

MANUEL D'UTILISATION
MODULE USB 4 ENTREES
CODEURS INCREMENTAUX
TC/USB-4I11A ou 4I1V



Electronique
Informatique
Industrielle



Révision document : 08
Révision module: 01

3C Concept S.A – 17 rue d'Ahuy, 21121 HAUTEVILLE LES DIJON
Tel : 0380.550.130 - Fax : 0380.556.449
www.3c-concept.fr

Sommaire

DESCRIPTION.....	2
SPECIFICATIONS	2
CARACTERISTIQUES ELECTRIQUES	3
PRESENTATION DU MODULE.....	4
BROCHAGE	5
SYNOPTIQUE GENERAL	6
FILTRAGE DES ENTREES	6
PROCEDURE D'INSTALLATION SOUS WINDOWS XP	7
PROCEDURE D'INSTALLATION SOUS WINDOWS 2000	11
EN CAS DE PROBLEME	15
PROGRAMME DE DEMONSTRATION.....	16
UTILISATION DE LA DLL TC/USB-4I	18
F.A.Q	23

Description

Le module TC/USB-4I est un module d'interface pour PC, comportant 4 entrées pour codeurs incrémentaux. Grâce à l'utilisation du port USB, ce module permet d'obtenir une grande facilité d'installation et de maintenance. Pour chaque codeur, on a un connecteur SubD 15 broches permettant le raccordement des signaux A / B / Z.

Le module est livré avec une DLL permettant de l'interfacer avec les langages de programmation les plus utilisés.

Il se présente dans un boîtier plastique (polycarbonate), il se fixe sur rail DIN 35mm, ses connecteurs sont débrochables.

Cette notice couvre les produits TC/USB-4IS11A (pour capteurs possédant des sorties 11 μ A) et TC/USB-4I1V (pour capteurs 1Vcc). Les fonctions logicielles sont identiques pour les deux produits (également identique aux produits TTL ou HTL)

Spécifications

- 4 x 3 entrées sinusoïdales 11 μ A ou 1Vcc selon version
- Fréquence maxi programmable entre 50KHz et 3MHz
- Interface USB isolée compatible USB 1.1 et USB 2.0
- Montage sur rail DIN 35mm
- Dimensions : (L)160 x (l)90 x (h)70
- Alimentation externe 10-24Vdc
- Alimentation capteurs 5V fournie (1.2A maxi au total)
- Consommation typique de 5W sous 24Vdc.

Utilisable avec la plupart des codeurs incrémentaux à sorties sinusoïdales 11 μ A ou 1Vc/c et notamment avec les produits Heidenhain MT2501, MT2581, ROD480 ...

Caractéristiques électriques

Caractéristiques générales :

Nom	Description	Min	Typ	Max	Unités
Vcc	Tension d'alimentation	10	24	28	volts
Iusb	Courant sur la ligne USB		50		mA
Pmax	Puissance consommée (@ 24V)	1	5	10	W

Entrées codeurs en mode 11 μ A :

Nom	Description	Min	Typ	Max	Unités
Rin	Impédance de charge		120		ohms
Vmax	Tension limite admissible sur les entrées	0		5	volts

Entrées codeurs en mode 1Vc/c :

Nom	Description	Min	Typ	Max	Unités
Rin	Impédance de charge		120		ohms
Vmax	Tension limite admissible sur les entrées	0		5	volts

Caractéristiques dynamiques :

