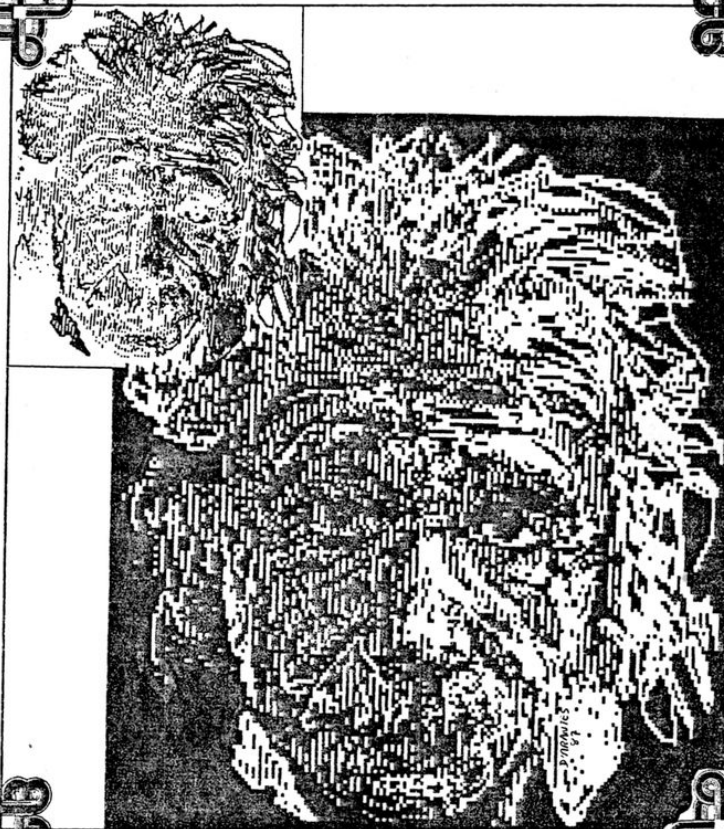




**DAI  
CLUB**



**17:12**



**INTERNATIONAL DAI CLUB  
COMPUTER ASSOCIATION**

**DAICLIC INFOS :**

DAICLIC paraît tous les trois mois.  
l'abonnement est compris dans la cotisation annuelle à I.D.C.Bordeaux et à D.C.A. (du 1/1 au 31/12). A l'inscription, les numéros déjà parus dans l'année sont envoyés.

International DAI Club Bordeaux C/O Bruno Delannay  
Res. Les Acacias St.B3, Avenue de Saige  
33600 PESSAC (FRANCE) tel : 56.45.87.70.

Dai Computer Association C/O D'ARANTES Luc  
10 D Res. Sainte Marie, 6 Chemin Sainte Marie  
30200 BAGNOLS sur CEZE (FRANCE)  
tel : 66.89.57.60

**ACTIONS RESPECTIVES de I.D.C.Bordeaux et D.C.A. :**

**I . D . C . B x . :**

- \* Edition de la revue + Achat de Logiciels
- \* Librairie + Bibliothèque + Service Listing
- \* Réparations des DAI , DCR et Ken-Dos
- \* Logithèque Ken-Dos , DCR , K7 audio

**D . C . A . :**

- \* Créations et réalisations de cartes HARD
- \* Logithèque VC 1541 Allemand

**MODALITES D' ADHESION**

200 FF pour la France Metropolitaine  
235 FF pour la C.E.E.

Pour tout paiement : par chèque ou mandats

Pour l'étranger : uniquement par mandat postal international  
Rédigés au nom du président trésorier Mr. DELANNAY Bruno.

Par virement sur le compte du Club :

Banque Crédit Lyonnais , 247 Cours GAMBETTA , 33 400 Talence  
Intitulé : International Dai Club Bordeaux , No = 79 233 H

Soit au président trésorier de D.C.A. : Mr. D'ARANTES Luc.

**FRAIS POSTAUX :**

		En cas d'achats groupés :	
	FRANCE	EUROPE	
T1 100g	3,50 FF	7,60 FF	T1 + T1 = T2
T2 250g	7,10 FF	14,30 FF	T1 + T2 = T3
T3 500g	10,30 FF	27,30 FF	T2 + T2 = T3
T4 1Kg	14,60 FF	47,80 FF	T2 + T3 = T4
			T3 + T3 = T4

**COPYRIGHT :**

Les articles publiés n'engagent que la responsabilité de leur auteur. Toute reproduction, même partielle, de ce magazine est interdite sans l'accord de l'éditeur responsable.

AVIS AUX KEN - DOSSEURS

DE TRES GROS PROBLEMES  
D' APPROVISIONNEMENT EN  
C. I. 8877. A. FUJITSU  
(contrôleur des disquettes)  
M'ONT OBLIGE A UN ACHAT  
DE PLUSIEURS DE CES CIRCUITS  
  
IL M'EN RESTE QUELQUES UNS A  
VENDRE ...

MISE A PRIX :

135FF port compris.

APPELEZ VITE :

International DAI Club Bordeaux  
C/O Bruno Delannay  
Res. Les Acacias St.B3  
Avenue de Saige  
33600 PESSAC (FRANCE)

☎ (56) 45.87.70

DAICLIC 12

1	SOMMAIRE	REDACTION
2	EDITO	I.D.C.Bx.
3	LE MUST DES GRATUITS	I.D.C.Bx.
7	LIBRAIRIE	I.D.C.Bx.
8	D.C.A. INFOS	D.C.A.
9	R.S. 232 x 2	Bruno DELANNAY
16	COMMANDES X-BASIC	Luc D'ARANTES
18	X-BUS REALISATION	DECUYPER
20	Tass' Data	Claudius MORIN
25	Resto' Rang	Claudius MORIN
29	DATA sans DIESE	Claudius MORIN
31	CLAVIER de rechange	Thierry BRIS
33	TIMMERS	Sébastien DUBOURG
36	INTERFACE MIDI	Hardy STROBEL
38	8 BIT - 8 KANAL A/D WANDLER	Hardy STROBEL
40	EPROM PROGRAMMER	Hardy STROBEL
43	PASCAL suite ( 2me partie )	Philippe CASIER
52	INVMAT Invention de Matrices	H. SAMAIN
53	LOGICIELS pub	I.D.C.Bx.
58	SOS LECTEURS	REDACTION
59	PETITES ANNONCES	REDACTION

éditoéditoéditoéditoéditoé  
 d d  
 i i  
 t **EDITORIAL** t  
 o o  
 éditoéditoéditoéditoéditoé

Diverses remarques :

D'abord un appel à tous pour envoyer des articles !! Il ne reste plus une seule page à paraître, et si la situation ne s'arrange pas pendant cet été, le prochain numéro de la revue sera plus que pauvre !!

Ensuite, je prie d'excuser le temps de plus en plus long mis à vous répondre : parfois un mois pour certains, mais il faut savoir que je suis désormais seul à Bordeaux à assurer les différentes activités : élaboration de la maquette de la présente revue, démarches auprès des imprimeurs, envoi postal, service télématique, réparations des DAI, Logithèque, Librairie etc... et tout ceci me prend un temps pas possible.

Vous pouvez constater dans le présent numéro l'apparition d'articles envoyés par certains de nos membres Allemands. Ne parlant pas un traître mot de cette langue, je les publie tels quels.

Les personnes désirant des précisions sur ces articles ou les possibilités d'achat des interfaces décrites (et bien d'autres) sont priées de contacter:

Mr.Hardy STROBEL, Neuselsbrunn 51, 8500 nürnberg 50, tel. (0911) 86.30.80 soit en Allemand soit en Anglais.

Et pour notre service télématique !! (vous en connaissez beaucoup des petits clubs qui ont un service télématique accessible en 3615 ?)

ALIENOR nous ouvre enfin ses portes !! tous ceux qui ont demandé un code d'accès gratuit le recevront dans quelques jours à peine. Déjà des informations sont chargées, vous pouvez me contacter directement dans ma BAL, je vous répondrait en priorité.

J'attends vos suggestions sur les infos que vous désirez trouver concernant votre club.

A dans trois mois , et bonnes vacances

mustmustmustmustmustmustmustmustmustm  
 u  
 s Le **MUST** des logiciels **GRATUITS** s  
 t t  
 mustmustmustmustmustmustmustmustmustu

Si vous désirez un de ces logiciels

Il vous suffit d'envoyer votre demande à I.O.C.Bordeaux.

Pour éviter les grandes pertes de temps dues à la copie de ces programmes , ils sont d'ores et déjà enregistrés sur K7 audio prêts à être envoyés , le tout vous étant proposé à prix coutant !!!

Donc pour chaque programme demandé , envoyer avec votre demande un chèque de 45FF destiné à couvrir le prix de la K7 de l'enveloppe ,de l'éventuelle copie du mode d'emploi et des frais d'envoi.

**NOUVEAU CE TRIMESTRE :**

Dans la série **\* UTILITAIRES \***

a) **BOURSI COTAGE** :

Il permet de stocker, comparer, surveiller et représenter sous forme de graphisme des actions boursières. Indispensable de nos jours...

Language = Basic adapté Ken:-Dos ou DCR (préciser la version voulue)

b) **THEME ASTRAL** :

Il calcule tous les paramètres de vos positions planétaires, et vous dessine votre carte du ciel. Très bien fait et précis.

Dans la série **\* DESSIN \***

Une compilation des plus belles images sur le DAI dont la plupart des images digitalisées par caméra vidéo ... surprenant et agréable.

Dans la série **\* JEUX \***

a) **DEFENDER** :

Un jeu d'arcade rapide et haut en couleurs. Un classique qui vous distraira des heures.



DEJA VU :

1/COMMUNICATION :

a) *L' EMULATEUR MINITEL* :

Auteur = Sébastien Dubourg  
Language = Source assembleur en SPL , ou en LM

b) *INTERDAI* :

Auteur = Sébastien Dubourg  
Language = Source assembleur en SPL , ou en LM

2/DESSIN :

a) *GRAPHIDEO* :

Auteur = ?  
Language = LM + Basic

b) *CLIO* :

Auteur = Multisoft  
Language = LM + Basic

3/UTILITAIRES :

a) *COPY K7 <=> DCR* :

Auteur = Sébastien Dubourg et Bruno Delannay  
Language = Source assembleur en SPL , ou en LM

b) *COPIEUR UNIVERSEL* :

Auteur = Sébastien Dubourg  
Language = Source assembleur en SPL , ou en LM

c) *COPIEUR KENDOS* :

Auteur = ?  
Language = LM

d) *SUPERDIRECTORY* :

Auteur = Sébastien Dubourg  
Language = Source assembleur en SPL , ou en LM

4/TRAITEMENT DE TEXTE :

*DAIWORDSTAR* :

Auteur = Uwe Wienkop (auteur de l'excellent SUPERBASE)  
Language = LM préciser version

Fourni avec les 12 pages correspondant à la version Française de ce programme.

5/\*\* JEUX \*\* :

a) *OTHELLO DAIKOND* :

Auteur =  
Language = LM

b) *ULTIME PLANETE* :

Fourni avec son mode d'emploi de 6 pages

Auteur = Eric Dubreuil  
Language = LM + Basic

c) *EXPANSION* :

Fourni avec son mode d'emploi de 3 pages

Auteur = Eric Dubreuil  
Language = LM + Basic

d) *SYMORGIA* :

Auteur = Mr.Stransky  
Language = Basic

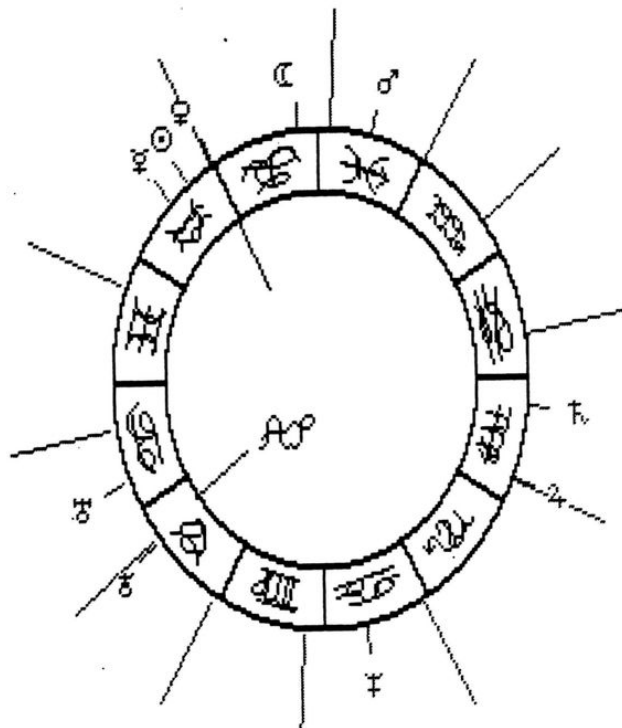
e) *GRIAR* :

Auteur = Mr.Stransky  
Language = Basic

Nous essaierons de préparer pour chaque revue de nouveaux logiciels d'accès gratuit afin de diffuser au maximum cette pratique.

**Demands à adresser à :**

International DAI Club Bordeaux  
C/O Bruno Delannay  
Res. Les Acacias St.B3  
Avenue de Saige  
33600 PESSAC (FRANCE)



```

librairielibrairielibrairi
i                               e
b                               l
r   LIBRAIRIE                 i
a                               b
irielibrairielibrairielibr

```

**\*\*\* Rappelons les livres proposés :**

Le secteur bibliothèque du club se développe grâce à la coopération de l'un de nos membres qui a traduit de l'Anglais en Français, de façon très professionnelle, le manuel complet du KEN-DOS.

\*En raison du don bénévole de ce travail au club, ce livre vous est offert à prix coutant pour le prix de 40FF port non compris. \* \* MANUEL KEN-DOS en FRANCAIS \* \* (pour le port rajouter le tarif postal T2)

\*Vu le nombre des demandes, une réédition du 'TOS COMMENTE' a été faite. Que tous ceux qui n'avaient pu l'obtenir l'an dernier me réécrivent. Le prix lui n'a pas augmenté : 150FF (pour le port rajouter le tarif postal T2)

\*Toujours disponibles : le livre indispensable au programmeur en assembleur, du club FLAMMAND associé DAINAMIC Le FIRMWARE MANUEL écrit en Anglais: 250FF (pour le port rajouter le tarif postal T2)

Et les deux tomes de 'Apprendre à programmer sur le DAI' 50FF pièce (pour le port rajouter le tarif postal T2)



### Deux types de montages :

\* \* \* Un à visée générale, qui commute vraiment tous les signaux de deux RS 232 du DAI, et sur laquelle on peut brancher n'importe quel type de périphérique RS 232.

Il utilise un C.I. MOS 74 HC 244 trois états, deux diodes de fond de poubelle (style 1N 914 ou n'importe quoi d'autre), et 8 résistances de 1 Mégohm.

#### C'est LE doubleur de 232

(mais si vous voulez brancher dessus votre minitel il vous faudra le munir de l'interface décrite dans DAICLIC 4 !)

