un minimum de 0,5 s pour démarrer la saisie des données. Il faudra prendre à cette broche une résistance de forçage au niveau haut (*pull up*) externe.

Pour sa seconde fonction, la broche ST peut servir à interroger la situation en ce qui concerne les données enregistrées. Une fois la capture des données lancée, la broche ST donne l'instruction au DS1615 de fournir, par le biais des broches INSPEC et OUTSPEC, l'état des données saisies.

X1, X2

On pourra connecter entre ces broches un quartz de 32,768 kHz standard. Pour la précision la meilleure, le DS1615 devra être doté d'un quartz ayant la

DS1615

Circuits Intégrés
Composants spéciaux

ELEKTOR

INFOCARTE

9 / 9 9

DS1615

Enregistreur de température

Fabricant

Dallas Semiconductor

Site Internet : www.dalsemi.com

Caractéristiques techniques :

- Thermomètre numérique mesurant les températures entre -40 °C et $+85\text{ °C}$ par pas de $0,5\text{ °C}$
- Précision du thermomètre numérique de $\pm 2\text{ °C}$
- Horloge/calendrier en temps réel au format BCD, compte les secondes, les minutes, les heures, donne la date, le mois, le jour de la semaine et l'année avec prise en compte de la « bisextilité » d'une année (compatible Y2K).
- Se réveille automatiquement et mesure la température à un intervalle programmable par l'utilisateur entre 1 et 255 mn
- Enregistre jusqu'à 2 048 mesures de température consécutives dans une mémoire non-volatile à écriture seule
- Enregistre un histogramme à long terme en 63 corbeilles et ce à une résolution de $2,0\text{ °C}$
- Possibilité de programmation de seuil de température haut et bas pour activation d'une alarme
- 2 options d'interface sérielle : synchrone et asynchrone
 - interface sérielle synchrone à 3 fils
 - interface sérielle asynchrone compatible avec les UART standard
- Mémoire subdivisée en pages de 32 octets pour compression des données
- Générateur CRC 16 bits embarqué pour suivi et sauvegarde des opérations de lecture de données en mode de communication asynchrone
- En option, numéro de série à 64 bits unique gravé au laser et testé à la fabrication

Exemple d'application

Enregistreur de température, *Elektor*, n°255, septembre 1999

Description

Le DS1615 est un enregistreur de température combinant une horloge en temps réel avec des capacités de saisie de données de température et d'histogramme. Il a été conçu pour des applications qui requièrent la saisie du profil de la température sur une plage de temps donnée. Son taux d'échantillonnage programmable font de ce composant la solution idéale lorsqu'il s'agit d'enregistrer la température sur une période plus ou moins longue.

L'horloge en temps réel (RTC = *Real Time Clock*) fournit les secondes, les minutes, les heures, donne la date, le mois, le jour de la semaine avec compensation de la « bisextilité »; elle fournit également une interruption pour une alarme. La mesure de température fait appel à un capteur intégré sur la puce permettant la mesure de températures allant de -40 °C à $+85\text{ °C}$ par pas de $0,5\text{ °C}$.

Le DS1615 est un composants d'enregistrement de donnée puissant fournissant à la fois une liste des valeurs de échantillons de température et son histogramme.

La fonction de centrale de données se contente d'échantillonner la température à un intervalle défini par l'utilisateur et écrit les données dans la mémoire dite de *Temperature Datalog*. Il est possible d'enregistrer un maximum de 2 048 échantillons. La fonction d'histogramme est réalisée par l'échantillonnage de la température pour ensuite incrémenter la valeur du compteur d'une corbeille de données associée à la température concernée.

Le DS1615 dispose de 63 corbeilles de données à 2 octets au pas de 2 °C . L'utilisateur peut programmer l'échantillonnage de données pour tout à la fois la saisie de données et la tabulation d'un histogramme à des intervalles s'étalant de 1 fois par minute jusqu'à une fois toutes les 255 mn.

Le DS1615 permet également la définition de points de température minimale et maximale d'entrée en fonction d'un dispositif d'alarme qui permet au composant de vérifier que la température reste à l'intérieur du domaine défini. Le composant peut déclencher une interruption ou attaquer une broche d'état lorsque la température sort du domaine prédéfini.

Le DS1615 peut être programmé pour démarrer



l'échantillonnage suite à une action sur un bouton-poussoir ou par le biais d'une commande émise au travers de l'interface sérielle par une machine-hôte.

Le DS1615 dispose également d'un numéro de série à 64 bits optionnel permettant l'identification et le suivi du composant.

Structure

La structure interne donnée ci-contre illustre la relation entre les différents blocs importants, commande (CONTROL) et mémoire (MEMORY) du DS1615. Ce composant comporte 5 composants principaux ayant trait aux données :

- 1) l'horloge en temps réel (RTC) et le bloc de commande (CONTROL),
- 2) 32 octets de NVRAM utilisateur avec un numéro de série à 64 bits optionnel gravé au laser,
- 3) 96 octets de mémoire d'événement/de durée pour l'alarme,
- 4) 128 octets de RAM d'histogramme,
- 5) 2 048 octets de mémoire de stockage des données (*data log*). Toutes ces mémoires sont prises dans un unique domaine d'adresse linéaire.

