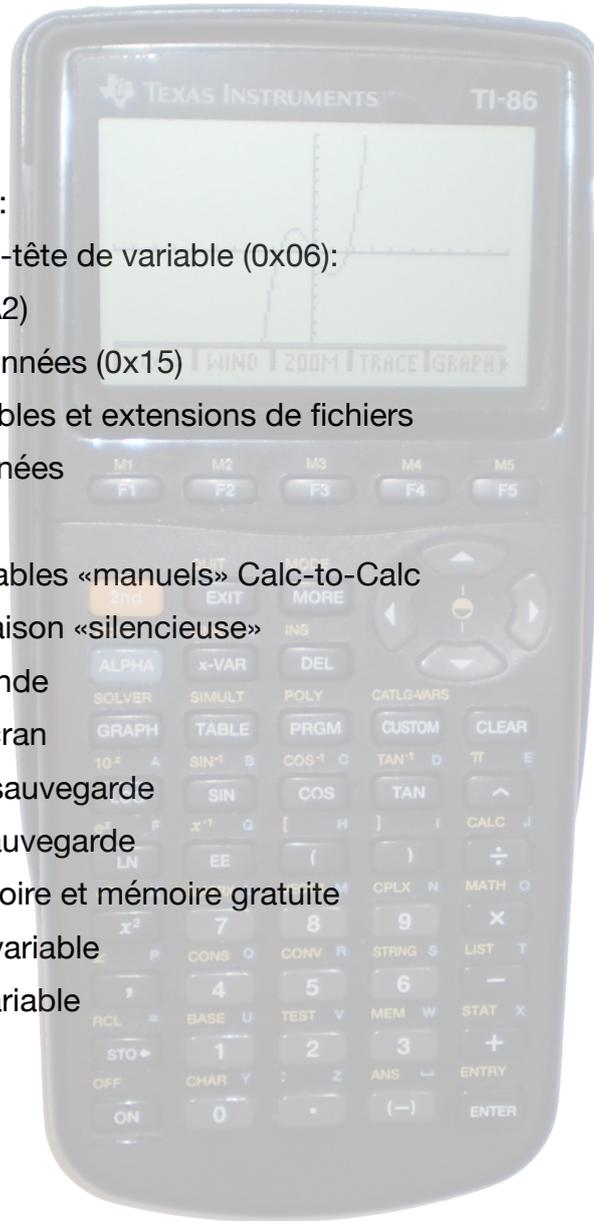


TI-86

(port 7) Link

A. HARDWARE:	2
B. SOFTWARE:	3
1. Paquets:	3
2. ID Machine:	4
3. ID Commande:	4
4. Le format d'en-tête de variable (0x06):	5
5. Demande (0xA2)	6
6. Format des données (0x15)	6
7. Types de variables et extensions de fichiers	7
C. Transferts de données	8
1. Appairage	8
2. Transferts variables «manuels» Calc-to-Calc	8
3. Capacité de liaison «silencieuse»	9
4. La télécommande	9
5. Le vidage d'écran	10
6. Recevoir une sauvegarde	10
7. Envoi d'une sauvegarde	11
8. Liste de répertoire et mémoire gratuite	11
9. Recevoir une variable	12
10. Envoi d'une variable	12