\* \* \* Un deuxième montage, pour ceux qui veulent se servir de ce montage **UNIQUEMENT** pour connecter leur DAI sur une **IMPRIMANTE** et leur **MINITEL**.  
Ce deuxième montage à l'avantage de réduire à sa plus simple expression la connection avec le minitel, car il intègre les modifications de signal que demande cette connection (c.f. le schéma de connection MINITEL-DAI paru dans DAICLIC 4)

#### C'est l'interface 2 en 1

Prudence cependant car le C.I. utilisé ici, afin de réduire le coût à son expression la plus ridicule, est un 74 LS 244, version TTL du MOS précédent.

Rappelons brièvement que les C.I. MOS fonctionnent en +12 Vcc, alors que les TTL demandent du +5 Vcc. Les signaux de RS 232 vont normalement de -12Vcc à + 12Vcc. Le DAI tolère de 0Vcc à + 12 Vcc, donc aucun problème quand vous utilisez le MOS mais un signal TTL de +5 Vcc sera trop peu élevé pour être reconnu, il faudra donc relever ce signal à +12 Vcc grâce à des petits transistors ce qui dans le fond n'est pas trop gênant puisque la connection avec le MINITEL exige déjà des transistors de remise à niveau du signal transmis.

On peut donc mettre à profit un simple 74 LS 244 très bon marché et quelques transistors de fond de poubelle style 2N 2222 pour faire cette interface très spécialisée à moindre frais.

Tout serait pour le mieux si l'on oubliait qu'un C.I. TTL digne de ce nom, s'alimente en +5 Vcc !!. Or, afin de ne pas tout compliquer par des fils inutiles, il est préférable de puiser l'alimentation de notre montage à même la prise RS 232 sur laquelle on se branche.

Deux solutions alors :

\* \* La 'propre'  
Qui consiste à utiliser une des bornes de la RS 232 du DAI non connectée, pour y amener un fil d'alimentation du +5 Vcc du DAI pour le C.I. TTL.  
Cette solution bien que la seule logiquement acceptable a pour gros inconvénient d'exiger une intervention au niveau du DAI lui même, rendant l'interface connectable aux seuls DAI modifiés. (je ne l'ai même pas essayée...)

\* \* La 'système D'  
Qu'utiliseront les sans peur de la bidouille (ou les plus inconscients...) c'est à dire la solution que j'ai préféré et qui marche très bien...  
Elle consiste à brancher quand même ce pauvre TTL sur le +12 Vcc fourni sur la prise RS 232 du DAI en sachant que c'est à travers une résistance en série de 220 Ohm et en faisant en sorte que les autres résistances de ce montage chutent au travers des transistors suffisamment de tension, pour que le TTL résiste à cette alimentation, mais pas trop pour que les signaux soient suffisants à la RS 232 ...

\* \* Je ne citerais que pour mémoire la possibilité de construire une alimentation externe +12 Vcc +5 Vcc, qui annulerait le seul avantage de ce montage par rapport au précédent : sont coût !! et sa simplicité.

#### POUR FINIR

Précisons que l'interrupteur symbolisé sur les deux schémas doit être connecté en parallèle sur deux interrupteurs :

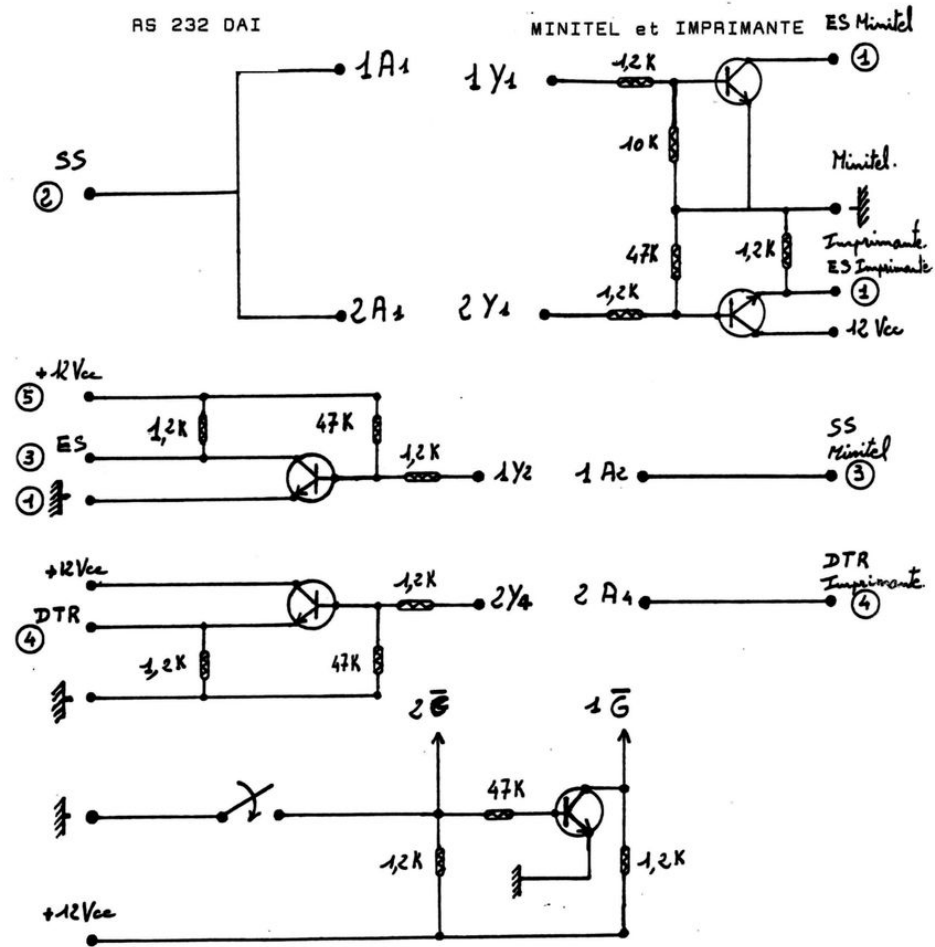
Un manuel simple, qui permet la commutation à la main de l'un ou l'autre des périphériques.

Et deux fils allant sur une prise connectable à l'interface K7, bornes 2 et 3, si vous désirez exploiter les possibilités de commutation par soft.  
N.B. L'interrupteur manuel doit alors être en position ouverte pour permettre l'action de l'interrupteur par soft.

On n'oubliera pas de connecter sur toutes les prises les différents signaux sans oublier les fils de masse et +12 Vcc

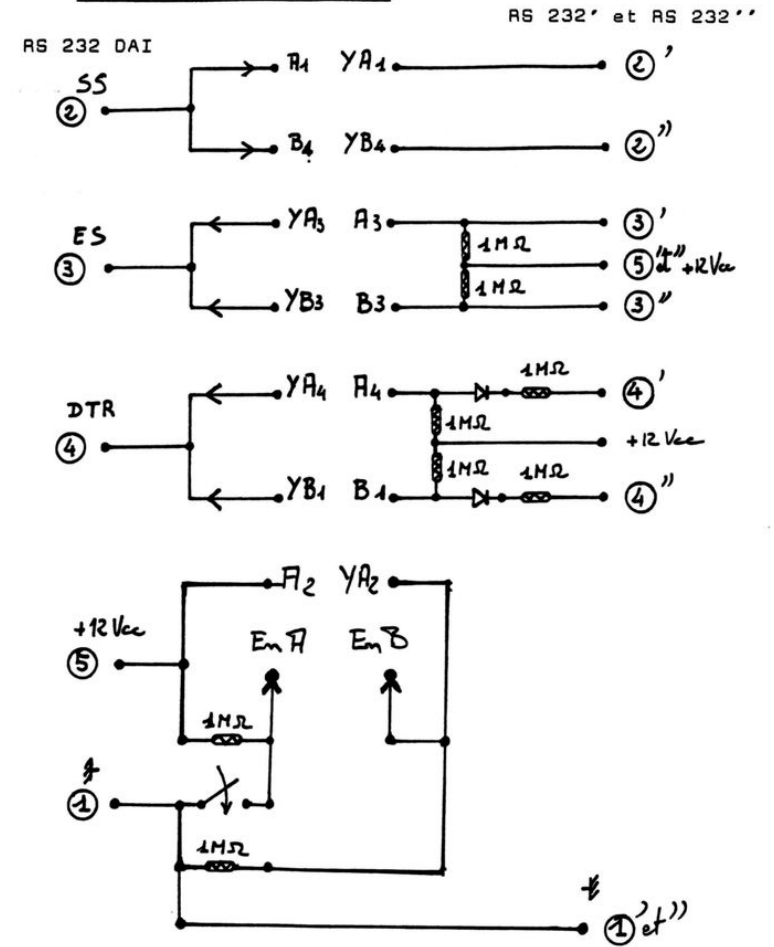
Les chiffres indiqués en face des bornes sur les schémas correspondent à ceux indiqués dans le manuel pour la prise RS 232.

C'est l'interface 2 en 1



Transistors = 2 N 2222  
C.I. = 74 LS 244 (TTL)

C'est LE doubleur de 232



Diodes = 1N 914  
C.I. = 74 HC 244 (MOS)

### TYPES SN54S240, SN54S241, SN74S240, SN74S241 BUFFERS/LINE DRIVERS/LINE RECEIVERS WITH 3-STATE OUTPUTS

REVISED AUGUST 1977

#### recommended operating conditions

PARAMETER	SN54S'			SN74S'			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, $V_{CC}$ (See Note 1)	4.5	5	5.5	4.75	5	5.25	V
High-level output current, $I_{OH}$			-12			-15	mA
Low-level output current, $I_{OL}$			48			54	mA
Operating free-air temperature, $T_A$ (See Note 3)	-55	125	0	70			°C

NOTES: 1. Voltage values are with respect to network ground terminal.  
2. An SN54S241J operating at free-air temperature above 116°C requires a heat sink that provides a thermal resistance from case to free-air,  $R_{\theta CA}$ , of not more than 40°C/W.

#### electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS <sup>1</sup>	'S240			'S241			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
$V_{IH}$ High-level input voltage		2			2			V
$V_{IL}$ Low-level input voltage		0.8			0.8			V
$V_{IK}$ Input clamp voltage	$V_{CC} = \text{MIN.}$ , $I_I = -18 \text{ mA}$	-1.2			-1.2			V
Hysteresis ( $V_{T+} - V_{T-}$ )	$V_{CC} = \text{MIN.}$	0.2, 0.4			0.2, 0.4			V
$V_{OH}$ High-level output voltage	SN74S'	$V_{CC} = \text{MIN.}$ , $V_{IH} = 2 \text{ V.}$ $V_{IL} = 0.8 \text{ V.}$ , $I_{OH} = -1 \text{ mA}$	2.7		2.7		V	
	SN54S' and SN74S'	$V_{CC} = \text{MIN.}$ , $V_{IH} = 2 \text{ V.}$ $V_{IL} = 0.8 \text{ V.}$ , $I_{OH} = -3 \text{ mA}$	2.4, 3.4		2.4, 3.4			
	SN54S' and SN74S'	$V_{CC} = \text{MIN.}$ , $V_{IH} = 2 \text{ V.}$ $V_{IL} = 0.5 \text{ V.}$ , $I_{OH} = \text{MAX.}$	2		2			
		$V_{CC} = \text{MIN.}$ , $V_{IH} = 2 \text{ V.}$ $V_{IL} = 0.8 \text{ V.}$ , $I_{OL} = \text{MAX.}$	0.55		0.55			
$V_{OL}$ Low-level output voltage	$V_{CC} = \text{MIN.}$ , $V_{IH} = 2 \text{ V.}$ $V_{IL} = 0.8 \text{ V.}$ , $I_{OL} = \text{MAX.}$	0.55		0.55		V		
$I_{OZH}$ Off-state output current, high-level voltage applied	$V_{CC} = \text{MAX.}$ , $V_{IH} = 2 \text{ V.}$	50		50		$\mu\text{A}$		
$I_{OZL}$ Off-state output current, low-level voltage applied	$V_{CC} = \text{MAX.}$ , $V_{IL} = 0.8 \text{ V.}$	-50		-50		$\mu\text{A}$		
$I_I$ Input current at maximum input voltage	$V_{CC} = \text{MAX.}$ , $V_I = 5.5 \text{ V.}$	1		1		mA		
$I_{IH}$ High-level input current, any input	$V_{CC} = \text{MAX.}$ , $V_I = 2.7 \text{ V.}$	50		50		$\mu\text{A}$		
$I_{IL}$ Low-level input current	Any A	-400		-400		$\mu\text{A}$		
	Any G	-2		-2		mA		
$I_{OS}$ Short-circuit output current*	$V_{CC} = \text{MAX.}$	-50	-225	-50	-225	mA		
$I_{CC}$ Supply current	Outputs high	SN54S'	80	123	95	147	mA	
		SN74S'	80	125	95	160		
	Outputs low	SN54S'	100	145	120	170		
		SN74S'	100	150	120	180		
	Outputs disabled	SN54S'	100	145	120	170		
		SN74S'	100	150	120	180		

<sup>1</sup>For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.  
\*All typical values are at  $V_{CC} = 5 \text{ V.}$ ,  $T_A = 25^\circ\text{C.}$   
\*Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

#### switching characteristics, $V_{CC} = 5 \text{ V.}$ , $T_A = 25^\circ\text{C.}$

PARAMETER	TEST CONDITIONS	'S240			'S241			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
$t_{PLH}$ Propagation delay time, low-to-high-level output	$C_L = 50 \text{ pF.}$ $R_L = 90 \Omega.$ See Note 4	4.5, 7			6, 9			ns
$t_{PHL}$ Propagation delay time, high-to-low-level output		4.5, 7			6, 9			ns
$t_{PZL}$ Output enable time to low level		10, 15			10, 15			ns
$t_{PZH}$ Output enable time to high level		8.5, 10			8, 12			ns
$t_{PLZ}$ Output disable time from low level	$C_L = 5 \text{ pF.}$ $R_L = 90 \Omega.$ See Note 4	10, 15			10, 15			ns
$t_{PHZ}$ Output disable time from high level		6, 9			6, 9			ns

NOTE 4: Load circuit and voltage waveforms are shown on page 3-10.

### MC54/74HCT244

#### HIGH-PERFORMANCE CMOS LOW POWER COMPLEMENTARY METAL SILICON GATE

OCTAL 3-STATE NONINVERTING BUFFER/LINE DRIVER/LINE RECEIVER  
(WITH LSTTL-COMPATIBLE INPUTS)



ORDERING INFORMATION  
M Series: -50°C to +125°C  
MC54HC1244J (Ceramic Package)  
MC74HC1244J (Ceramic Package)  
N Series: -40°C to +85°C  
MC54HC1244N (Plastic Package)  
MC74HC1244N (Plastic Package)

#### FUNCTION TABLE

Inputs	Outputs
Enable A, Enable B	A, B, YA, YB
L	L, H
H	H, Z

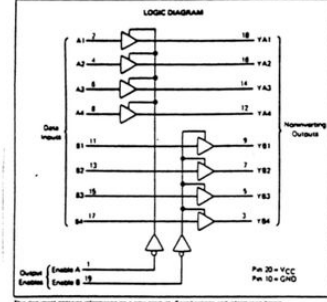
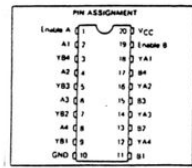
Z = High Impedance

#### Advance Information

##### OCTAL 3-STATE NONINVERTING BUFFER/LINE DRIVER/LINE RECEIVER

The MC54/74HCT244 is identical in pinout to the LS244. The device inputs are compatible with standard CMOS outputs, with pullup resistors, they are compatible with LSTTL outputs.  
The octal noninverting buffer/line driver/line receiver is designed to be used with 3-state memory address drivers, clock drivers, and other bus-oriented systems. The device has noninverting outputs and two active-low output enables.  
The HC244 is similar in function to the HC240 and HC241.