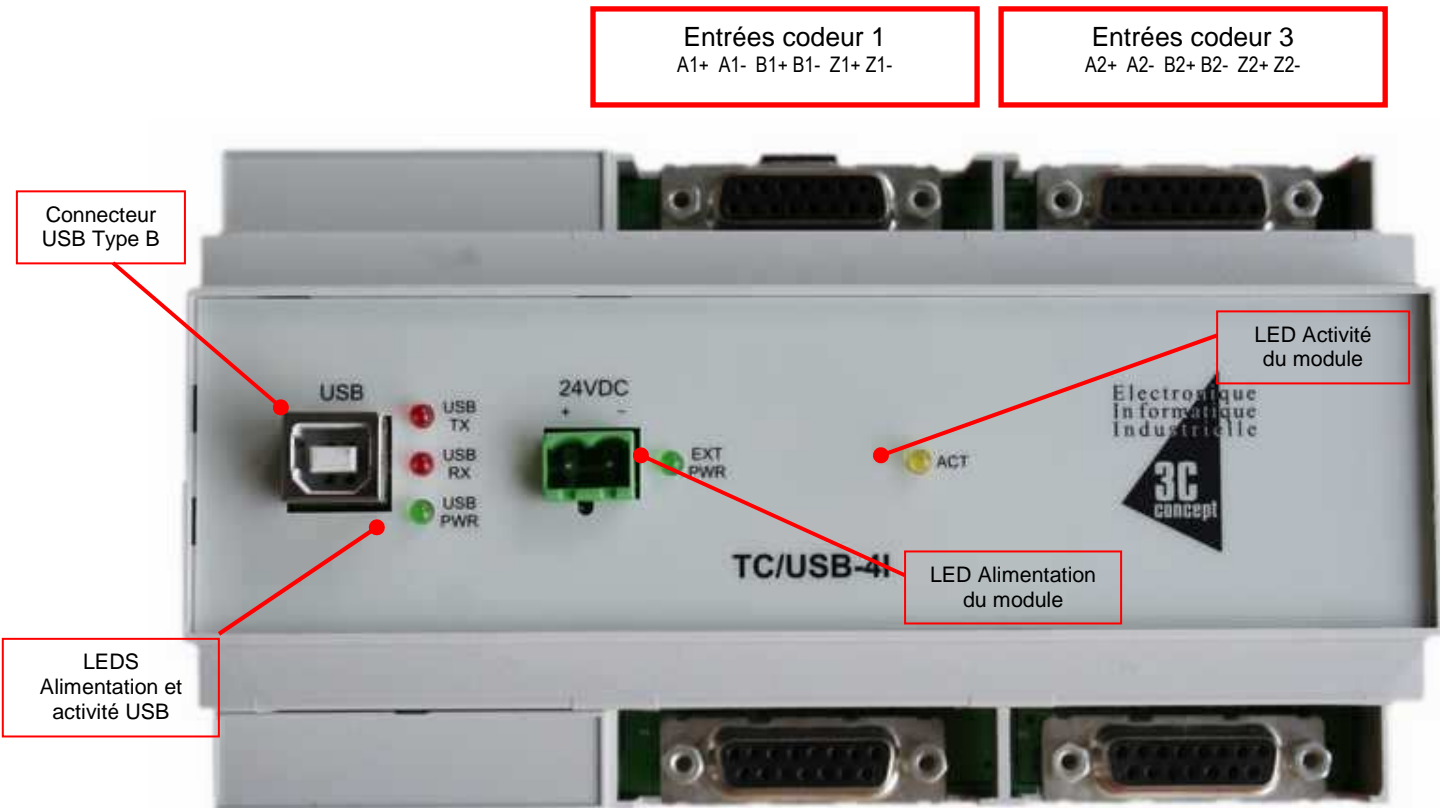
Nom	Description	Min	Typ	Max	Unités
Fe	Fréquence de comptage (A/B/Z) (1)			500	KHz
Fbft	Fréquence de la base de temps		3.20		MHz
Tw	Largeur impulsion (niveau 0 & 1)	300			ns

(1) avec fréquence de filtrage maximale sur les entrées

Spécifications USB :

- Nombre maxi de périphériques = 127
- Distance maxi entre périphériques = 5 m
- Possibilité d'utiliser des hubs USB
- Possibilité d'étendre les distances (exemple extremeUSB(R) www.icron.com)
- Hub industriel USB (4 ou 7 ports) voir hubport sur www.digi.com

Présentation du module



Z4- Z4+ B4- B4+ A4-A4+
Entrées codeur 4

Z3- Z3+ B3- B3+ A3-A3+
Entrées codeur 2

Description des Leds :

Led USB_TX : Led rouge, indique une activité sur la ligne USB, en direction du module

Led USB_RX : Led rouge, indique une activité sur la ligne USB, en direction du PC

Led USB_PWR : Led verte, témoin d'alimentation par la ligne USB

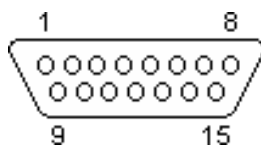
Led EXT_PWR : Led verte, témoin d'alimentation externe 10-24VDC

Led ACT : Led jaune, indique le bon fonctionnement du processeur interne du module.