Description des signaux

V_{CC}

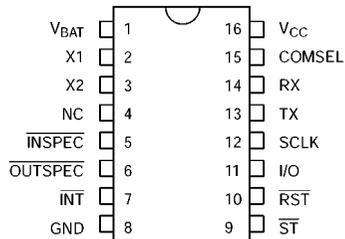
V_{CC} est l'entrée d'alimentation à +5 V. Il faut que cette broche soit connectée à l'alimentation +5 V pour qu'une communication avec le DS1615 soit possible.

V_{BAT}

Entrée batterie pour une pile au lithium standard ou toute autre source d'énergie. Toutes les fonctions du DS1615, exception faite de la circuiterie de l'interface sérielle, sont alimentées par V_{CC} lorsque V_{CC} > V_{BAT}. Il faudra, si l'on ne prévoit pas d'utiliser une pile ou toute autre source d'énergie, connecter la broche V_{BAT} directement à la masse.

GND

Masse (GROUND)



DS1615 16±PIN DIP
DS1615S 16±PIN SOIC (300 MIL)
993007 - 11

Brochage

COMSEL (COMmunication SELect)

Cette broche d'entrée détermine le mode, asynchrone ou synchrone, de la communication sérielle. Lorsqu'elle est forcée au niveau haut à V_{CC}, la communication se fera sériellement par le biais des broches SCLK, I/O et RST. Si COMSEL est forcée à la masse, on se trouvera en mode de communication asynchrone qui se servira des broches TX et RX. Si on laisse cette broche en l'air le DS1615 passera en mode asynchrone sachant que la broche COMSEL est dotée d'une résistance de forçage au niveau bas (*pull down*) interne de faible valeur.

Tx (Transmit)

Broche Transmit (sortie de transmission) de l'interface sérielle asynchrone. Tx se trouve à l'état de haute impédance (*tri-state*) dès que V_{CC} < V_{BAT}.

Rx (Receive)

Broche d'entrée Receive (réception) de l'interface sérielle.

SCLK (Serial Clock)

La broche SCLK est l'entrée d'horloge sérielle pour le canal de communication synchrone à 3 fils.

I/O (Input/Output)

La broche I/O est le signal d'Entrées/Sorties de données pour le canal de communication synchrone à 3 liaisons.

RST (Reset)

La broche RST est la broche de remise à zéro pour le canal de communication synchrone trifilaire.

ADDRESS		
0000H TO 001FH	RTC AND CONTROL REGISTERS	PAGE 0
0020H TO 003FH	(RESERVED)	PAGE 1
0040H TO 005FH	USER NV RAM	PAGE 2
0060H TO 0217H	(RESERVED FOR FUTURE EXTENSIONS)	PAGE 3 TO PAGE 16 (EXCLUDING LAST 8 BYTES OF PAGE 16)
0218H TO 021FH	SERIAL NUMBER (OPTIONAL)	PAGE 16 (LAST 8 BYTES)
00220H TO 027FH	ALARM TIME STAMPS AND DURATIONS	PAGE 17 TO PAGE 19
0280H TO 07FFH	(RESERVED FOR FUTURE EXTENSIONS)	PAGES 20 ± 63
0800H TO 087FH	TEMPERATURE HISTOGRAM (63 BINS OF 2 BYTES EACH)	PAGE 64 TO PAGE 67
0880H TO 0FFFH	(RESERVED FOR FUTURE EXTENSIONS)	PAGES 68 ± 127
1000H TO 17FFH	TEMPERATURE DATALOG MEMORY (64 PAGES)	PAGE 128 TO PAGE 191
1800H AND HIGHER	(RESERVED FOR FUTURE EXTENSIONS)	PAGE 192 AND HIGHER

993007 - 13

Structure interne du DS1615.

Mémoire

capacité de charge spécifiée de 6 pF. Aucun condensateur ou résistance externe n'est nécessaire. Note : X1 et X2 sont des broches à impédance très élevée. Il est recommandé de les mettre, de même que le quartz, à un plan de masse en veillant en outre à garder les signaux HF à distance de la zone du quartz.

Pour de plus amples informations concernant le choix du quartz et le dessin de la platine à proximité immédiate, on consultera la note d'applications n°58 baptisée « *Crystal Considerations with Dallas Real Time Clocks* ».

NC (No Connect)

Cette broche ne devra pas être connectée.

La cartographie mémoire ci-dessus montre l'organisation générale du DS1615. On constate que la mémoire est subdivisée en pages de 32 octets. Les pages 0 et 1 contiennent l'horloge en temps réel (RTC) et les registres de commande. La RAM N(ON) V(olatile) réside en page 2. Les pages 17 à 19 sont assignées au stockage des tampons horodateurs et des durées des alarmes. Les corbeilles de récupération pour l'obtention de l'histogramme débutent en page 64 et utilise jusqu'à 4 pages.

La mémoire de saisie de la température couvre les pages 128 à 191. Les pages de mémoire 1, 3 à 16, 20 à 63, 68 à 127 et 192 et au-delà sont réservées à de futures extensions.