- Low Power Consumption Characteristics of CMOS Devices
- Output Drive Capability: 15 LSTTL Loads Minimum
- Operating Speeds Similar to LSTTL
- Wide Operating Voltage Range: 2 to 6 Volts
- Low Input Current: 1  $\mu\text{A}$  Maximum
- Low Quiescent Current: 80  $\mu\text{A}$  Maximum (74HC Series)
- High Noise Immunity Characteristics of CMOS Devices
- Diode Protection on All Inputs



### TYPES SN54LS240, SN54LS241, SN54LS244, SN54S240, SN54S241, SN74LS240, SN74LS241, SN74LS244, SN74S240, SN74S241 OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS

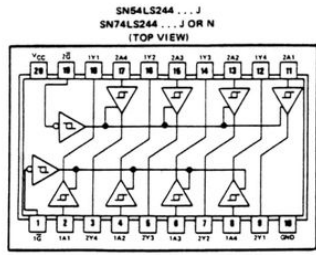
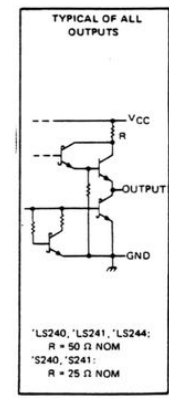
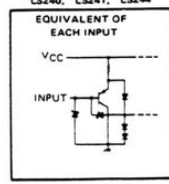
	Typical $I_{OL}$ (Sink Current)	Typical $I_{OH}$ (Source Current)	Typical Propagation Delay Times	Typical Enable/Disable Times	Typical Power Dissipation (Enabled)
SN54LS'	12 mA	-12 mA	10.5 ns (Inverting), 12 ns (Noninverting)	18 ns	130 mW (Inverting), 135 mW (Noninverting)
SN74LS'	24 mA	-15 mA	10.5 ns (Inverting), 12 ns (Noninverting)	18 ns	130 mW (Inverting), 135 mW (Noninverting)
SN54S'	48 mA	-12 mA	4.5 ns (Inverting), 6 ns (Noninverting)	9 ns	450 mW (Inverting), 538 mW (Noninverting)
SN74S'	64 mA	-15 mA	4.5 ns (Inverting), 6 ns (Noninverting)	9 ns	450 mW (Inverting), 538 mW (Noninverting)

- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- P-N-P Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

#### description

These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical G (active-low output control) inputs, and complementary G and G inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS' and SN74S' can be used to drive terminated lines down to 133 ohms.

#### schematics of inputs and outputs



Les commandes du X-BASIC : partie 1.

Voici une serie de petits articles . Ils respectent l'ordre du manuel.

RUN  
RUN n°de ligne  
RUN variable  
La traditionnelle commande RUN admet maintenant une variable et est programmable.

IF condition THEN commande :ELSE commande  
                          n°ligne          n°ligne

Enfin sur DAI l'interessante commande basic if then else.

page 78 du manuel vert du DAI : basic Indata  
100 IF A=B THEN 110  
105 A=A+B  
110 PRINT A,B

en XBASIC ça devient:  
100 IF A=B THEN 110:ELSE A=A+B  
110 PRINT A,B

IF condition commande

Il s'agit là d'une puissante commande analogue à la IF GOTO du basic Indata.Mais qui supporte pratiquement toutes les commandes basic, x-basic, voire basic V1.2  
par ex: IF RETURN; IF GOSUB; IF PRINT;IF CIRCLE.....

Le programme execute la commande si la condition est vraie.

IF condition THEN commande  
                          inchangée.

RQ: plusieurs commandes IF ne peuvent s'imbriquées:  
IF A=0 THEN IF B<>0 THEN W=0:ELSE ?"IMPOSSIBLE"

MPMOVE ladr hadr dadr

permet sous basic d'obtenir l'équivalent de la commande M sous utilitaire.(déplacement d'image par exemple)  
cette commande peut également permettre le transfert vers les rams 6264 auxilliaires, mises éventuellement sur la carte X-bus:

MPMOVE ladr hadr dadr bank

VC1541 DCA VC1541 XBASIC DCA VC1541 XBASIC DCA VC1541 XBASIC DCA

MPINIT ladr hadr n  
                  identique à F sous UT.

BCALL bankadr  
BCALL bankadr, var  
                  analogue à CALLM mais sur les banks. Des précisions sur ces commandes dans un prochain article sur l'utilisation de ram 6264 sur la carte Xbus.

\$ nom  
permet d'appeler des programmes utilitaires en EPROM par ex : \$ALLADER

\$RESET  
reset des pointeurs banks

\$USR n  
0<n<=41      appel d'un canal utilisateur.

IF=OFF  
désactive les commandes xbasic avec IF

IF=ON  
reactive les nouvelles commandes xbasic avec IF  
IF=on à la mise sous tension

cela permettrait de transcrire un programme xbasic sans ELSE en programme basic V1.1. Il faut également qu'il n'y ai pas de commande xbasic style WHILE.....

NCU=OFF  
desactive completement le xbasic, les commandes xbasic provoquent alors des "SYNTAXE ERROR".  
Il est impossible de travailler en xbasic tant que l'on n'a pas réactivé avec:  
NCU=ON

.....suite au prochain article.....  
.....d'Arantes Luc pour D.C.A.....

VC1541 DCA VC1541 XBASIC DCA VC1541 XBASIC DCA VC1541 XBASIC DCA



Chaufontaine, le 21 mars 1987

Maurice DECUYPER  
rue A. Dumont 11  
4601 CHAUFONTAINE  
Tél: 041/65.75.72  
Belgium

Cher Luc,

Nous avons, quelques amis et moi, effectivement réalisé une carte XBASIC beaucoup plus soignée et plus petite que celle de Mr TEGETHOFF.

Afin de conserver les meilleures relations avec nos amis allemands, nous ne désirons pas envoyer directement les schémas.

Le Soft (XBASIC, COMDOS, SGT...) est encore plus délicat à distribuer car il appartient aux allemands. Ils sont donc détenteurs des fichiers sources.

Ainsi, nous avons déjà inclus une participation pour Mr TEGETHOFF et Mr WIENKOP.

Il vous sera donc possible de leur demander les éproms pour des versions particulières éventuelles.

Il serait possible de vendre la carte XBASIC nue (circuit imprimé seul).

Cependant, cette solution n'est pas souhaitable car nous ne pouvons garantir que l'exactitude du schéma de la plaque.

Le prix de la carte XBASIC montée et testée avec une éprom 27256 est de 8200 FB.

Nous avons prévu la carte XBASIC en fonction d'une autre petite carte pour le basic V1.2 (enfichable très simplement sur les sockets des anciennes roms).

Vous pouvez obtenir cette petite carte montée et testée avec les éproms du V1.2 pour 2100 FB.

L'ensemble des deux cartes était accessible pour 10.000 FB.

Les circuits imprimés sont des doubles faces mais les trous métallisés étant hors de prix (excepté par 100 pièces !), le montage est assez délicat.

Sur l'éprom fournie avec la carte XBASIC nous placerons la version que nous possédons du XBASIC, c'est à dire le XBASIC complet, le COMDOS (pour drive Commodore), les TED-Tools, le SGT, le SPU et le MDCR.

Toutes les connections s'effectuent via de petits connecteurs propres et fiables.

Tous les choix à effectuer sur les cartes (le type d'éprom utilisé par ex.) se font par Jumpers.

Ainsi, on peut sans problème placer les roms du basic V1.1 ou les éproms du V1.2 dans les mêmes supports sans "monticules" de sockets (comme dans la solution allemande).

Nous avons utilisé ce système suite aux nombreux problèmes de mauvais contact avec le système allemand (surtout lors des transports).

Malgré ce prix modique, le matériel utilisé est de toute première qualité. L'ensemble obtenu est parfaitement fiable, propre et durable.

Je peux également, par la suite, fournir toutes les explications nécessaires à l'installation de softs (tel qu'un traitement de texte) sur les éproms de la carte XBASIC (3 \* 32 K restent disponibles).

Le formidable UDOS de Wienkop et le CP/M sont disponibles aussi sur éprom (voir Dainamic n° 33 page 30, mars 86) offrant des possibilités et une rapidité extraordinaire.

Je reste à votre disposition pour tous renseignements complémentaires.

*Maurice Decuyper*





# tass'data

Claudius Morin

Diminuer la longueur d'un programme BASIC c'est surtout gagner du temps au chargement, même pour les possesseurs de lecteurs rapides.

Les programmes graphiques et musicaux contiennent d'interminables lignes de DATA.

Elles sont généralement écrites pour que le programmeur s'y retrouve. Ainsi nous pouvons avoir des lignes avec une seule donnée. Quelle perte de place !

Je vous propose donc d'utiliser TASS'DATA, la routine suivante, lorsque le programme est bien au point.

Vous allez retrouver vos données rassemblées dans un minimum de lignes.

Une obligation tout de même, toutes les lignes de DATA doivent se trouver à la fin du programme sans autre instruction intercalée. La routine le vérifie et donne un message d'erreur s'il y en a.

Les lignes de DATA contiennent au maximum 122 octets (caractères ou séparateurs). L'analyse de la position des séparateurs (virgule ou fin de ligne) permet de ne pas couper les données en deux.

La table des symboles est recopiée à la suite des nouvelles lignes. Ce décalage est possible puisque les étiquettes des variables sont repérées relativement au début de la table des symboles dont l'adresse est contenue dans les pointeurs #2A1/2A2, remis à jour par la routine.

Comment s'en servir ?

Chargez le programme contenant les DATA à tasser.

Commandez CLEAR#200, pour réserver la place de la routine.

En utilitaire, chargez la routine TASS'DATA.

Revenez au BASIC.

Lancez la routine par CALLM#2FO.

J'ai gagné 400 octets sur un programme de dessin de la planisphère !!

SPL V1.1 PAGE 1

```

0000 ;
0000 ;
0000 TITL 'TASS DATA'
0000 ; Claudius Morin
0000 ;
0000 @=029F DEBBAS EQU 29FH ;Debut du basic
0000 @=02A1 DEBSYM EQU 2A1H ;Debut table symboles
0000 @=02A3 DEBLIB EQU 2A3H ;Debut memoire libre
0000 @=DAD4 ECRLIG EQU 0DAD4H ;Ecrit une ligne
0000 @=DE14 CPHLDE EQU 0DE14H ;Compare HL et DE
0000 @=DE39 ADLISU EQU 0DE39H ;HL=adr. ligne suivante
0000 ;
0000 ; ORG 2FOH
02F0 ; ;
02F0 ; ; Debut et fin de routine
02F0 ; ;
02F0 C5 DEBUT PUSH B
02F1 CD4F03 CALL INIT
02F4 C1 POP B
02F5 C9 FIN RET
02F6 ; ;
02F6 ; ; Memoires auxiliaires
02F6 ; ;
02F6 00 INDATA DB 0H ;FF si ligne DATA
02F7 0000 ADECRI DW 0H ;Adr. ecriture courante
02F9 00 CPTOCT DB 0H ;Octets restants, en ecriture
02FA 0000 ADCINQ DW 0H ;Adr. 5eme octet ligne, en ecriture
02FC 0000 NUMLIG DW 0H ;Numero ligne, en ecriture
02FE 00 GUIMET DB 0H ;Guillemet, FF=ouverture, O=fermeture
02FF 0000 OCTSEP DW 0H ;Octets restants, apres un separateur
0301 0000 ADSEPA DW 0H ;Adr. separateur en lecture
0303 00 FNMAUX DB 0H
0304 ; ;
0304 ; ; Transfert d'un octet
0304 ; ;
0304 7E LECECR MOV A,M
0305 02 STAX B
0306 23 INX H
0307 03 INX B
0308 C9 RET
0309 ; ;
0309 ; ; Mise a jour des 5 premiers octets
0309 ; ; d'une ligne de DATA
0309 ; ;
0309 2AFA02 MISJOU LHLD ADCINQ
030C 3E7A MVI A 7AH ;122 DATA max.
030E 91 SUB C
030F 77 MOV M,A ;Nbre d'octets de DATA
0310 2B DCX H
0311 36A2 MVI M 0A2H ;Code DATA
0313 2B DCX H
0314 EB XCHG
0315 2AFC02 LHLD NUMLIG ;Numerote la ligne
0318 EB XCHG
0319 73 MOV M,E
031A 2B DCX H
031B 72 MOV M,D
031C 2B DCX H
031D C604 ADI 4H
031F 77 MOV M,A ;Nbre d'octets ligne basic
0320 13 INX D
0321 EB XCHG
0322 22FC02 SHLD NUMLIG ;Numero ligne suivante
0325 EB XCHG
0326 CD39DE CALL ADLISU
0329 1600 MVI D 0H
032B 1E04 MVI E 4H
032D 19 DAD D
032E 22FA02 SHLD ADCINQ
0331 23 INX H
0332 22F702 SHLD ADECRI
0335 AF XRA A
0336 32FE02 STA GUIMET
0339 ; ;
0339 E5 DATMAX PUSH H

```

SPL V1.1 PAGE 2 TASS DATA

```

033A 0E7A MVI C 7AH ;Max octets ligne DATA
033C 21F902 LXI H CPTOCT
033F 71 MOV M,C
0340 E1 POP H
0341 C9 RET
0342 ;
0342 ; Rencontre d'un separateur
0342 ;
0342 E5 MEMSEP PUSH H
0343 60 MOV H,B
0344 69 MOV L,C
0345 22FF02 SHLD OCTSEP ;Nbre octets restants (lect. et escrit.)
0348 EB XCHG
0349 220103 SHLD ADSEPA ;Adr. dernier separateur
034C EB XCHG
034D E1 POP H
034E C9 RET
034F ;
034F ; Programme principal
034F ;
034F 21F602 INIT LXI H INDATA
0352 110303 LXI D FNMAUX
0355 AF MEMAUX ;Mise a 0 mem. auxiliaires
0356 77 MOV M,A
0357 23 INX H
0358 CD14DE CALL CPHLDE
035B C25503 JNZ MEMAUX
035E 2A9F02 LHL D DEBBAS
0361 1600 MVI D OH
0363 1E03 MVI E 3H
0365 4A MOV C,D
0366 7E AUTILIG MOV A,M
0367 B7 ORA A
0368 CA8403 JZ FNPGBA ;Saut si fin de programme
036B E5 PUSH H
036C 19 DAD D
036D 7E MOV A,M
036E E1 POP H
036F EEA2 XRI OA2H
0371 CA7F03 JZ MELIDA ;Saut si ligne DATA
0374 79 MOV A,C
0375 B7 ORA A
0376 CD2804 JNZ ECRTXT ;Saut si autre ligne apres ligne DATA
0379 CD39DE SUIMEM CALL ADLISU
037C C36603 JMP AUTILIG
037F 0EFF MELIDA MVI C OFFH
0381 C37903 JMP SUIMEM
0384 79 FNPGBA MOV A,C
0385 B7 ORA A
0386 C8 RZ ;Retour au BASIC si pas de DATA
0387 ;
0387 ; Validation terminee, debut lecture et eciture
0387 ;
0387 ;
0387 2A9F02 LHL D DEBBAS ;1ere ligne
038A 1600 LIGLEC MVI D OH
038C 1E03 MVI E 3H
038E 7E LIGSUI MOV A,M
038F B7 ORA A
0390 CA5A04 JZ DECSYM ;Saut si fin de programme
0393 E5 PUSH H
0394 19 DAD D
0395 7E MOV A,M
0396 E1 POP H
0397 EEA2 XRI OA2H
0399 CAA203 JZ UNDATA ;Saut si DATA
039C CD39DE CALL ADLISU
039F C38E03 JMP LIGSUI
03A2 ;
03A2 ; Une ligne de DATA
03A2 ;
03A2 E5 UNDATA PUSH H
03A3 57 MOV D,A
03A4 1E04 MVI E 4H
03A6 19 DAD D
03A7 01F602 LXI B INDATA