Le module se monte sur Rail-DIN 35mm

Connecteurs codeurs 1 à 4



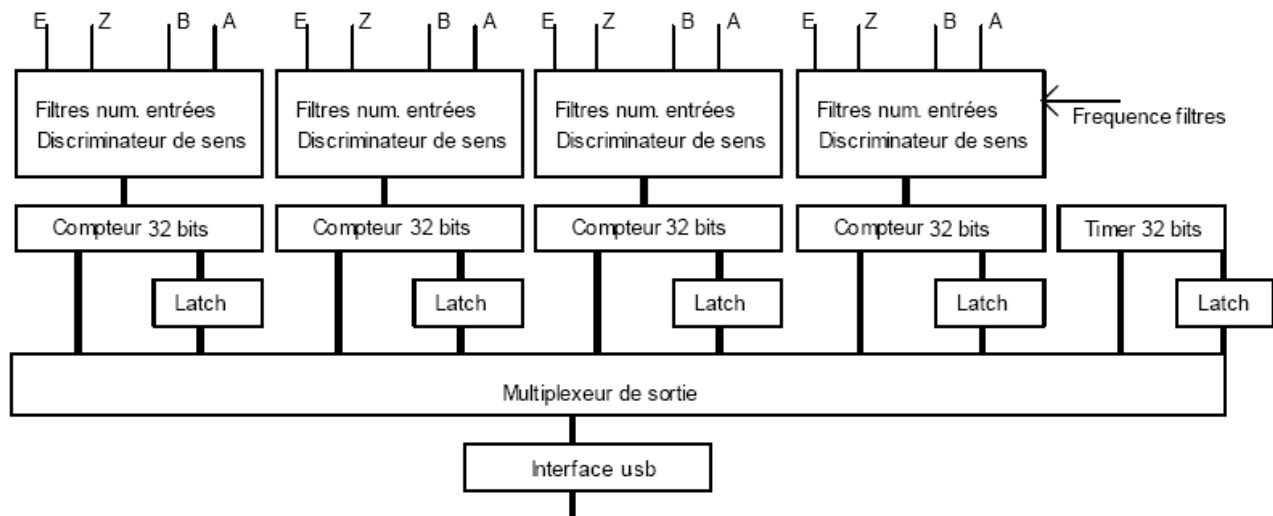
SubD 15 femelle (mâle sur câble codeur)

Broche	Signal
1	Ua1+
2	0V
3	Ua2+
4	+5V
5	-
6	-
7	Ua0-
8	-
9	Ua1-
10	0V
11	Ua2-
12	+5V
13	-
14	Ua0+
15	-

Ce brochage est normalement compatible avec les prises SubD 15 plots montées par Heidenhain. Nous vous invitons toutefois à vérifier le brochage de votre capteur.

Une alimentation 5V est disponible sur les bornes 4 et 12 permettant d'alimenter le capteur sous un courant maximum de 300mA par connecteur.

Synoptique général



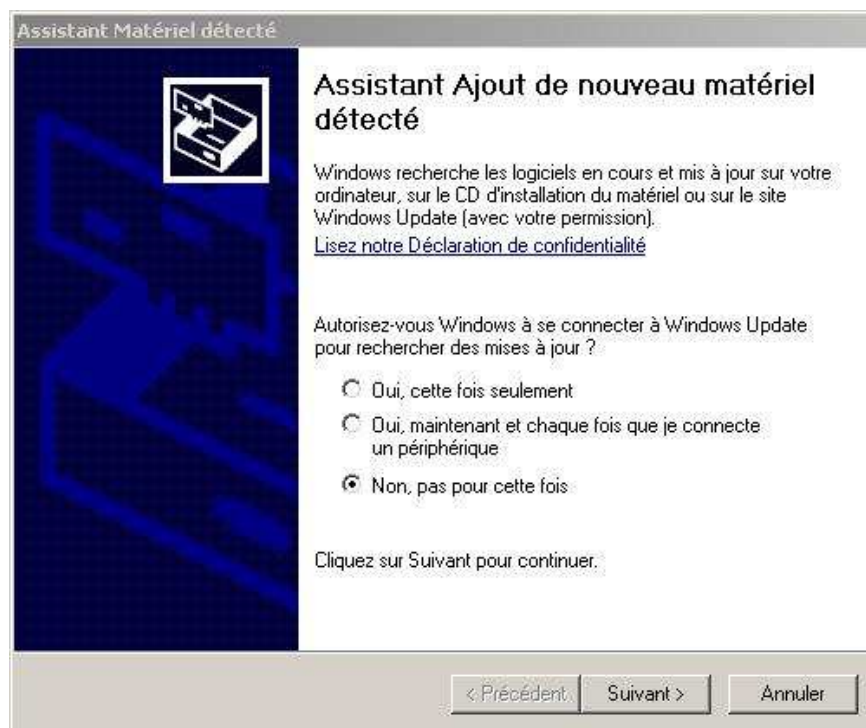
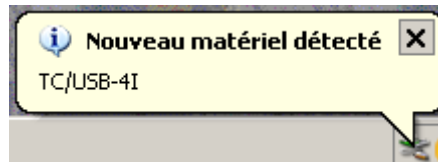
Filtrage des entrées

Le filtrage numérique mis à disposition de l'utilisateur permet de filtrer les « glitches » éventuellement présents sur les lignes d'entrées. Il n'est pas régi par la notion de fréquence de coupure, cette fréquence indique seulement la largeur minimale d'une impulsion. Si une impulsion sur la ligne est inférieure à cette valeur, l'impulsion n'est pas prise en compte.