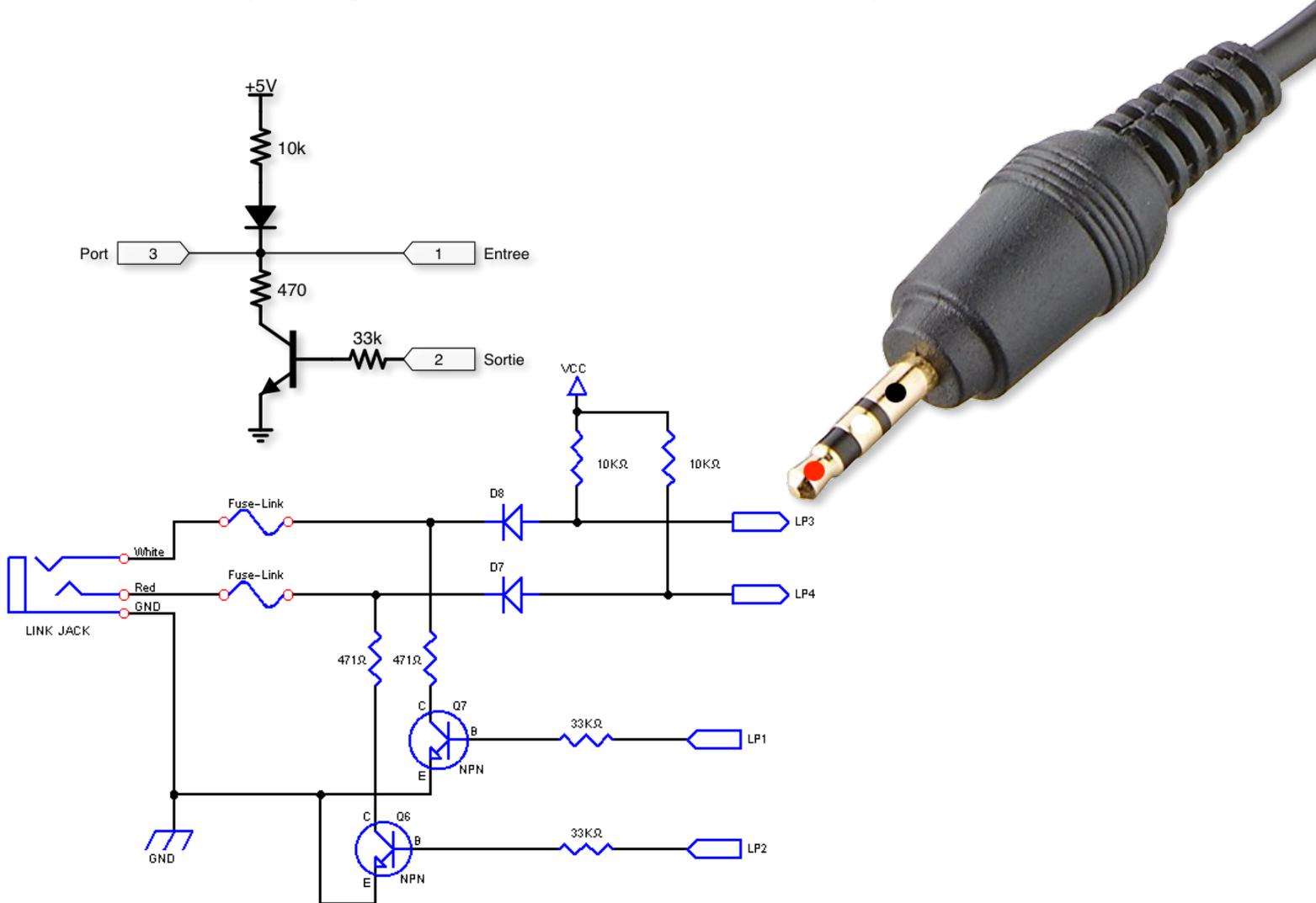


A. HARDWARE:

Le bus TI est une interface série à deux fils composée d'une ligne « rouge » et d'une ligne « blanche ».

Les deux lignes sont bidirectionnelles, donc les sorties doivent être de type « collecteur ouvert » ou « drain ouvert ».

Chaque ligne est connectée à la tension d'alimentation via une résistance de pull-up. Une ligne est alors à l'état logique haut lorsqu'aucun des périphériques connectés ne conduit la ligne et logique bas si un ou plusieurs déclenche la ligne bas.

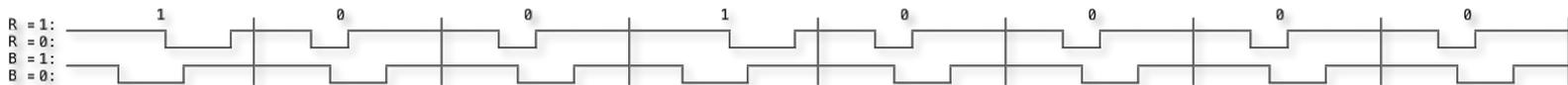


B. SOFTWARE:

1. Paquets:

Un exemple de paquet :

l'octet représenté est 0x09 (00001001 en binaire) le premier bit transmis est le bit le moins significatif (LSb) et le dernier est le bit le plus significatif (MSb) comme avec le flux série RS232.



Légende: R est le fil rouge et B est le fil blanc.

Chaque bit est composé de deux impulsions sur deux fils différents.

C'est l'ordre de ces deux impulsions qui déterminent l'état du bit.

Enfin, le taux de transfert est de 45 à 50 Kbits / seconde.

Cette calculatrice possède la capacité de liaison silencieuse.

La transmission de données entre la calculatrice et le PC est faite par paquets. Un paquet a le format suivant:

Position	Longueur	Nom	Description
0	1	MID	Machine ID
1	1	CID	Commande ID
2	2	PL	Longueur du paquet des données en octets si données
4	n	DATA	Données
4+n	2	CHK	Checksum si les données sont jointes. La somme de contrôle est la somme de tous les octets de données

La somme de contrôle indique à la machine réceptrice si les données d'un paquet avec des données ont été correctement reçues. Il s'agit en fait des 16 bits les moins significatifs

la somme de tous les octets de données. Le PL peut être différent de 0 sans données et somme de contrôle. En effet, ce mot est une information.