```

SPL V1.1 PAGE 3 TASS DATA

```

03AA 0A LDAX B
03AB B7 ORA A
03AC C2E703 JNZ DEBLEC ;Suite si 1ere ligne DATA
03AF 2F CMA
03B0 02 STAX B ;Memo. 1ere ligne
03B1 22FA02 SHLD ADCINQ
03B4 E3 XTHL
03B5 23 INX H
03B6 56 MOV D,M
03B7 23 INX H
03B8 5E MOV E,M
03B9 EB XCHG
03BA 22FC02 SHLD NUMLIG ;Numero 1ere ligne
03BD E3 XTHL
03BE CD3903 CALL DATMAX
03C1 23 INX H
03C2 22F702 SHLD ADECRI
03C5 2B DCX H
03C6 EB XCHG
03C7 1B DCX D
03C8 0600 MVI B OH
03CA 0EFF MVI C OFFH
03CC 2AA302 LHL D DEBLIB ;)Modification
03CF E5 PUSH H ;)de l'adr. basse
03D0 09 DAD B ;)de la table
03D1 22A302 SHLD DEBLIB ;)des symboles
03D4 44 MOV B,H
03D5 4D MOV C,L
03D6 E1 POP H
03D7 7E DECALT MOV A,M ;)Decalage
03D8 02 STAX B ;)de la table
03D9 2B DCX H ;)des symboles
03DA 0B DCX B ;)et des DATA
03DB CD14DE CALL CPHLDE
03DE C2D703 JNZ DECALT
03E1 23 INX H
03E2 0600 MVI B OH
03E4 0EFF MVI C OFFH ;HL=adr. en lecture
03E6 09 DAD B
03E7 D1 DEBLEC POP D
03E8 46 MOV B,M
03E9 23 INX H
03EA EB XCHG
03EB 21F902 LXI H CPTOCT
03EE 4E MOV C,M
03EF 21FE02 LEDATA LXI H GUIMET
03F2 1A LDAX D
03F3 FE22 CPI ;'
03F5 C2FB03 JNZ GUILME
03F8 7E MOV A,M
03F9 2F CMA ;)FF = ouverture guillemet
03FA 77 MOV M,A ;) O = fermeture
03FB AF GUILME XRA A
03FC BE CMP M
03FD 2AF702 LHL D ADECRI ;)Virgule = separateur
0400 C20904 JNZ ECDDATA ;)si non comprise
0403 1A LDAX D ;)entre guillemet
0404 FE2C CPI ;'
0406 CC4203 CZ MEMSEP
0409 1A ECDDATA LDAX D
040A 77 MOV M,A
040B 13 INX D
040C 23 INX H
040D 22F702 SHLD ADECRI
0410 05 DCR B
0411 C23804 JNZ NFNLEC ;Suite si fin lect. d'une ligne
0414 1A LDAX D
0415 B7 ORA A
0416 CA5104 JZ FINBAS ;Saut si fin prog.
0419 0D DCR C
041A CA5204 JZ FINECR ;Saut si 0 octet en escrit.
041D 3E01 MVI A 1H ;)Ne pas ecrire
041F B9 CMP C ;)de separateur
0420 CA5204 JZ FINECR ;)en fin de ligne
0423 362C MVI H ;)Ecriture du separateur

```

```

SPL V1.1 PAGE 4 TASS DATA
0425 1B DCX D
0426 04 INR B
0427 CD4203 CALL MEMSEP
042A 23 INX H
042B 0D DCR C
042C 22F702 SHLD ADECRI
042F 21F902 LXI H,C CPTOCT
0432 71 MOV M,C
0433 13 INX D
0434 EB XCHG
0435 C38A03 JMP LIGLEC
0438 ;
0438 0D ;NFNLEC DCR C
0439 C2EF03 JNZ LEDATA ;suite si fin ecrit. d'une ligne
043C 2AFF02 LHLD OCTSEP
043F 44 MOV B,H
0440 4D MOV C,L
0441 CD0903 CALL MISJOU
0444 2A0103 LHLD ADSEPA
0447 EB XCHG
0448 13 INX D
0449 05 DCR B
044A C2EF03 JNZ LEDATA ;suite si fin lect. d'une ligne
044D EB XCHG
044E C38A03 JMP LIGLEC
0451 ;
0451 ; Fin d'écriture
0451 ;
0451 OD FINBAS DCR C
0452 D5 FINECR PUSH D
0453 CD0903 CALL MISJOU
0456 E1 POP H
0457 C38A03 JMP LIGLEC
045A ;
045A ; Fin prog. BASIC, decalage table symbole
045A ;
045A EB DECSYM XCHG
045B 2AF702 LHLD ADECRI
045E 06FF MVI B OFFH
0460 0EFB MVI C OFBH
0462 09 DAD B
0463 44 MOV B,H
0464 4D MOV C,L
0465 EB XCHG
0466 CD0403 CALL LECECR
0469 EB XCHG
046A 60 MOV H,B
046B 69 MOV L,C
046C 22A102 SHLD DEBSYM
046F 2AA302 LHLD DEBLIB
0472 EB XCHG
0473 CD0403 ECRTAB CALL LECECR
0476 CD14DE CALL CPHLDE
0479 C27304 JNZ ECRTAB
047C 60 MOV H,B
047D 69 MOV L,C
047E 22A302 SHLD DEBLIB
0481 C9 RET
0482 ;
0482 ; Ecriture d'un texte
0482 ;
0482 218904 ECRTXT LXI H MESERR
0485 CDD4DA CALL ECRLIG
0488 C9 RET
0489 ;
0489 ; Message d'erreur
0489 ;
0489 457272 MESERR DB 'Erreur, les DATA doivent '
04A2 657472 DB 'etre regroupes en fin '
04B8 646520 DB 'de programme.'
04C5 00 DB OH
04C6 ;
04C6 END

```

# RESTO'RANG

- Claudius Morin -

Est-il possible d'utiliser les DATA de manière très différente ? Nous sommes habitués à l'instruction RESTORE ou à la routine RESTORE LINE qui rapproche nos DAI des autres machines.

Voilà, RESTO'RANG, une routine à trois entrées, qui offre d'autres facilités. Le pointeur de DATA peut se placer :

- au rang absolu N, compté à partir du premier DATA.
- au rang relatif, de -N à +N à partir du dernier DATA utilisé.
- après le Nième séparateur (caractère écrit comme un DATA pour servir de point de repère).

L'utilisation de ce PLM, dans un programme BASIC, demande le transfert d'informations, à l'appel de celui-ci, de la manière suivante :

A% = informations  
CALLM adresse, A%

Le contenu de la variable dépend de l'entrée de la routine. Seuls les deux derniers octets de la variable sont utilisés ainsi :

- rang absolu : accepte un nombre positif au plus égal à 32767.
- rang relatif : accepte un nombre compris entre -32767 et +32767. Le chiffre 0 permet une répétition du dernier DATA.
- séparateur : le 4ème octet contient le code ASCII du séparateur choisi, le 3ème octet contient l'occurrence (rang absolu du séparateur). Un exemple, pour obtenir la 4ème occurrence du séparateur \*, il faut écrire A%=#043A. Des séparateurs quelconques peuvent être utilisés simultanément. Ils doivent seulement être différents des caractères utilisés dans les DATA. Une information complémentaire peut les accompagner à condition qu'elle les précède, par exemple :

1010 DATA A, B, CD, \*, E, FG, 1&, G, H, 2&, IJ, K

où \*, & sont des séparateurs,

1, 2 devant & ne servant que de repères au programmeur.

Utilisez le principe du manuel page 53 pour stocker la routine dans une variable dimensionnée, A%(80), déclarée au début du programme BASIC, ainsi vous n'aurez plus qu'à assembler la routine à partir de #2F0.

Appelez chaque entrée ainsi :

CALLM#2F0,A% pour les séparateurs.  
CALLM#330,A% pour le rang relatif.  
CALLM#380,A% pour le rang absolu.

SPL V1.1 PAGE 1

```

0000 ;
0000 ; TITL 'RESTO RANG ou RESTORE RANG'
0000 ; ; Claudius Morin
0000 ;
0000 @=029F ; INDBAS EQU 29FH ;Indicateur debut prog. BASIC
0000 @=0123 INDDAT EQU 123H ;Indicateur de DATA
0000 @=DE39 LIGSUI EQU ODE39H ;Adresse ligne suivante
0000 @=DE14 CPHLDE EQU ODE14H ;Compare HL et DE
0000 ;
0000 ; ORG 2FOH
02F0 ; ;
02F0 ; ; Debut de la routine "Nieme separateur"
02F0 ; ;
02F0 ; ;
02F0 CD1004 DEBSEP CALL DEBCOM
02F3 3A2604 LDA CPTDAT+1H
02F6 B7 ORA A
02F7 C2FF02 JNZ BC1
02FA 3E01 MVI A 1H
02FC 322604 STA CPTDAT+1H
02FF 2A9F02 BC1 LHL D INDBAS
0302 CDCB03 BC5 CALL LIDATA
0305 D5 PUSH D ;Ligne suivante
0306 4F MOV C,A ;Nombre total d'octets
0307 CDEB03 BC4 CALL UNDATA
030A 2B DCX H
030B 78 MOV A,B
030C B7 ORA A
030D C21103 JNZ BC2 ;Suite si fin de ligne
0310 23 INX H
0311 7E BC2 MOV A,M
0312 23 INX H ;Adresse de la virgule
0313 EB XCHG
0314 212504 LXI H CPTDAT
0317 BE CMP M
0318 C22503 JNZ BC3 ;Suite si separateur
031B 3A2604 LDA CPTDAT+1H
031E 3D DCR A
031F CAB903 JZ DATFIN ;Saut si derniere occurrence
0322 322604 STA CPTDAT+1H
0325 EB BC3 XCHG
0326 78 MOV A,B
0327 B7 ORA A
0328 C20703 JNZ BC4 ;Suite si fin de ligne
032B E1 POP H
032C C30203 JMP BC5
032F 00 ; DB OH
0330 ; ;
0330 ; ; Debut de la routine "rang relatif"
0330 ; ;
0330 ; ;
0330 CD1004 DEBREL CALL DEBCOM
0333 210000 LXI H OH
0336 222704 SHLD NBUTIL
0339 2A9F02 LHL INDBAS
033C CDCB03 BCL4 CALL LIDATA
033F E5 PUSH H
0340 2A2401 LHL INDDAT+1H
0343 47 MOV B,A
0344 CD14DE CALL CPHLDE
0347 C25203 JNZ BCL1
034A 3A2301 LDA INDDAT
034D FFFF CPI OFFH
034F C25303 JNZ BCL2
0352 78 BCL1 MOV A,B
0353 E1 BCL2 POP H
0354 D5 PUSH D
0355 CDEB03 BCL3 CALL UNDATA
0358 EB XCHG ;)Incremente
0359 2A2704 LHL NBUTIL ;)le compteur
035C 23 INX H ;)de DATA utilises
035D 222704 SHLD NBUTIL
0360 EB XCHG
0361 78 MOV A,B
0362 B7 ORA A
0363 C25503 JNZ BCL3 ;Suite si fin de ligne

```

SPL V1.1 PAGE 2

RESTO RANG ou RESTORE RANG

```

0366 D1 POP D
0367 2A2401 LHL INDDAT+1H
036A CD14DE CALL CPHLDE
036D EB XCHG
036E C23C03 JNZ BCL4 ;Suite si dernier DATA utilise
0371 2A2704 LHL NBUTIL
0374 EB XCHG
0375 2A2504 LHL CPTDAT
0378 19 DAD D
0379 222504 SHLD CPTDAT
037C C38603 JMP RANABS
037F ; ;
037F 00 ; DB OH
0380 ; ;
0380 ; ; Debut de la routine "rang absolu"
0380 ; ;
0380 ; ;
0380 CD1004 DEBABS CALL DEBCOM
0383 2A2504 LHL CPTDAT
0386 ; ;
0386 OE09 RANABS MVI C 9H ;Code d'erreur
0388 7C MOV A,H
0389 B7 ORA A
038A FA2104 JM ERR ;Saut si rang<0
038D B5 ORA L
038E CA2104 JZ ERR ;Saut si rang=0
0391 CD0804 CALL RESTDA
0394 7C MOV A,H
0395 B5 ORA L
0396 3EFF MVI A OFFH ;)Equivalent de RESTORE
0398 2A9F02 LHL INDBAS ;)
039B CAC203 JZ RETBAS ;Saut si rang=1
039E CDCB03 BL2 CALL LIDATA
03A1 D5 PUSH D ;Debut ligne suivante
03A2 4F MOV C,A ;Garde le nombre d'octets
03A3 CDEB03 BL7 CALL UNDATA
03A6 EB XCHG
03A7 CD0804 CALL RESTDA
03AA EB XCHG
03AB 7A MOV A,D
03AC B3 ORA E
03AD CAB903 JZ DATFIN ;Saut si plus de DATA
03B0 78 MOV A,B
03B1 B7 ORA A
03B2 C2A303 JNZ BL7 ;Suite si fin de ligne
03B5 E1 POP H
03B6 C39E03 JMP BL2
03B9 E1 DATFIN POP H ;Adresse ligne suivante
03BA 78 MOV A,B
03BB B7 ORA A
03BC 2F CMA
03BD CAC103 JZ BL6
03C0 79 MOV A,C
03C1 90 SUB B ;A=#FF ou A=C-B
03C2 322301 BL6 RETBAS STA INDDAT
03C5 222401 INDDAT+1H
03C8 C1 POP B ;)Retour au BASIC
03C9 E1 POP H ;)
03CA C9 RET ;)
03CB ; ;
03CB ; ; Recherche des lignes DATA
03CB ; ;
03CB ; ;
03CB OE02 LIDATA MVI C 2H ;Code d'erreur
03CD 7E MOV A,M
03CE B7 ORA A
03CF CA2104 JZ ERR ;Erreur si fin de programme
03D2 54 MOV D,H
03D3 5D MOV E,L
03D4 CD39DE CALL LIGSUI
03D7 EB XCHG
03D8 23 INX H
03D9 23 INX H
03DA 23 INX H ;HL=Adr. code basic
03DB 7E MOV A,M
03DC FEA2 CPI OA2H ;Code DATA
03DE C2E703 JNZ BL1