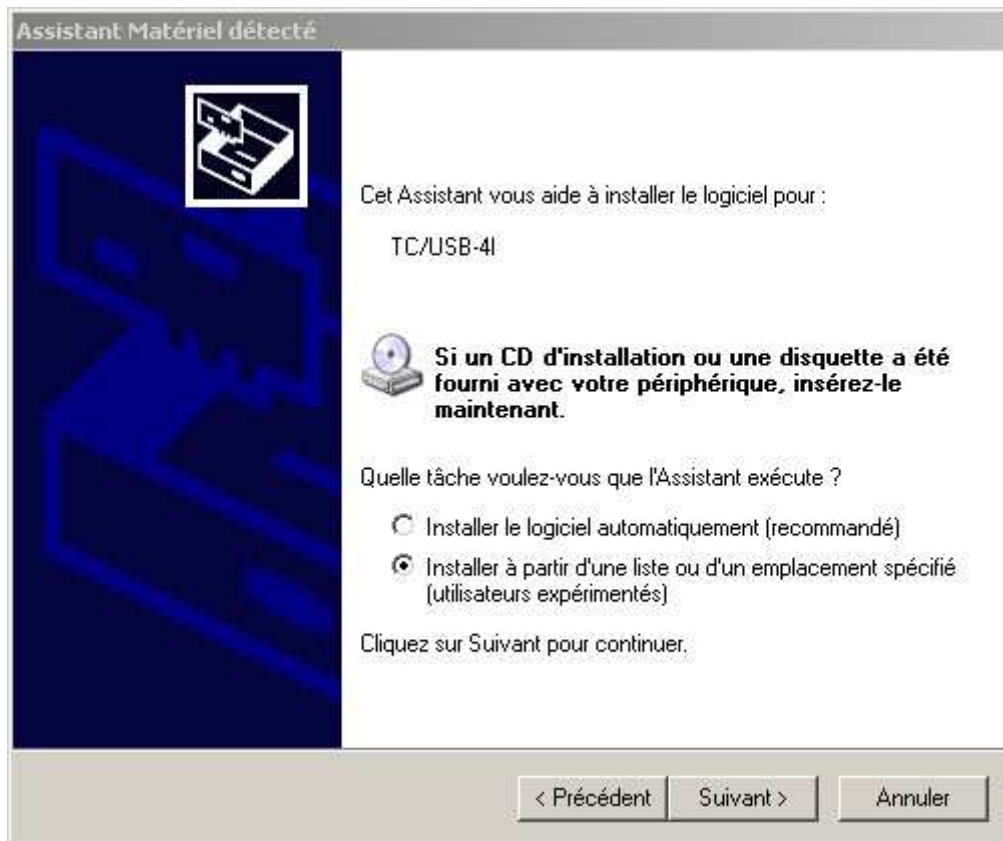
Procédure d'installation sous Windows XP

Le contenu du CD-ROM rassemble les drivers, le programme de démonstration, ainsi qu'une DLL et les fichiers d'entête pour concevoir ses propres programmes.