Sinon, c'est un paquet normal. Exemple:

```

+---ID Machine=Ordinateur vers TI-86
|   +---ID Commande=Skip/Exit
|   |   +---Data length=0001 bytes
|   |   |   +---Data=01 (Exit pressed)
|   |   / \ |   Checksum=0001
    
```

Un paquet typique serait: 06 36 01 00 01 01 00

Les paquets peuvent contenir jusqu'à 65535 octets dans la partie données.

2. ID Machine:

L'octet ID machine identifie la machine afin que la calculatrice puisse décider accepter ou non la transmission. Voici les valeurs possibles:

ID Machine	Description
06	PC -> TI86
86	TI86 -> PC ou TI86 -> TI86

3. ID Commande:

L'octet ID de commande spécifie le type de paquet.

Legende:

* signifie que des données sont jointes

signifie que l'identifiant du périphérique est ignoré avec cette commande.

\$ signifie que c'est une commande silencieuse.

ID	Leg	Nom	Description
06	*--	VAR	Inclut un en-tête de variable std (utilisé dans la réception)
09	---	CTS	Utilisé pour signaler OK pour envoyer une variable
15	*--	XDP/DATA	Transmettre le paquet de données (données pures)
36	*--	SKIP/EXIT	Skip / exit - code de rejet d'un octet (*)
56	---	ACK/OK	Reconnaissance
5A	---	ERR	Erreur de contrôle: envoyer le dernier paquet à nouveau
6D	--\$	SCR	Demande de capture d'écran
92	---	EOT/DONE	Fin de la transmission: plus de variables à envoyer
A2	*-\$	REQ	Demande de variable - inclut une variable standard
C9	*-\$	RTS	Demande d'envoi - comprend un en-tête de variable « remboursé »

(*) Codes de rejet:

Valeur	Description
01	EXIT, l'intégralité du transfert a été annulé
02	SKIP, la variable actuelle a été ignorée par l'utilisateur
03	MEM, plus de mémoire

4. Le format d'en-tête de variable (0x06):

Un en-tête de variable est utilisé chaque fois qu'une information doit être transmise. L'en-tête de variable est la donnée d'un paquet lorsque la commande est 06. L'en-tête de variable a une longueur maximale de 12 octets. Il a le format suivant:

Offset	Longueur	Nom	Description
0	1	MID	ID de la machine
1	1	CID	ID de la commande, 06 dans ce cas
2	2	PI	Longueur du paquet (LL-HH)
2	2	DI	Longueur des données (LL-HH)
3	1	Ty	Type de variable (voir paragraphe suivant)
4	1	NI	Longueur nom (0..8)
5	n	Nom	Nom (pas terminé par zéro)
6+n	n	CHK	Checksum

Remarque:

Si l'en-tête de la variable est "rembourré" (CID = C9), le nom est toujours 8 octets de long, mais rembourré à droite avec 0x20. Ce format est seulement utilisé lors de la demande d'envoyer une variable à la TI-86 ou lors de la réception d'une entrée du répertoire.

Si la variable est WIND ou DIR, alors le nom est un 0 octet unique.

Si l'en-tête décrit une sauvegarde de mémoire, l'en-tête de variable prendra le format suivant:

Offset	Longueur	Description
0	2	Longueur de données du premier paquet de données
2	1	Type de variable = 1D
3	2	Longueur de données du deuxième paquet de données
5	2	Longueur de données du troisième paquet de données
7	2	Longueur de données du quatrième paquet de données

5. Demande (0xA2)

Une demande est utilisée chaque fois que l'ordinateur demande une action.

Le format d'une requête est identique au format d'en-tête variable mais la longueur de données est nulle et le nom de fichier n'est pas complété par des caractères d'espace.

Il a le format suivant:

Offset	Longueur	Nom	Description
0	1	MID	ID de la machine
1	1	CID	ID de la commande, 06 dans ce cas
2	2	PI	Longueur du paquet (LL-HH)
2	2	DI	Longueur des données (LL-HH)
3	1	Ty	Type de variable (voir paragraphe suivant)
4	1	NI	Longueur nom (0..8)
5	n	Nom	Nom (pas terminé par zéro)
6+n	n	CHK	Checksum

Pour une liste d'annuaire, le type de demande est Ty = 15 et le nom est une chaîne nulle: 00.