```

```

SPL V1.1 PAGE 3 RESTO RANG ou RESTORE RANG
03E1 23          INX H
03E2 7E          MOV A,M      ;Nombre d'octets
03E3 229102     SHLD      291H
03E6 C9          RET
03E7 EB          XCHG
03E8 C3CB03     JMP          LIDATA
03EB           ;
03EB           ; Analyse des octets d'une ligne de DATA
03EB           ;
03EB 47          UNDATA MOV B,A      ;Compteur d'octets
03EC 1E00       MVI E      ;Memoire de guillemet
03EE 23          BL5      INX H
03EF 7E          MOV A,M      ;Octet de DATA
03F0 57          MOV D,A
03F1 FE22       CPI
03F3 7B          MOV A,E
03F4 C2F903     JNZ      BL3      ;Suite si guillemet
03F7 2F          CMA
03F8 5F          MOV E,A
03F9 B7          BL3      ORA A
03FA C20304     JNZ      BL4
03FD 7A          MOV A,D
03FE 05          DCR B
03FF FE2C       CPI
0401 C8          RZ          ;Retour si separateur
0402 04          INR B
0403 05          BL4      DCR B
0404 C8          RZ          ;Retour si fin de ligne
0405 C3EE03     JMP          BL5
0408           ;
0408           ; Decremente le compteur de DATA
0408           ;
0408 2A2504     RESTDA LHL      CPTDAT
0408 2B          DCX H
040C 222504     SHLD      CPTDAT
040F C9          RET
0410           ;
0410           ; Debut commun
0410           ;
0410 EB          DEBCOM XCHG
0411 E3          XTHL
0412 EB          XCHG
0413 C5          PUSH B
0414 23          INX H      ;)Contenu de la variable
0415 23          INX H      ;)BASIC
0416 7E          MOV A,M      ;)
0417 322604     STA      CPTDAT+1H
041A 23          INX H
041B 7E          MOV A,M
041C 322504     STA      CPTDAT
041F EB          XCHG
0420 E9          PCHL
0421           ;
0421           ; Erreur
0421           ;
0421 79          ERR      MOV A,C
0422 C3F5D9     JMP          OD9F5H
0425           ;
0425 0000       CPTDAT DW      OH      ;Compteur de DATA
0427 0000       NBTIL  DW      OH      ;DATA utilises
0429 FIN       END

```

**D A T A S A N S D I E S E**  
**Claudius Morin**

Non je ne vous propose pas de faire de la musique, mais seulement de supprimer les # devant les nombres hexadécimaux. Ceux-ci sont fréquemment utilisés pour charger un PLM à partir de DATA. Le nombre de caractères est ainsi diminué.

Certains BASIC acceptent l'écriture suivante :  
 B\$="FF" : B\$="#" + B\$ : A%=VAL(B\$)  
 A% contient le nombre 255

La routine, DATA SANS DIESE, simule cette possibilité.  
 Elle ne fonctionne que pour des nombres au plus égaux à 255. Elle peut être adaptée pour transformer des nombres plus grands.

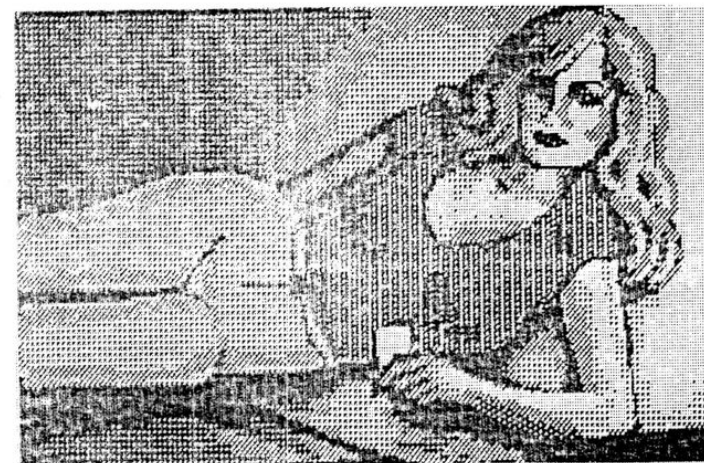
L'assemblage de la routine se fait dans une variable dimensionnée, A%(23), déclarée au début du programme BASIC, comme cela l'origine de la routine se trouve à l'adresse de la dimension 0 de la variable, utilisée pour récupérer le nombre.

Si vous n'avez pas réservé de mémoire, la première dimension est en #2F0, vous pouvez utiliser la routine comme suit :

```

DATA FF, ...
READ B$
CALLM#2F4,B$
PRINT A%(0), retourne 255.

```



SPL V1.1 PAGE 1

```

0000 ;
0000 ; TITL 'DATA sans DIESE'
0000 ; Claudius Morin
0000 ;
0000 ; ORG 2FOH
02F0 ;
02F0 0000 ;ZERO DW OH ;)Dimension zero de la variable
02F2 0000 DW OH ;)contenant la routine
02F4 ;
02F4 CS ; DEB PUSH B
02F5 SE MOV E,M
02F6 23 INX H
02F7 56 MOV D,M ;)HL=adr du nbre de
02F8 EB XCHG ;)caracteres transmis
02F9 0600 MVI B OH
02FB 1E0A MVI E OAH ;Code "INVALID NUMBER"
02FD 7E MOV A,M ;)Seulement
02FE B7 ORA A ;)1 ou 2
02FF CA4603 JZ ERR ;)caracteres
0302 FE03 CPI 3H ;)a traduire
0304 D24603 JNC ERR
0307 E5 PUSH H
0308 4F MOV C,A
0309 3630 MVI M 'O'
030B 09 DAD B
030C 7E MOV A,M
030D CD2203 CALL TRANSF
0310 47 MOV B,A
0311 2B DCX H
0312 7E MOV A,M
0313 CD2203 CALL TRANSF
0316 07 RLC
0317 07 RLC
0318 07 RLC
0319 07 RLC
031A 80 ADD B
031B 32F302 STA ZERO+3H ;memorisation du nombre
031E E1 POP H
031F 71 MOV M,C
0320 C1 POP B
0321 C9 RET
0322 ;
0322 ; Transformation d'une chaine "O" a "F"
0322 ; en nombre de 0 a 15
0322 ;
0322 1E09 TRANSF MVI E 9H ;Code "NUMBER OUT OF RANGE"
0324 FE30 CPI 30H ;'O'
0326 DA4603 JC ERR
0329 FE3A CPI 3AH ;'9'+1H
032B DA3803 JC BC1
032E FE41 CPI 41H ;'A'
0330 DA4603 JC ERR
0333 FE47 CPI 47H ;'F'+1H
0335 D24603 JNC ERR
0338 S7 BC1 MOV D,A
0339 E60F ANI OFH ;Elimine le poids fort
033B 5F MOV E,A ;0 a 9 ou 1 a 6
033C 7A MOV A,D
033D 07 RLC ;)eb7 de parite=1
033E B7 ORA A ;)si A>nbre<F
033F 7B MOV A,E
0340 F24503 JP BC2
0343 C609 ADI 9H
0345 C9 BC2 RET
0346 ;
0346 ; ecriture des codes d'erreur
0346 ;
0346 ;
0346 E1 ERR POP H
0347 E1 POP H
0348 71 MOV M,C
0349 7B MOV A,E
034A C3F5D9 JMP OD9F5H
034D ;
034D ; FIN END

```

BRIS thierry 46 rue d'AVRON 94170 Le PERREUX

CLAVIER POUR NOTRE CHER DAI

Le but du jeu consiste à raccorder un clavier à notre ordinateur préféré. Encore un ! vous allez penser, oui mais, (car bien sur il y a un mais...) cet article décrit une réalisation sans électronique sans alimentation et sans mise au point. Un fer à souder, du fil, et un clavier mécanique suffisent.

D'abord le clavier:

- N'importe lequel convient à condition qu'il soit mécanique et bien sûr sans rebonds!!!!, à éviter les claviers à effet Hall car ils nécessitent une alimentation.

- Le plus gros travail consiste à retrouver la matrice du clavier ou alors, ce qui était mon cas, en refaire une sur de la plaquette à trous, ou pour les plus courageux sur un circuit imprimé.

- La matrice est constituée de lignes en X et colonnes en Y, vous la trouverez dans le manuel du DAI à la page n.132.

- Donc reliez un point de toutes les touches de la lignes 0 entre elles, puis celles de la ligne 1, et ainsi de suite. Chaque ligne doit être indépendante les unes des autres.

- Refaire de même avec les colonnes sur l'autre point de libre de chaque touche, courage il n'y a que 15 fils à souder.

- A la fin de ce travail vous aurez donc 15 fils à sortir constitués de 7 fils pour les colonnes et 8 fils pour les lignes. Repérez les et soudez les sur un connecteur ayant au moins 18 broches. Un connecteur 25 points mâle Cannon suffit largement, ne pas oublier les verrouillages du connecteur!



Ensuite le DAI:

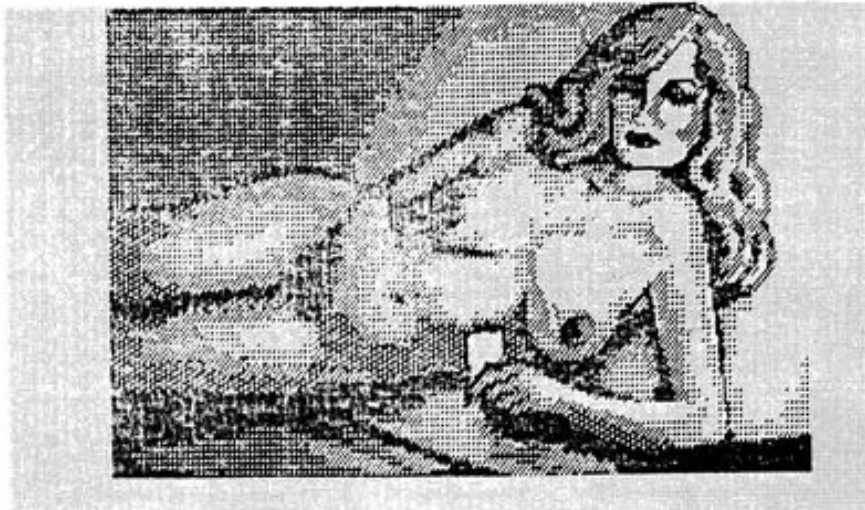
- Percez une ouverture sur un des côtés pour fixer le connecteur femelle, avec verrouillage bien sûr!

- De ce connecteur soudez vos 15 fils toujours en les repérant d'après le clavier, les extrémités de ces fils seront à souder sur les points correspondant du clavier d'origine du DAI. (toujours d'après la matrice de la page n.132)

- Finalement les 2 claviers seront branchés en parallèle, une petite précaution: passez un fil de masse dans votre cordon de liaison, mais du fil blindé n'est pas utile. Vous pouvez apporter aussi une autre amélioration: sortir deux fils du bouton reset et installer un poussoir reset sur le clavier déporté, choisir judicieusement sa place et son type sinon attention aux resets malencontreux.!!!!!!

J'ai testé cette réalisation avec 2 mètres de fils sans aucun problème quand au boîtier je laisse libre cours à votre imagination

Bon courage !



### Des timers

Pour pouvoir exécuter un sous programme périodiquement, il y a la possibilité d'utiliser l'interruption 7 qui se déclenche toutes les 20 ms (comme on l'a vu sur l'article Dai & minitel à propos du clignotement caractères/courseur). Mais las, 20 ms, c'est beaucoup, c'est parfois trop. Heureusement il y a une autre solution: le timer.

Le Dai en possède 5, qui sont gérés par notre bien aimé 5501 (oui, tu sais: c'est lui qui gère aussi les interruptions et la RS232). Ces 5 timers sont, il faut l'avouer assez rudimentaires par rapport à ceux que l'on trouve généralement sur le marché: on les charge avec une valeur qu'ils décrémentent toutes les 64 microsecondes, et quand ils ont atteint 0, ils s'arrêtent de compter et provoquent une interruption.

A chaque timer correspondent une adresse de chargement et une interruption associée (et dont un vecteur d'interruption), voici les correspondances:

No de timer	ad. de charg.	inter. associée	ad. du v. d'inter.	fonction usuel- -le de l'inter.
1	#FFF9	0	#62 / #63	L de l'utilty.
2	#FFFA	1	#64 / #65	UT & encodage.
3	#FFFB	3	#68 / #69	Pour les sons.
4	#FFFC	6	#6E / #6F	Clavier.
5	#FFFD	7	#70 / #71	Horloge (TV)*

\*: Normalement l'interruption 7 fonctionne en tant qu'interruption externe (elle est reliée au circuit vidéo qui délivre une impulsion toute les 20 ms). Pour transformer l'int. 7 en interruption liée au timer 5, il suffit d'envoyer au registre de commande du 5501 (adresse #FFF4) l'octet #80, pour revenir à l'état normal, il faut envoyer #C0.

Comme on le voit, il faudra faire attention

1) de bien autoriser l'interruption liée au timer choisi en mettant à 1 le bit correspondant à cette interruption dans le registre de masquage (adresse: #FFF8).

2) de ne pas perturber le système d'exploitation qui se sert déjà de nombre vecteurs d'interruption pour ses besoins personnels.

Enfin pour éclaircir les idées, un petit exemple, qui modifie à intervalle de temps régulier un caractère sur l'écran:

Sébastien Dubourg,  
le 22 janvier 1987

TITL 'TIMMER by Sebastien DUBOURG 21 Janvier

```

0000 ;
0000 ;(c) IDC Bordeaux
0000 ;
0000 ;
0000 ;#####
0000 ;# Point d'entree #
0000 ;#####
0000 ;
0000          ORG      400H
0400 C30304    JMP      INTON
0403 ;
0403 @=00A0    TIMING  EQU    0A0H          ;Temporisation en unites
0403 ;                          ;de 64 microsecondes.
0403 ;
0403 @=8F81    DIODE   EQU    0BF81H      ;Une case de l'ecran.
0403 ;
0403 @=005F    RMASK   EQU    5FH         ;Copie du registre de
0403 ;                          ;masquage en ram.
0403 @=0062    VO      EQU    62H         ;Vecteur d'int. 0
0403 ;                          ;(timmer0).
0403 @=FFF8    MASK    EQU    0FFF8H     ;Registre de masquage.
0403 @=FFF9    TIMMER  EQU    0FFF9H     ;Adresse du timmer 0.
0403 ;
0403 ;#####
0403 ;# Mise en fonction du timmer #
0403 ;#####
0403 ;
0403 F3        INTON   DI
0404 3ECS      MVI A   OCSH          ;Autorise l'int. 0.
0406 32F8FF   STA    MASK
0409 325F00   STA    RMASK
040C 211904   LXI H   INTIME        ;Met en place notre
040F 226200   SHLD   VO              ;propre vecteur d'int.
0412 3EAO     MVI A   TIMING        ;Charge le timmer
0414 32F9FF   STA    TIMMER        ;(initialisation).
0417 FB      EI
0418 C9      RET
0419 ;

```

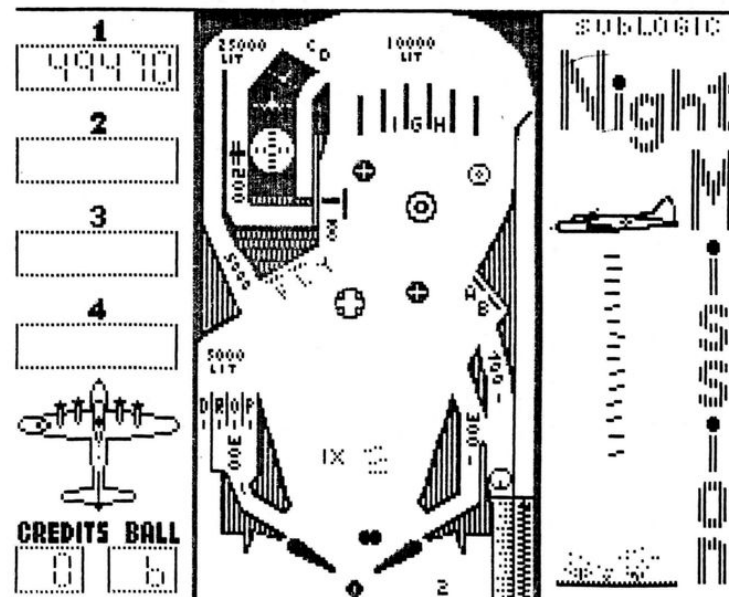
```

0419 ;
0419 ;#####
0419 ;# Routine appelee par le timmer #
0419 ;#####
0419 ;
0419 F5        INTIME  PUSH PSW
041A 2181BF   LXI H   DIODE          ;Modifie le caractere
041D 34        INR M                    ;d'adresse DIODE.
041E 3EAO     MVI A   TIMING        ;Recharge le timmer.
0420 32F9FF   STA    TIMMER
0423 F1        POP PSW
0424 E1        POP H
0425 FB      EI
0426 ;
0426 C9      RET                      ;Si on n'autorise pas
0427 ;                          ;a nouveau les int.,
0427 ;                          ;ce bloque le systeme.
0427 ;
0427          END                      ;That's all folks !!!