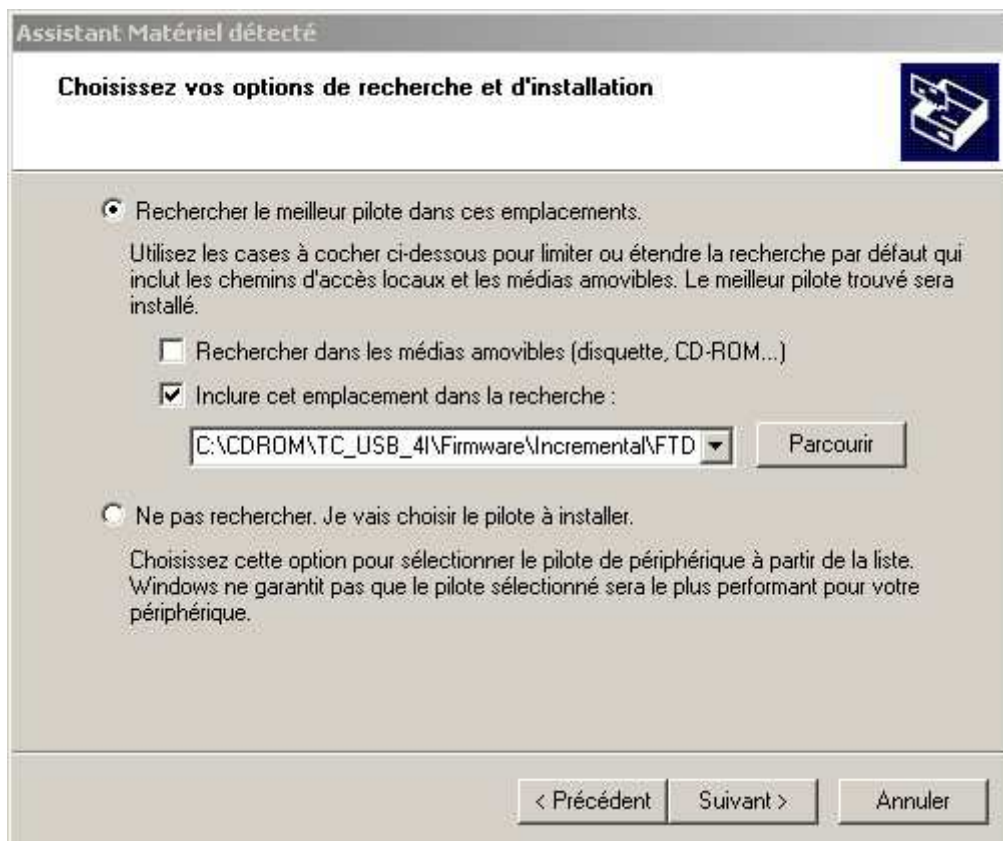
- Vous pouvez brancher le module: Windows détecte alors l'apparition d'un nouveau périphérique :



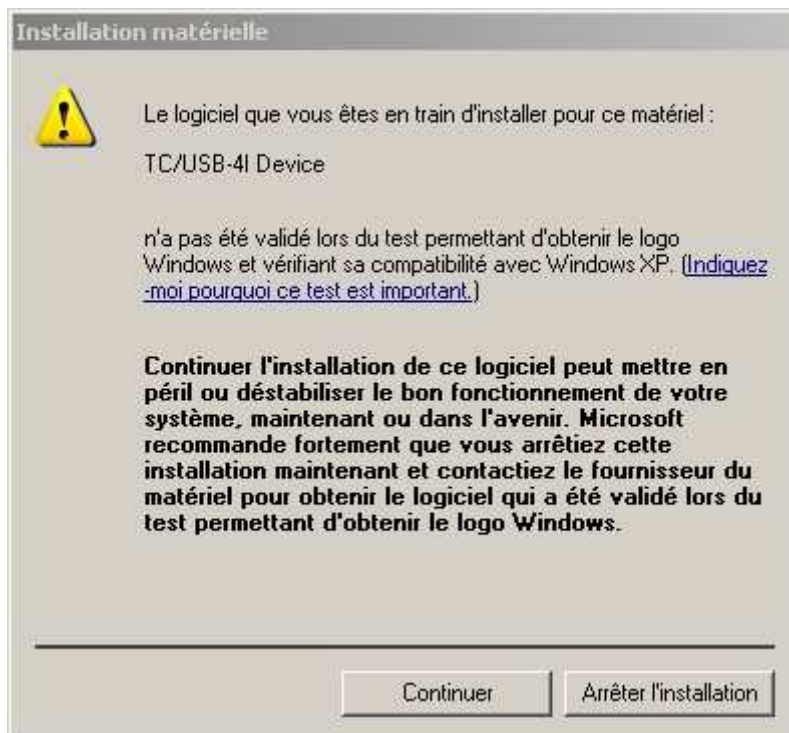
- Windows propose de se connecter au site Windows Update, Sélectionner « non pas pour cette fois »
- Sélectionnez « installer à partir... » , « Suivant »



- Dans la fenêtre suivante, cliquez sur « parcourir », cherchez sur le CDROM fournit le répertoire **TCUSB4I\Win32\Drivers**, cliquez sur Ok.