6. Format des données (0x15)

Une partie de données est utilisée chaque fois que des données ou un contenu variable doivent être transmis.

La partie de données est identique à celle de la mémoire. Il a le format suivant:

Offset	Longueur	Nom	Description
0	1	MID	ID de la machine
1	1	CID	Nom de la commande, 15 dans ce cas
2	2	DI	Longueur des données (LL-HH)
3	n	DATA	Données
6+n	2	CHK	Checksum

où $HHLL + 4 + \text{strlen}(\text{Nom})$ est la longueur de la variable dans le menu MEM.

7. Types de variables et extensions de fichiers

Type de variable:

0	nombre réel	REAL	.86n
1	nombre complexe	CPLX	.86c
2	vecteur réel	VECTR	.86v
3	vecteur complexe	CVECT	.86v
4	vraie liste	LISTE	.86l
5	liste complexe	CLIST	.86l
6	matrice réelle	MATRX	.86m
7	matrice complexe	CMATR	.86m
8	vraie constante	CONS	.86k
9	constante complexe	CCONS	.86k
0A	équation	EQU	.86e
0C	chaîne	STRNG	.86s
0D	Fonction GDB	GDB	.86d
0E	GDB polaire	GDB	.86d
0F	GDB paramétrique	GDB	.86d
10	Différent GDB	GDB	.86d
11	image	PIC	.86i
12	programme	PRGM	.86p
13	gamme	GAMME	.86r
14	écran	SCRN	
15	annuaire	DIR	
17	Paramètres de la fenêtre de fonction	WIND	.86w
18	Paramètres de la fenêtre polaire	POLAIR E	.86r
19	Paramètres paramétriques	PARAM	.86r
1A	Différentielles paramètres	DIFEQ	.86r
1B	taille de la fenêtre enregistrée	ZRCL	
1D	sauvegarde		.86b
2A	équation	EQU	.86e

C. Transferts de données

1. Appairage

Dans la plupart des cas, le protocole est le suivant.

* En envoi (PC -> TI):

- 1 - PC: en-tête VAR ou RTS var (nom, type, ...)
- 2 - TI: ACK continue
- 3 - TI: données d'attente CTS
- 4 - PC: ACK continue
- 5 - PC: partie données XDP
- 6 - TI: ACK continue
- 7 - PC: EOT fin de transmission
- 8 - TI: ACK fin du transfert

Si elles sont plusieurs variables, répétez les étapes 1 à 7. Il paraît que L'étape 8 n'est pas utilisée sur la TI86.

* En réception (TI -> PC):

- 1 - PC: variable requête REQ
- 2 - TI: ACK continue
- 3 - TI: en-tête VAR var
- 4 - PC: ACK continue
- 5 - PC: données d'attente CTS
- 6 - TI: ACK continue
- 7 - TI: partie de données XDP
- 8 - PC: fin de transfert ACK

2. Transferts variables «manuels» Calc-to-Calc

La TI-86 suit ses prédécesseurs dans le protocole de transfert variable utilise, ce qui lui permet de communiquer avec une TI-85. Le menu SND85 uniquement invite la TI-86 à envoyer un octet d'identification de la machine TI-85 et à ne pas transférer certains types de données.

Voici l'idée générale (l'unité de transmission est "xtre", l'unité de réception est "recv"):

- 1) XMIT -> RECV Variable Header - spécifie la variable envoyée

- 2) <- Reconnaître
- 3) <- Continuer ou Passer / Quitter: Si vous passez à l'étape 6,
Si la sortie passe à l'étape 8.
- 4) -> Reconnaître
- 5) -> Données – les données variables
- 6) <- Reconnaître
- 7) -> EOT ou un autre en-tête de variable: si en-tête de variable,
passez à l'étape 2.
- 8) <- Reconnaître

Remarque: Le paquet "Skip / Exit" a un octet de données: 01 signifie que le la variable a été ignorée, et 02 signifie que le lien a été annulé, et 03 signifie 'hors mémoire'.

Exemple: 86 36 01 00 xx xx 00 où xx = 01,02 ou 03

3. Capacité de liaison «silencieuse»

TI a intégré une capacité "Silent Link" dans la TI-86 .Il vous permet d'envoyer et de recevoir des variables à partir de la calculatrice sans mettre la calculatrice en mode lien. Le seul inconvénient est que la fonction d'envoi silencieux remplace automatiquement toutes les données cette variable sur la calculatrice. Par conséquent, tout programme qui intègre cette capacité doit également vérifier pour s'assurer que cela ne fonctionne pas se produire.