```

SPL V1.1 PAGE 1

BAUDS :0000 DIODE :8F81 INPUT :0000 INTIME:0419 INTON :0403 MASK :FFF8  
 OUTPUT:0000 RMASK :005F STATUT:0000 TEST1 :0000 TIMING:00A0 TIMMER:FFF9  
 VO :0062







### Einfacher 8 Bit - 8 Kanal A/D Wandler

Autor: Hardy Strobel

Technische Daten: Bereich 0 - +5V regelbar  
Impedanz ca. 4,5 KΩ  
Umsetzzeit < 100µs

Kurzbeschreibung: 74HC138 - Kartenadresse  
74LS125 - Clockgenerator, Buffer,  
RD-Generierung  
ADC0808/0809 - A/D-8 Bit / 8 Kanal

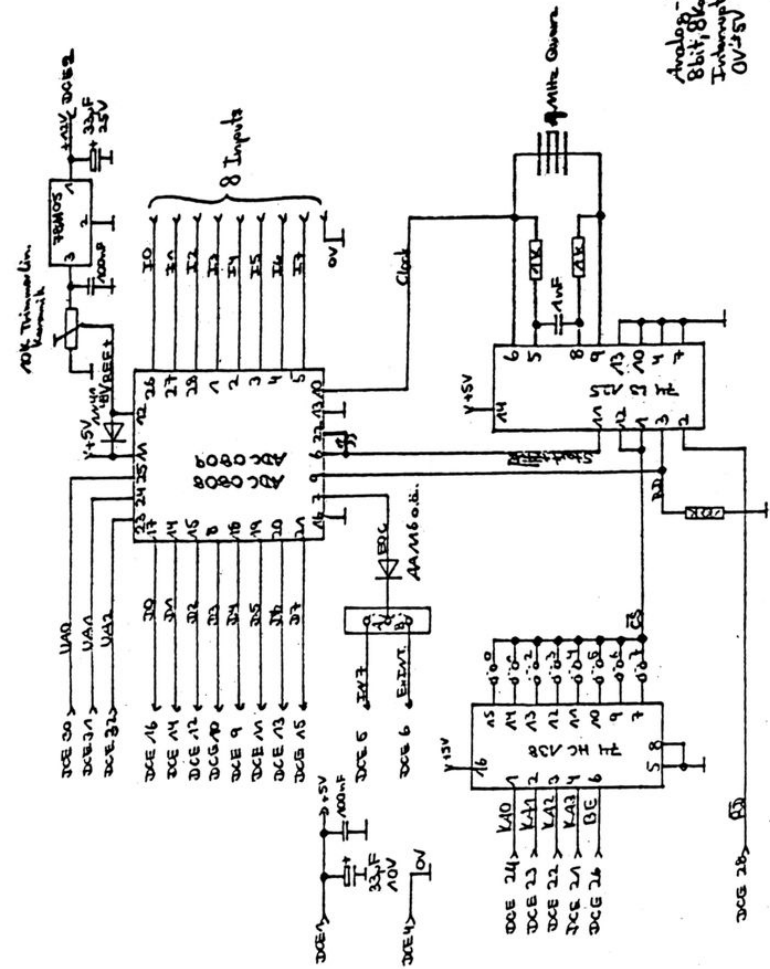
Bedienung: Das Interface kann mit folgendem BASIC Befehl arbeiten:

AZ=INP(#n)      n = Kartenadresse 0-7  
                  m = Kanal 0-7

Daten:            AZ kann Daten von #00-#FF enthalten.

Interruptsteckbrücke:    EDC wird "1", wenn der Umsetzvorgang beendet ist

Die Einstellung des zu wandelnden Spannungsbereichs wird mit  $V_{ref}$  vorgenommen. Bitte fragen Sie bei Interesse beim Club nach den Datenblättern.



Analog-Digitalwandler,  
8 Bit, 8 Kanal, < 100µs  
Trennung, regelbarer Bereich  
0V-5V  
HM 4/8.82  
© Bastard  
D. Strobel

### EPROM PROGRAMMER

Mit dem nachfolgend vorgestellten EPROM-Programmer ist es auf einfache und preiswerte Weise moeslich EPROMs folgender Typen zu programmieren : 2716,2732,2732a,2764 und 27128.

### DIE SOFTWARE

Mit der Software fuer den Programmer sind folgende Operationen moeslich:

#### 1. Leertest:

Bei diesem Test stellt der Rechner fest, ob das EPROM gelöscht ist, d.h. ob alle Speicherzellen den Wert #FF enthalten. Stoesset er bei diesem Test auf eine ungeloeschte Speicherzelle, so gibt er deren Adresse aus und fuehrt den Test fort.

#### 2. Lesen:

Mit dieser Operation ist es moeslich den Inhalt des EPROMs in den Speicher des Rechners einzulesen. Hierzu verlangt der Rechner die Einsabe einer Startadresse, ab der der Inhalt des EPROMs abgespeichert wird.

#### 3. Programmieren:

Wird diese Funktion angewaehlt, so verlangt der Rechner drei Einsaben: a. Anfangsadresse des zu programmierenden Speicherbereichs; b. Adresse, ab der das EPROM programmiert werden soll. c. Anzahl der zu programmierenden Bytes. Beispiel: Man moechte in ein EPROM vom Typ 2764 folgende Daten programmieren: 3248 Bytes-ab Adresse #B350 sollen im EPROM ab Adresse #400 abgespeichert werden. Dazu gibt man zunaechst die Startadresse der zu programmierenden Daten ein (B350), dann die Adresse, ab der das EPROM programmiert werden soll (400) und schliesslich die Anzahl der zu programmierenden Bytes (CB0). Der Rechner teilt dann mit, dass das EPROM programmiert wird, und wann er fertig ist.

Vorher wird ein Loeschtest durchgefuehrt und evtl. darauf hingewiesen, dass das EPROM nicht vollstaendig gelöscht ist. Nach dem Programmieren wird ueberprueft ob der Programmiervorgang erfolgreich war.

#### 4. Verifizieren:

Mit dieser Funktion ist es moeslich den Inhalt des EPROMs mit einem Speicherbereich des Rechners zu vergleichen. Die Einsabe der Parameter ist gleich der beim Programmieren. Tritt eine Unstimmigkeit zwischen Rechnerspeicher und EPROM auf, so wird diese Adresse ausgegeben. Der Vergleichstest kann durch den Druck einer Taste abgebrochen werden.

Die Einsaben der Parameter erfolgt im UTILITY-Format und muessen durch 'RETURN' abgeschlossen werden. Bei jeder Fehlermeldung (z.B. es wurde keine Hex-Zahl eingegeben) wird die Operation abgebrochen, eine Fehlermeldung ausgegeben und zurueck zum Menu ueberfuehrt.

### EPROM PROGRAMMER

### DIE HARDWARE

Der EPROM-Programmer besteht aus einer Platine, die mit einem RWC-Stecker ausgeruestet ist, (wahlweise kann auch ein 34-poliges Flachbandkabel mit DCE-Bus Stecker angeschlossen werden). Ein gleichzeitiger Betrieb des EPROM- Programmers und anderer RWC-Karten ist nicht moeslich.

Die Hardware selbst ist sehr einfach gehalten und daher auch preiswert. Die Adressierung des EPROMs erfolgt ueber Zaehlbau- steine, der Datenbus ist direkt verbunden, und das Anlegen der Programmierspannung erfolgt ueber Transistoren, die in einer Sicherheitsschaltung so verknuepft sind, dass eine groesst- moegliche Sicherheit gegen Beschaedigungen des EPROMs erreicht wurde (selbst bei Systemabstuerzen passiert meistens nichts). Die Anpassung der Programmierspannung an die verschiedenen EPROMs ist ebenfalls ueber Transistoren geregelt und voll unter Software-Kontrolle.

Auf der Platine ist alles ausser dem Netztrafo untergebracht. Es muss also nur noch eine Wechselspannung von ca. 25 V angeschlossen werden.

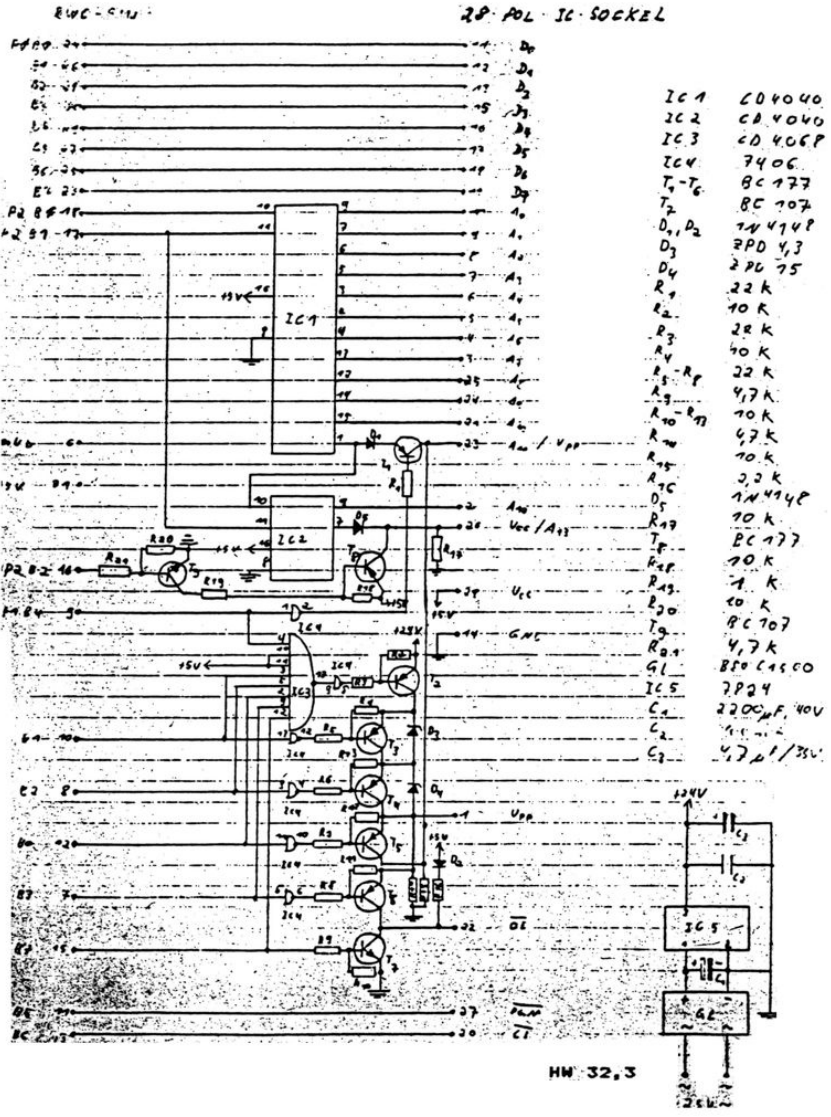
Fuer Interessenten, die die Schaltung nicht in Wire-Wrap- Technik erstellen wollen, besteht die Moeglichkeit eine gebohrte Platine (oder auch bestueckt und getestet) bei mir zu bestellen. Die Preise belaufen sich auf ca. 25,-DM fuer eine unbestueckte und ca. 100,-DM fuer eine bestueckte Platine.

Die Software ist zum Selbstkostenpreis beim Club erhaeltlich, bzw. wird beim Bestellen einer Platine mitgeliefert.

Norbert Rinnen

Norbert Rinnen  
Moerikestr. 10  
5060 Bergisch-Gladbach 1  
Tel. 02204/81963

EPROC PROGRAMMEUR 2746 / 32 (A) / 64 / 128 ->



PASCAL & ALGORITHME :  
de nouveaux horizons

Les articles qui vont être publiés tout au long de cette année suivront le plan suivant :

- I Introduction et historique
- II L'écriture algorithmique
- III Mode d'emploi du DAI Tiny Pascal
- IV Ordres du Pascal. Parallèle algorithme-Pascal
- V Bibliographie

Aujourd'hui donc :

II L'écriture algorithmique



## II L'écriture algorithmique

### A. Les différentes types de programmation

La programmation, art d'utiliser la puissance des ordinateurs pour déguiser en intelligence leur extrême stupidité se décompose en trois phases :

- \* Aller du problème vers un modèle
- \* Construire un algorithme à partir du modèle
- \* Transcrire l'algorithme en programme

Comme l'algorithme est indépendant de la machine et du langage utilisé, il sera rédigé en français clair et explicite afin d'être aisément compris par un tiers (et de pouvoir être relu quelques mois plus tard lors d'une modification...). Cependant il suit une méthode de programmation particulière qui peut être :

\* La PROGRAMMATION SYSTEMATIQUE revient à construire le programme en suivant les 3 phases énoncé précédemment. C'est la base de toute programmation qui se veut sérieuse et non improvisation (voir soirée d'hiver en basic...).

\* La PROGRAMMATION MODULAIRE. Dans de nombreux cas, la complexité du problème impose de le décomposer en sous-problèmes et d'aboutir à des algorithmes pour ces sous-problèmes. Ainsi, la démarche de la programmation modulaire met l'accent sur l'importance du découpage d'une application en unités logiques (non faites par DCAhard !!!) définissant les liens entre ces diverses unités.

\* La PROGRAMMATION STRUCTUREE. La construction d'un algorithme s'appuie sur un certain nombre de structures primitives et qui fournissent une méthode de pensée claire et simple qu'il convient de conserver tout au long des développements successifs de l'algorithme: c'est la démarche de la programmation structurée. L'intérêt de ces structures de base, outre leur simplicité d'utilisation, est de permettre à partir du texte du programme, la démonstration de certaines des propriétés de son exécution. C'est là un pas considérable vers la solution d'un problème fondamental :

PEUT-ON FAIRE LA PREUVE DU PROGRAMME AVANT SON  
EXECUTION. C'EST A DIRE DEMONTRER QU'IL VA  
FONCTIONNER CORRECTEMENT ?