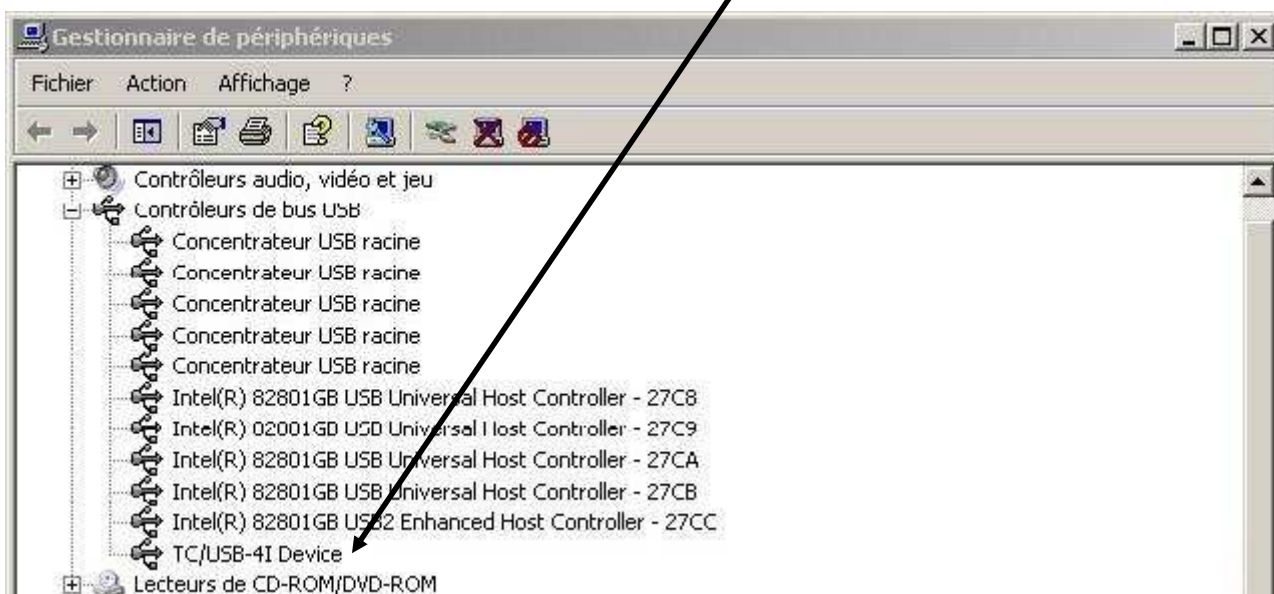
Lors de l'installation et le transfert de fichiers, WindowsXP indiquera que le pilote en cours d'installation n'est pas signé WHQL (ie :validé par Microsoft), c'est normal, veuillez cliquer sur « continuer ».



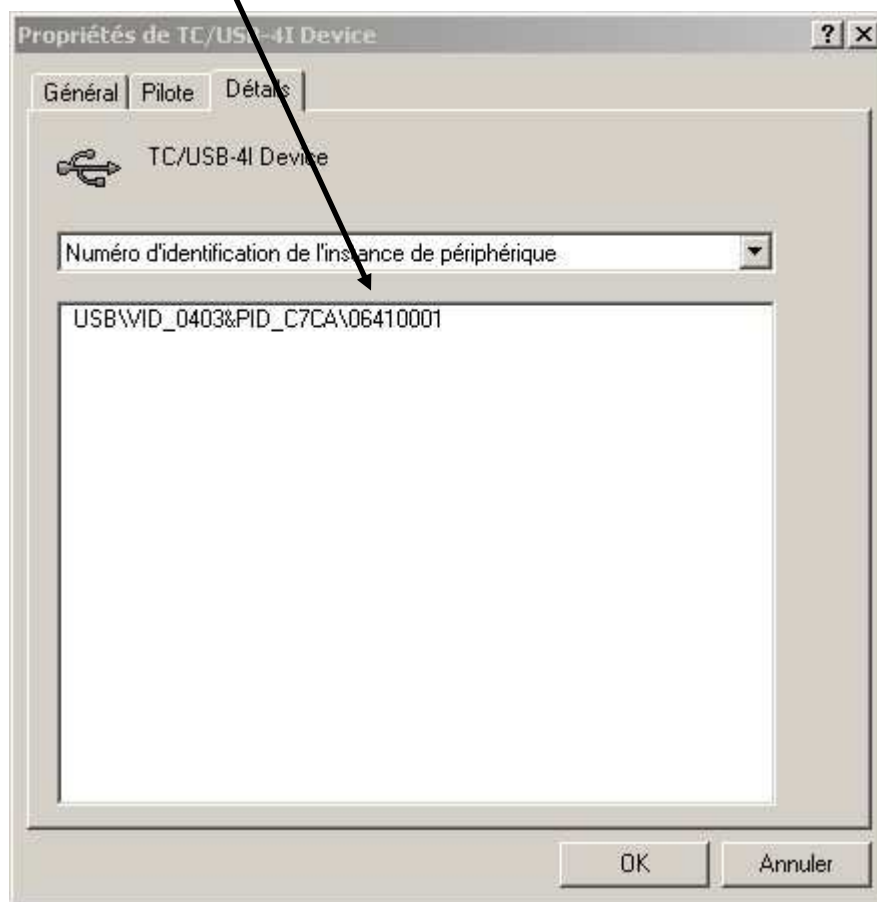
Une fois l'installation terminée, Windows affiche le message suivant :