Le paragraphe suivant utilise la capacité du lien silencieux ...

4. La télécommande

Pas de télécommande.

5. Le vidage d'écran

PC: 06 6D 00 00 PC demande un vidage d'écran

TI: 86 56 00 00 TI réponse OK

86 15 00 04

<... 0x0400 = 1024 octets ...> CHK TI envoie l'écran bitmap

PC: 06 56 00 00 Réponse du PC OK

L'écran de la calculatrice est une matrice de pixels de 128 colonnes et 64 lignes.

Donc, il y a $128 \times 64 = 8192$ pixels -> 1024 octets.

$128 = 16 \times 8$ bits -> 16 octets et

$64 = 8 \times 8$ bits -> 8 octets.

Donc, $16 \times 64 = 1024$ et $1024 \times 8 = 8192$ bits.

6. Recevoir une sauvegarde

La sauvegarde est divisée en trois ou quatre parties (car la TVA du TI86 est divisé en trois parties) en fonction de la version ROM et / ou de l'utilisation Mémoire.

TI: 86 06 09 00 xx XX Ty PC reçoit l'en-tête de sauvegarde

yy YY zz ZZ ss SS CHK

PC: 06 56 00 00 OK

06 09 00 00 continuer

TI: 86 56 00 00 OK

86 15 xx XX <... données ...> CHK partie 1

PC: 06 56 00 00 OK

TI: 86 15 yy YY <... données ...> CHK partie 2

PC: 06 56 00 00 OK

TI: 86 15 zz ZZ <... données ...> CHK partie 3 (si partie 2 > 64KB)

PC: 06 56 00 00 OK

TI: 86 15 ss SS <... données ...> CHK partie 4

PC: 06 56 00 00 OK

XXxx est la taille de la première partie, YYyy la deuxième partie, ZZzz le troisième part et SSs la quatrième partie si ZZzz est différent de 0.

7. Envoi d'une sauvegarde

La sauvegarde peut être divisée en 3 ou 4 parties (selon la taille de la troisième bloc: moins de 64 Ko ou non).

```
PC: 06 06 09 00 xx XX Ty PC envoyer l'en-tête de sauvegarde (Ty = 1D)
    yy YY zz ZZ ss SS 85 03
TI: 86 56 09 00 la réponse TI OK et la longueur du paquet
    86 09 09 00 continuer
PC: 06 56 00 00 OK
    06 15 xx XX <... données ...> CHK partie 1
TI: 86 56 00 00 OK et longueur du paquet
PC: 06 15 yy YY <... données ...> CHK partie 2
TI: 86 56 00 00 OK et longueur du paquet
PC: 06 15 zz ZZ <... données ...> CHK partie 3
TI: 86 56 00 00 OK et longueur du paquet
PC: 06 15 ss SS <... données ...> CHK partie 4
TI: 86 56 00 00 OK et longueur du paquet
PC: aucun ACK
```

8. Liste de répertoire et mémoire gratuite

```
PC: 06 A2 05 00 00 00 15 01 00 16 00 demande un dir. liste
TI: 86 56 05 00 OK
    86 15 03 00 AA BB CC CHK Mem libre
PC: 06 56 00 00 OK
-> répéter
TI: 86 xx 0C 00 LL HH envoie le varname (xx = 06)
    Ty NI <... VarName ...> CHK
PC: 06 56 00 00 OK
-> jusqu'à xx = 92
PC: 06 56 00 00 OK
```

La partie entre les flèches doit être répétée alors que xx = 06 (en-tête var).

Remarque: la valeur de memoire libre est composée de 3 octets comme suit:
memfree = AA * 65536 + CC * 256 + BB

9. Recevoir une variable

PC: 06 A2 XI Xh 00 00 Ty NI demande un var par son nom
<VarName> CHK

TI: 86 56 XI Xh OK

86 06 0C 00 LL HH Ty NI TI répond le nom

<VarName> CHK de la variable (8 caractères)

PC: 06 56 00 00 OK

06 09 00 00 prêt à recevoir

TI: 86 56 00 00 OK

86 15 LL HH <... données ...> CHK TI envoie un contenu variable

PC: 06 56 00 00 OK

10. Envoi d'une variable

PC: 06 C9 0C 00 LL HH Ty NI envoie la variable
<... VarName ...> CHK

TI: 86 56 0C 00 OK

86 09 0C 00 données d'attente

PC: 06 56 00 00 OK

06 15 LL HH <... données ...> CHK envoyer un contenu variable

TI: 86 56 LL HH OK

PC: 06 92 00 00 fin de transmission

TI: étrange, mais il n'y a pas de réponse