\* La PROGRAMMATION DEFENSIVE. La sécurité est une exigence importante et naturelle de l'utilisateur d'un programme. Sa définition est évidente dans certains domaines d'application comme la commande des processus, les systèmes d'exploitations. La notion de sécurité existe cependant pour tous les types de programmes; on peut la définir comme l'aptitude du programme à fournir des résultats raisonnables dans

tous les environnements possibles, et en particulier dans des conditions anormales. L'exemple le plus simple de condition anormale est celui où l'on fournit à un programme des données erronées qui doivent être décelées et rejetées et non pas conduire à une catastrophe. Pour prévenir de telles situations il est naturel de pratiquer une programmation défensive qui consiste à prévoir le programme pour un environnement réel ou tout peut se produire, plutôt que pour des circonstances idéales (comparez à ce sujet DAITEXT v21 et DAINOR-DSTAR ...). Il s'agit en quelque sorte d'élever au rang de principe de programmation la loi, dite de Murphy (non prénommé Eddy...), selon laquelle "si le pire peut arriver, il se produira".

A l'heure actuelle, il n'existe pas de méthode de programmation miracle mais sachez que l'écriture algorithmique que nous allons voir à présent s'appuie sur les 4 méthodes précédentes et donc, si elle est respectée par le programmeur, conduit à des programmes fort correct dans tous les domaines.

### B. Rappels et définitions

OBJET : ce terme désigne toutes les données manipulées par le programme (ex: variables alphanumériques, constantes, tableaux).

IDENTIFICATEUR : c'est le nom d'un objet. Il n'est que trop conseillé de choisir des identificateurs clairs (ex: nbrepas est plus facile à relire que n...).

AFFECTATION : on donne une valeur à un objet.

SEQUENCE D'INSTRUCTIONS : c'est une suite ordonnée d'instructions (la fin d'une instruction déclenche l'exécution de la suivante).

BLOC : ensemble d'instructions enchaînées séquentiellement, dont l'exécution ne peut commencer que par la première instruction du bloc et ne peut se terminer que par la dernière; un bloc apparaît comme un ensemble inséparable et donc comme une instruction de plus haut niveau sémantique.

```

ex : instruction 1
      instruction 2
      instruction 3
      si condition alors | instruction 4
                        | instruction 5
                        | instruction 6
                        | instruction 7
                        | instruction 8
      instruction 9
      instruction 10

```



les traits verticaux signifient qu'il y a enchainement séquentiel de haut en bas.

COMMENTAIRES : utile aussi pour le concepteur, on les trouvera entre accolades.

ALGORITHME : ensemble structuré de blocs d'instructions permettant de décrire complètement une méthode de résolution d'un problème donné.

C. Un principe de programmation

Un bon programme est fainnant car il utilise beaucoup d'astuces lui permettant de résoudre le problème le plus rapidement possible en un minimum d'algorithmes. Ainsi une bonne analyse mathématique évite bien du travail. En effet pour calculer la somme des N premiers nombres entiers positifs, nul besoin d'une boucle, il suffit de savoir que  $S=N(N+1)/2$  ... On aboutit alors à l'algorithme suivant :

```

Algorithme somme N nbres entiers : titre algorithme
  entrer (N)      : algo. prédéfini d'entrée
  somme ← N(N+1)/2 : affectation
  sortir (somme)  : algo. prédéfini de sortie

```

Les tabulations utilisées en algorithmique sont destinées à faire apparaitre clairement les blocs et les sous-blocs qu'ils contiennent. Ceci est particulièrement vrai lors d'alternatives multiples (voir plus loin), illisibles sans tabulations. Les règles de tabulations ne sont pas strictes, il s'agit simplement de rendre clair l'algorithme à la lecture. Aussi, point de nombre de blanc imposé mais je vous invite pour chaque fonction a regarder l'illustration qui est faite tout au long de ces articles.

D. Les types non standard

En algorithmique (comme en Pascal) tous les objets manipulés doivent être d'un type bien défini c'est à dire fixé par l'utilisateur une fois pour toute. Cela peut paraître contraignant de prime abord mais nous allons voir tout de suite que cela présente un énorme avantage sur des langages "sous-évolués" comme les premiers basics.

Parmis les types standard on a les entiers, les réels, les caractères. Pour ceux-ci, nul n'est besoin de spécifier qu'une variable sera d'un de ces types car cela sera sous-entendu. Par contre on précisera le type booléen. Une variable de ce type prendra la valeur vrai ou la valeur faux. On voit tout de suite l'intérêt de ce type lors de test.

Mais là où se situe l'innovation, c'est que l'utilisateur peut créer ses propres types. Voyons divers exemple.

Vous voulez utiliser un variable E entière mais qui est toujours comprise dans l'intervalle 2 à 10. Nul n'est besoin de définir la variable comme de type entière, il suffit de créer le type INTERVALLE, intervalle de 2 à 10, qui sera le type de la variable. On peut procéder de la même manière avec des variables réelles ou alphanumériques (caractères). De plus le compilateur vérifiera lors de l'exécution que la variable reste bien du type défini, c'est à dire reste dans l'intervalle. Eh oui, vous avez ainsi économisé 1 (ou plusieurs) test en cours de programme.

Autre cas de figure, vous utilisez 5 variables qui sont des tableaux de n fois n nombres entiers. Au lieu de déclarer les variables une par une, il suffit de définir le type MATRICE qui est un tableau de n fois n nombres entiers et de déclarer les variables comme du type MATRICE.

Vous désirez aussi utiliser une chaîne de 20 caractères. Il suffit de définir une type CHAINE qui est en fait un tableau d'une ligne à 20 colonnes.

Enfin vous traitez des mois de l'année. Pourquoi ne pas créez un type MOIS ?

Tous ces définitions se présentent alors sous la forme :

```

Algorithme trucmuche
  n<-100 : il faut fixer n quelque part
  type : définition des types
  INTERVALLE = 2..10
  MATRICE = tableau(1..n,1..n) d'entiers
  CHAINE = tableau(1..20) de caractères
  MOIS = Janvier,Février,etc...,Décembre
  var : déclaration des objets
  TEST : BOOLEEN
  E : INTERVALLE
  A : MATRICE
  B : MATRICE
  C : MATRICE
  D : MATRICE
  F : MATRICE
  NOM : CHAINE
  DATE : MOIS

```

Remarque : diverses fonctions sont possibles sur les intervalles. ex : dim(MOIS)=12; dans INTERVALLE  $2 < 4$  et  $6 > 3$  mais aussi précesseur(5)=4 et successeur(7)=8; de même dans MOIS Janvier<Février et Novembre>Octobre mais aussi précesseur(Mars)=Février et successeur(Juillet)=Aout. Essayez donc de faire les mêmes fonctions en basic...

E. La structure alternative

On distingue deux types d'alternatives :

\* L'alternative simple \*

SI (condition) ALORS (bloc d'instructions 1)  
SINON (bloc d'instructions 2)

La condition est une expression logique.

Dans le cas d'une variable booléenne :

SI test ALORS bloc <=> SI test=vrai ALORS bloc  
SINON bloc SINON bloc  
SI non(test) ALORS bloc <=> SI test=faux ALORS bloc  
SINON bloc SINON bloc

La directive SINON est obligatoire; dans le cas où il n'y a rien faire si la condition n'est pas vérifiée on écrira :

SI (condition) ALORS (bloc d'instructions 1)  
SINON rien

où rien est un algorithme prédéfini qui ne fait rien !!!

Algorithme conversion décimal-binaire sur 2 bits

```
entrer(d)
si (d=0) alors b <- 0
sinon
si (d=1) alors b <- 1
sinon
si (d=2) alors b <- 10
sinon
si (d=3) alors b <- 11
sinon b <- -1
si (b=-1) alors sortir('dépassement capacité') : '<=>'basic
sinon sortir(b)
```

\* L'alternative multiple \*

AU CAS OU

condition 1 : (bloc d'instructions 1)  
condition 2 : (bloc d'instructions 2)  
.....  
condition n : (bloc d'instructions n)  
SINON : (bloc d'instructions n+1)

Les conditions 1 à n sont des expressions logiques; dans le cas où aucune des n conditions n'est vérifiée (SINON), on effectue le bloc n+1.

Dans le cas d'une variable booléenne :

AU CAS OU test : bloc <=> AU CAS OU test=vrai : bloc  
AU CAS OU non(test) : bloc <=> AU CAS OU test=faux : bloc

La directive SINON est obligatoire.

La structure AU CAS OU est équivalente à l'imbrication de plusieurs structures SI ALORS SINON :

```
SI condition 1
ALORS (bloc d'instructions 1)
SINON SI condition 2
ALORS (bloc d'instructions 2)
SINON SI condition 3
ALORS
...
ALORS SI condition n
ALORS (bloc d'instructions n)
SINON (bloc d'instructions n+1)
```

Algorithme conversion décimal-binaire sur 2 bits

```
entrer(d)
au cas où :
d=0 : b <- 0
d=1 : b <- 1
d=2 : b <- 10
d=3 : b <- 11
sinon : b <- -1
au cas où :
b=-1 : sortir('dépassement capacité')
sinon : sortir(b)
```

F. Les structures itératives

3 types d'itérations sont possibles :

\* La répétition avec compteur \*

Similaire à celle du basic elle se présente sous la forme :

incréméntation : POUR compteur DANS intervalle  
FAIRE (bloc d'instructions)  
décréméntation : POUR compteur DANS intervalle INVERSE  
FAIRE (bloc d'instructions)

rappel : \*intervalle=(min..max); les valeurs de min et max ne doivent pas être modifiées par l'exécution du bloc d'instructions.

\*compteur est l'identificateur d'une variable de type ordinal T; l'exécution du bloc d'instructions ne doit pas modifier la valeur de compteur.

\*le bloc d'instructions est exécuté n=(max-min+1)fois.

Bien que possible, cette répétition avec compteur sera rarement utilisé car elle ne répond pas aux critères de la programmation défensive. Les concepteurs de PASCAL se sont

d'ailleurs demandé s'ils devaient ou non l'inclure dans leur langage.

\* La répétition avec controle à la fin \*

REPETER (bloc d'instructions) JUSQU'A (condition)

La condition est une expression logique.

Le bloc d'instructions est répété tant que la condition n'est pas vérifiée; il est toujours exécuté au moins une fois.

La condition peut être toujours vérifiée; dans ce cas, le bloc est exécuté une seule fois.

Attention !!! La condition peut ne jamais être vérifiée; dans ce cas l'exécution du bloc se poursuit indéfiniment !!!...

Algorithme factorielle de n

```

| entrer('n=?',n)
| fact<-1
| REPETER fact<-fact*n
|     n<-n-1
| JUSQU'A (n<=1)
| sortir('n!','=',fact)

```

\* La répétition avec controle à la fin \*

TANT QUE (condition) EXECUTER (bloc d'instruction)

La condition est une expression logique.

Le bloc d'instructions est répété tant que la condition reste vérifiée.

La condition peut ne jamais être vérifiée; dans ce cas le bloc ne sera jamais exécuté.

Attention !!! La condition peut être toujours vérifiée; dans ce cas, l'exécution du bloc se poursuit indéfiniment !!!...

Remarque : la structure REPETER JUSQU'A est équivalente à :

```

| bloc d'instructions 1
| TANT QUE non(condition) EXECUTER (bloc d'instructions 1)

```

Algorithme racine carrée entière

```

| entrer(a)
| r<-0
| y<-1
| z<-1
| TANT QUE (y<=a) EXECUTER z<-z+2
|     y<-y+z
|     r<-r+1

```

```

| racine<-r
| sortir(racine)

```

Ici se termine cette deuxième partie des articles sur l'écriture algorithmique. La prochaine fois nous verrons les variables globales et locales, les procédures et fonctions, la récursivité. Mais afin que vous ne ennuyez pas je vous propose la recherche des algorithmes répondant aux cahiers des charges suivants :

- 1°) Recherche du PGCD de 2 nombres entiers  
Utiliser l'algorithme d'Euclide (TANT QUE..EXECUTER).
- 2°) Recherche des racines d'une équation du second degré  
Utiliser les structures SI ALORS SINON et AU CAS OU SINON.
- 3°) Recherche dichotomique  
On dispose d'un tableau de nombres entiers rangés dans l'ordre croissant. On veut savoir si un nombre donné se trouve dans le tableau : s'il ne s'y trouve pas on l'inclut dans le tableau et on détermine son rang : s'il s'y trouve, on détermine son rang.
- 4°) Racine carrée  
Calculer la racine carrée entière d'un nombre positif en utilisant une structure TANT QUE..EXECUTER.
- 5°) Réservation SNCF (ou autre pour nos amis européens)  
On dispose de 60 places assises dont 20 fumeurs. Ecrire l'algorithme permettant de réserver une place, annuler une réservation, arrêter les réservations par la méthode de votre choix.

BON COURAGE !!!

Philippe Casier, membre de DCA  
31ter rue de Paris  
92190 Meudon



INVMAT

Le sous programme très court (moins de 500 octets) nommé INVMAT a l'avantage d'inverser toute matrice carrée dont la dimension n'excède pas 62. La technique utilisée est très simple et est connue sous le nom de "méthode par recherche de pivot". Pour l'utiliser, il faut charger le sous-programme INVMAT (ligne 10000 et plus), dimensionner un tableau A(N,2\*N) où N est la dimension, placer la matrice dans la partie A(0&N,0&N) et faire GOSUB 10000. La matrice inversée se trouve dans la partie A(0&N,N+1à2\*N).

Tout ceci est très simple à utiliser. Parlons maintenant des performances. Vous pouvez juger de la rapidité de INVMAT en calculant le temps d'inversion de votre matrice par la formule :

$$t(s) = N^2 * (0.14 + 0.0285(N-2))$$

ce qui donne une demi seconde pour une matrice 2\*2, 5s pour une matrice 5\*5, 36s pour une matrice 10\*10, 263s (4'23) pour une matrice 20\*20 et 25' pour la matrice maxi 62\*62 (prévoir

CLEAR 32500).

Si vous avez doté votre DAI d'une extension RAM de 4Mo vous pouvez alors inverser la reine des matrices, la matrice 1000\*1000 en ... 11 mois ...