A titre de vérification vous pouvez aller dans le panneau de configuration, système, gestionnaire de périphérique et vérifier ce qui suit :



De plus , on peut retrouver le numéro de série du module en cliquant sur « propriétés » de TC/USB-4I, onglet Détails :

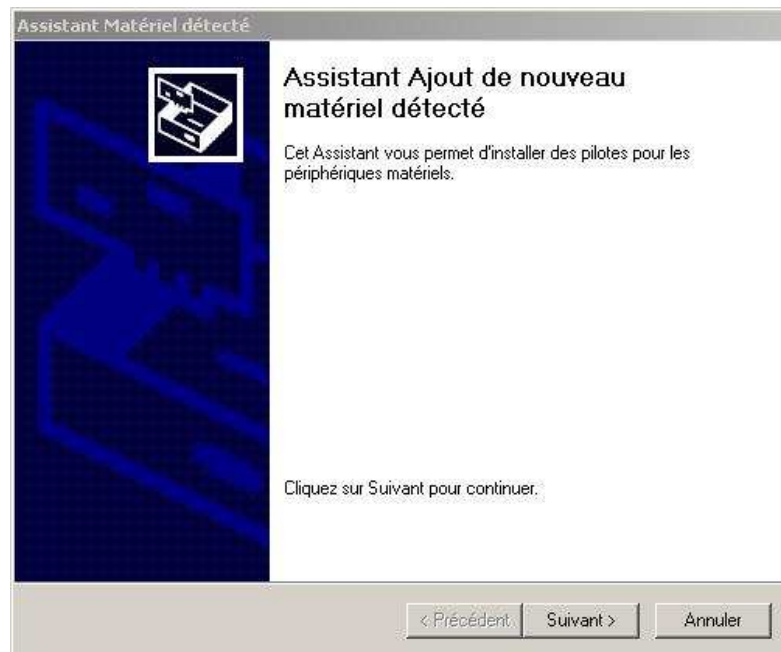


En dernier lieu, lancez le programme ***TCUSB4I\Win32\Programme de démonstration\DemoTCUSB4I.EXE*** présent sur le CD-ROM pour vérifier que l'installation s'est bien déroulée et que le module est accessible. (voir partie Programme de démonstration)

Procédure d'installation sous Windows 2000

Le contenu du CD-ROM rassemble les drivers, le programme de démonstration, ainsi qu'une DLL et les fichiers d'entête pour concevoir ses propres programmes.

- Vous pouvez brancher le module : Windows détecte alors l'apparition d'un nouveau périphérique :



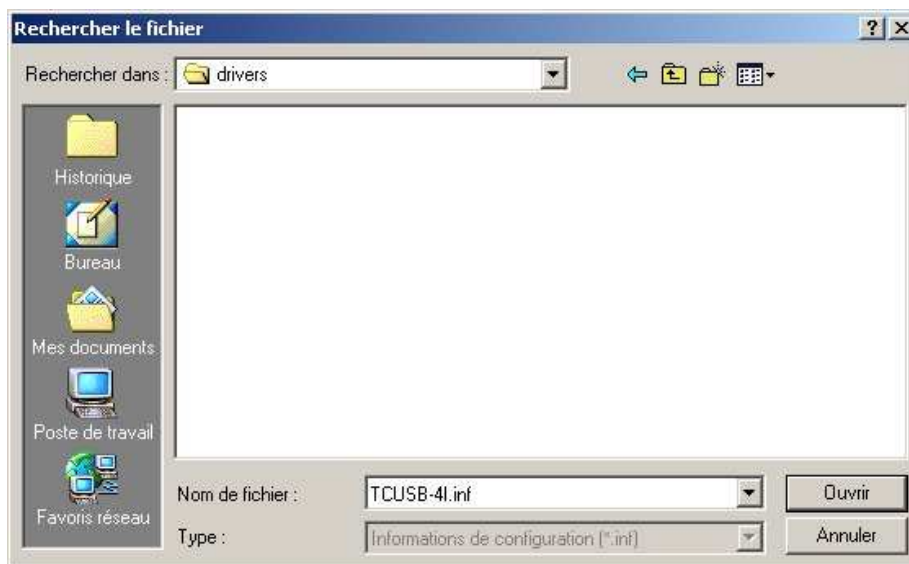
- Cochez « Rechercher un pilote ... » , « suivant »