H. Samain, membre de DCA  
13 rue du Loup pendu

91570 Bievres

```

9998 REM INVERSION DE MATRICE (DIM MAX=62) PAR H.SAMAIN POUR D.C.A.
9999 REM DIMENSIONNEZ A(N,2*N) OU N EST LA DIM DE LA MATRICE A INVERSER
10000 FOR I=-N!+1.0 TO 2.0*N!:A!(I!-N!,I!)=1.0:NEXT I!
10010 FOR J!=1.0 TO N!:PIVOT!=0.0:FOR I!=J! TO N!
10020 IF ABS(A!(I!,J!))>PIVOT! THEN 10040
10030 GOTO 10050
10040 X!=I!:PIVOT!=ABS(A!(I!,J!))
10050 NEXT I!
10060 IF PIVOT!=0.0 THEN PRINT "MATRICE SINGULIERE":RETURN
10070 FOR K!=J! TO 2.0*N!:A!=A!(J!,K!):B!=A!(X!,K!):A!(J!,K!)=B!:
      A!(X!,K!)=A!: NEXT K!
10080 V!=A!(J!,J!)
10090 FOR K!=J! TO 2.0*N!:A!(J!,K!)=A!(J!,K!)/V!:NEXT K!
10100 FOR K!=1.0 TO N!:IF K!=J! THEN 10130
10110 Z!=A!(K!,J!)
10120 FOR L!=J! TO 2.0*N!:A!(K!,L!)=A!(K!,L!)-A!(J!,L!)*Z!:NEXT L!
10130 NEXT K!
10140 NEXT J!
10150 RETURN

```

```

logicielslogicielslogiciel
o                                     o
g                                     g
i                                     i
c   LOGICIELS                       c
i                                     i
e                                     e
logocielslogicielslogiciel

```

Résumé des logiciels proposés par certains membres

Les prix indiqués correspondent à la version K7 audio , à laquelle il faut rajouter +35 FF si vous désirez une version DCR ou Disquette.

Les frais d'envoi sont compris.

Envoyez vos demandes à :

\*\*\* Version K7 audio , Ken-Dos et DCR :

I.D.C.Bordeaux s/c DELANNAY Bruno  
Res. Les ACACIAS St. B3  
Avenue de SAIGE  
33600 PESSAC  
FRANCE

\*\*\* Version VC 1541 DOS allemand :

D.C.A. s/c D'ARANTES Luc  
Chemin Ste. Marie  
No 10.D Res. Ste. Marie  
30200 BAGNOLS sur CEZE  
FRANCE

N. B.

Les logiciels que proposait l'ex DAInamic sont toujours disponibles aux anciens tarifs , ils sont trop nombreux pour pouvoir figurer dans ces pages , se reporter à la liste de la LOGITHEQUE , dans laquelle ils sont inclus et se renseigner directement à I.D.C.Bordeaux.

J E U X :

QUIX : Jeu agréable, en Language Machine, de Pascal Janin. Vous connaissez le jeu d'arcade STYX ? : il s'agit de remplir le plus rapidement possible une zone de l'écran. Livré avec un livret expliquant l'emploi des routines des SPRITES et du Paint et des musiques. : 150 FF  
Disponible aussi sur DCR ,VC 1541 et Ken Dos.

**OTHELLO** : version classée 4me au concours de l'Ordinateur Individuel il y a deux ans. Très belle présentation , un des meilleurs. De Christian Poels : 150 FF  
Disponible aussi sur DCR ,VC 1541 et Ken Dos.

**SUPERCASEL** : aventure en graphique dans un immense chateau de trois étages , des centaines de pièces !! des tas de monstres. Belle présentation , un bon point d'humour , de R. SIP : 150 FF  
Disponible aussi sur DCR ,VC 1541 et Ken Dos.

**DAI-RUNNER** : Une belle imitation du Lode Runner sur Apple. Un très bon jeu , 50 niveaux de jeu !!! Par Mr. Stransky.

Attention non disponible en Audio. Prix incluant les 35 FF supplémentaires du DCR ou de la disquette : 100 FF

**DAITONA** : pilotez une voiture de course en toute sécurité, de Mr. Billot. : 100 FF  
Disponible aussi sur DCR ,VC 1541 et Ken Dos.

UTILITAIRES :

**EDITEUR DE DESSINS** : un très bon logiciel d'aide au dessin , tout pour dessiner et corriger , mixer deux couleurs d'un paint , utiliser les lettres ou une table de caractères que vous aurez redéfinis , bref tout pour plaire. Par Walter Costa. : 110 FF  
Disponible aussi sur DCR ,VC 1541 et Ken Dos.

**MAILING LIST** : un bon fichier pour tenir à jour , corriger , imprimer , trier , éditer vos étiquettes etc... Par Christian Poels : 150 FF  
Disponible aussi sur DCR ,VC 1541 et Ken Dos.

**DAILINK** : Allo ? pour communiquer avec tous les serveurs du monde ou entre deux DAI , à condition de posséder un modem bien sur ... , compatibilité des fichiers avec DAIWORDSTAR. Par Christian Poels : 150 FF  
Disponible aussi sur DCR ,VC 1541 et Ken Dos.

**DAITEXT V21** : bon traitement de texte adapté aux imprimantes au standard EPSON très original grace à ses possibilités télématiques intégrées permettant le transfert de ses fichiers textes par modem! Par Alain Mariette : 150 FF  
Disponible aussi sur DCR ,VC 1541 et Ken Dos.

**COM** : sans doute le meilleur programme de communication pour DAI , livré sur EPROM en version DCR ou Ken Dos Par Eric Choppinet :250 FF

**DAIBASE II** : un des meilleurs traitement de fiches sur DAI , qui reprend l'esprit des commandes du célèbre DBASE II.

Attention non disponible en Audio. Prix incluant les 35 FF supplémentaires du DCR ou de la disquette : 250 FF

**DISKWIFE** : véritable éditeur de directory , indispensable à tout possesseur de Ken-Dos , permet essentiellement de :  
1/ se balader dans les directory et modifier les paramètres.  
2/ faire automatiquement un fichier de toutes vos disquettes et retrouver très rapidement le titre recherché !  
3/ imprimer en deux format les données du directory.  
4/ copier la piste 0 à l'abri en piste 1 et/ou 2 afin de pouvoir récupérer le contenu de la disquette si cette piste 0 est détruite ! très utile Ken-Dos uniquement ! sur EPROM Par Xavier Drèze : 250 FF

Attention , se reporter un peu plus loin pour ce programme pour les précisions à fournir lors de sa commande !!

**MAXI-DIR** : pour Ken Dos uniquement , garde tous les paramètres des différents programmes de vos disquettes dans une disquette 'catalogue' qui vous permet essentiellement :  
1/ de lister et/ou imprimer tout ou partie de ce catalogue selon des critères que vous définissez , des catégories propres aux fichiers Ken-Dos ou définies par vous!!  
2/ de récupérer les fichiers perdus grace aux indications des pistes et secteurs gardés en mémoire ! Par Didier Boiteau : 150 FF

\* \* \*

Encore un nouveau programme ,réalisé par Mr. Boiteau , pour les heureux possesseurs de Ken-Dos qui veulent échanger des fichiers avec un I.B.M. ou compatible !!

\*\* \*\* \* P.C. DAI \*\* \*\* \*

Il vous permet :

-De transférer tous les fichiers DAI vers IBM.PC et tous les fichiers IBM.PC vers le DAI.

-Les fichiers ASCII IBM sont compatibles avec FWP sur le DAI un utilitaire permet la transformation des fichiers FWP en DTE (Daiwordstar) et un autre permet de transformer un fichier FWP en fichier ASCII assimilable pour un traitement de texte sur IBM...

-Un même disque DAI peut contenir des fichiers Ken-Dos et IBM.PC.

-La limite de transfert est de 256K octets ! Ce qui signifie que vous pouvez transférer sur disquette DAI tous les logiciels IBM les plus connus.

-Les fonctions Ken-Dos (copy , rename , backup...) sont utilisables avec des fichiers IBM.PC : vous pouvez par exemple , apres avoir recopié 'multiplan' sur disquette Ken-Dos taper 'COPY "OMULTIPLAN,2\*" ... et ça marche !!

Ceci m'amène à attirer votre attention sur l'utilisation de ce programme qui ne doit s'adresser qu'à des personnes désirant faire des copies de sauvegarde des ces programmes (valant plusieurs milliers de francs!) pour leur propre usage en respectant la loi sur les COPYRIGHTS ...

Toute autre utilisation pourrait donner lieu à des poursuites et nous ne pouvons que trop vous mettre en garde contre de telles pratiques.

Le prix de ce petit bijou est de 150FF tout compris.

\* \* \*

Des versions toutes nouvelles du programme de Mr. Uwe Wienkop en trois versions différentes.

Le meilleur traitement de fichier existant sur DAI , par l'auteur du non moins célèbre traitement de texte pour DAI (le meilleur et celui qui sert pour la revue !!) : DAIWORDSTAR , d'accès gratuit

\* \* \* SUPERBASE \* \* \*

250 FF tout compris

\* version DCR : en Français , ou Allemand

\* version Xbasic-ComDOS : en Allemand

\* version KEN-DOS : En Français , ou Allemand

N.B. Les versions en Allemand seront néanmoins fournies avec un double mode d'emploi : Français et Allemand.

Des articles Portant sur le codage des informations de Superbase suivront dans la revue.

Les fichiers du Club sont gardés sur Ken-Dos grace à ce programme !!

\* \*. \* DAIPAIN \* \* \*

Prix sur K7 audio = 150FF

Un des meilleurs utilitaires de dessin, réunissant des possibilités jamais encore vues sur le DAI de mixages des couleurs avec dégradé automatique !!, avec effet d'aerographe!! et bien entendu le tout accessible très facilement grace à son menu reprenant les principes des icones du Mac.! Réalisé par Bruno Cimbalnik.

Dernière minute !!! pour les personnes désirant le programme DISK-WIFE pour KEN-DOS et dont l'imprimante n'est pas au standard EPSON , doivent fournir avec leur demande d'achat les caractères de contrôle suivants s'ils veulent bénéficier des possibilités d'impression du directory :

- \* Passage en caractères condensés et retour à la taille normale.
- \* Mot de contrôle du saut de page à trois lignes ou saut de page complet selon préférence.

Demands à adresser à :

International DAI Club Bordeaux  
C/O Bruno Delannay  
Res. Les Acacias Bt.B3  
Avenue de Saige  
33600 PESSAC (FRANCE)

**POUR LES PROGRAMMES DISTRIBUES  
PAR DAINAMIC VOUS POUVEZ  
TOUJOURS PASSER PAR I.D.C.Bx.  
QUI SE CHARGERERA DE SATISFAIRE  
VOS DEMANDES**

sosososososososososososos  
 o o  
 s S.O.S. LECTEUR s  
 o o  
 sosososososososososososos

Cette rubrique est ouverte à ceux qui ont des questions à poser aux autres ou à ceux qui nous ont posé des questions auxquelles nous n'avons pas su répondre!!

Mr. P. Verhoost, 89 Rue de l'école, 5158 DAVE, Belgique, désire entrer en contact avec tout utilisateur du X-DOS de N.P. Looije (Pays-Bas), le plus rapidement possible.

Mr. F.L.Menciàre désirerait savoir comment, sur Ken-Dos, on peut charger en autostart ou avec un lanceur un programme en langage machine descendant en RAM en dessous de #29B et qui doit être lancé par RUN. Envoyer la réponse à I.D.C.Bordeaux

Les utilisateurs de Ken-Dos voudraient entrer en contact avec les possesseurs de X-DOS qui ont pu mettre sur la même carte RAM les EPROMS du Ken-DOS (dans une 27128) et le X-DOS, pour échanger entre Ken-Dos avec ou sans X-DOS. Envoyer la réponse à I.D.C.Bordeaux

VC. 1541 - X-BUS : faites parvenir à Luc D'ARANTES D.C.A. tout ce que vous avez fait, écrit, traduit, ou des explications sur ses différentes possibilités en vue de la préparation d'articles sur ce système.

Désires renseignements, programmes, pour relier DEUX DAI par la RS 232. But final : relier un DAI + Ken-Dos à un DAI + V.C. 1541 sans passer par un modem ... et les PTT (cher!). Documents à envoyer à Luc D'ARANTES.

**PERDU !**

La logithèque du club se trouve amputée de certains logiciels suite à une fausse manip sur les disquettes de sauvegarde !!!

Aidez nous à la reconstituer et à l'enrichir :

recette (gestion et comptabilité)  
conquest of memory alpha (jeu de role I face 2)

Envoyez SVP votre secours à I.D.C. Bordeaux !!!

**PETITES ANNONCES**

**ACHAT**

\*\* Mr. Morin Claudius, 6 Rue Robert Houdin, 14000 Caen. Tel 31.85.15.32.

ACHETE des K7 DCR achat en nombre si prix intéressant.

**VENTES**

Beaucoup de personnes pour brader leur matériel. Que ceux qui restent en profitent pour compléter leur configuration avec le périphérique désiré !!

\*\* Mr. Vandermeersch Marc, 17 Av. du Vert Bocage, B-1410 Waterloo, Belgique.

Vends pour DAI 2 Kits pour Basic V1.2 (3 eproms+adaptateur) 1000 FB le Kit ainsi que 22 disquettes reprenant la quasi totalité des softs pour DAI en KENDOS 2 \* 80 Tck pour 3000 FB.

\*\* Mr. D. Bollé, Streeklaan 2, B-3060 Bertem, Belgique  
Tel. 32/16/48.97.84 après 18h 30  
ou le W.E. 32/16/20.10.15 de 9h 30 à 17h 30.

Vends : DAI + DCR+ TOS + Cables + Moniteur couleur Barco + Documentation. Ensemble pour +/- 35 000 FB

\*\* Mr. Duluins Fabrice, 4 Allée Tour Renard, B-1400 Nivelles, Belgique.

Vends : DAI + DCR + TOS + Cables + 6 K7 DCR + des Kilos de documentation. Prix à débattre.

\*\* Je suis un DAI de Septembre 84 en parfaite santé et je cherche un nouvel acquéreur sympathique, qui puisse continuer à m'aimer. Je ne suis pas cher : 300FF et en plus je suis fourni avec de nombreux programmes (une dizaine de K7), le manuel d'utilisation et des revues.

Alors si tu veux passer d'agréables heures en ma compagnie, contactes : Cimbalk Bruno, 12 Rue des Crêtes, 62118 Biache St. Vaast.

\*\* Mr. D'Haene Filip , 6 Blvd. P.H. Spaak , 7900 Leuze en Hainaut , Belgique.

Vends : DAI environ 2500 FF Avec Basic V1.2 et V1.2 Ken Dos 2\*800K octets = 38.000FB soit environ 6.400 FF. Ou l'ensemble pour 8.900FF

Pour tout achat de l'ensemble , donne collection complète des Dainamic , Daiclics , Firmware Manual et 35 disquettes de programmes.

\*\* Mr. Rouan Jean Claude , 3 Rue de la Fontaine St. Gervais 60300 Aumont en Halatte

Vends un Firmware Manual

\*\* Mr. Menciàre F.L. , CORMICY, 51220 Hermonville, France tel: 26.61.30.25

Vends Floppy DAI 2x80K en panne mais réparable : 1000FF  
Floppy DAI 2x320K en parfait état de marche, avec disquette DOS 2.0 et 2.1 : 4500FF  
DAI P.C. 1980, parfait état de marche 1500FF)  
Manuel d'utilisation du DAI , neuf : 60FF

**ATTENTION ! !**

Un réparateur de DAI sur PARIS !! contacter :

Mr. LEBLOND M. Tel : 69.96.02.51  
12 Rue de Morsang  
91600 Savigny/ Orge

ou C/O I.G.I. Tel : 45.31.11.19  
10 Rue (illisible !!) ( Louis Ricat ??)  
75015 Paris

# I D C

**V** Présentation des services

**B** Les annonces

**B** Boîtes aux lettres

**D** Dates à retenir

**E** Informations générales

**C** Annuaire

Pour lire directement vos messages,  
tapez : MES + **ENVOI**

choix du mot-clé + **ENVOI**