- Windows vous propose différents emplacement de recherche, veuillez sélectionner « emplacement spécifique, suivant »

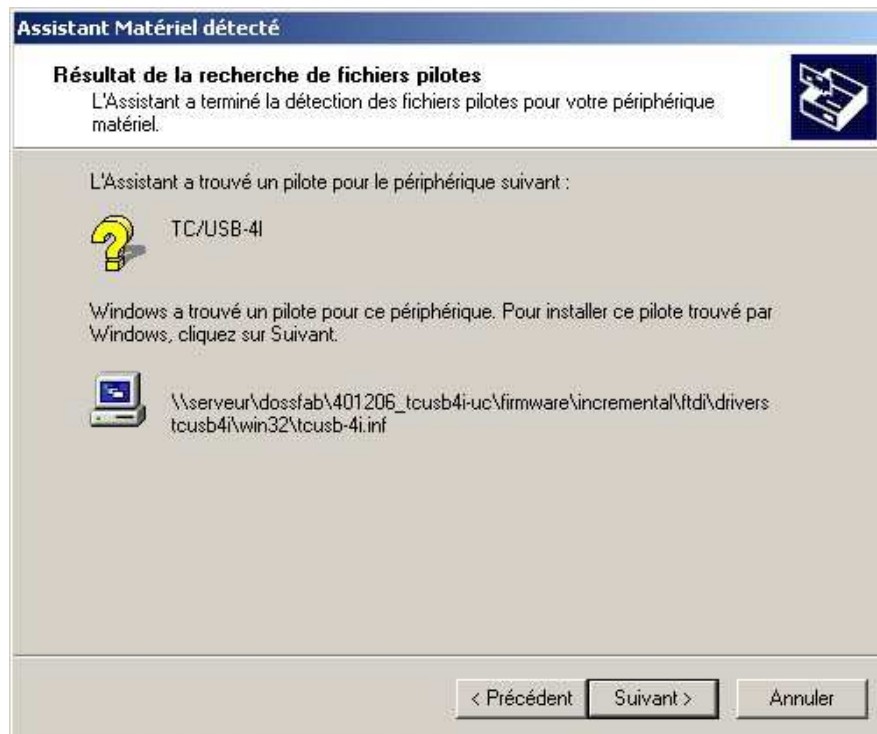


- Dans la fenêtre suivante, cliquez sur « parcourir », cherchez sur le CDROM fournit le répertoire **TCUSB4I\Win32\Drivers**, cliquez sur Ouvrir.





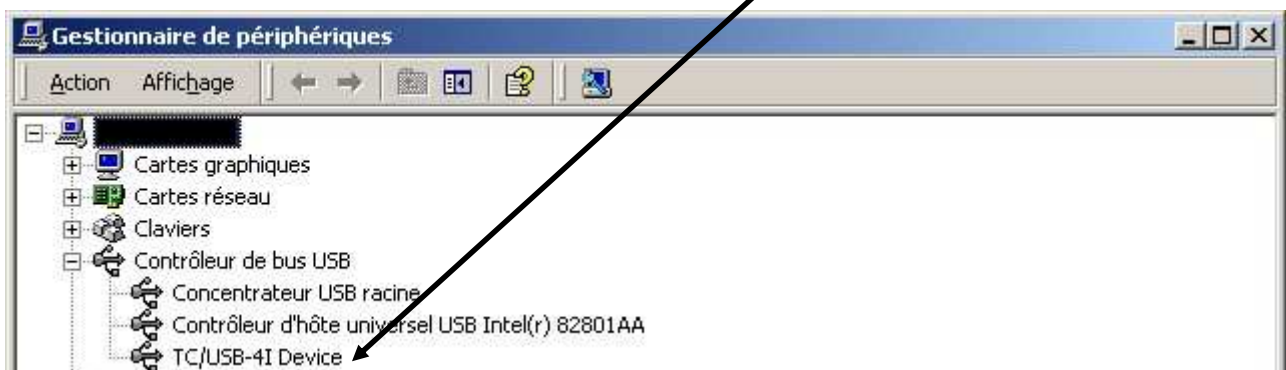
- Cliquez sur OK
- Windows affiche un résumé des informations récoltées, appuyer sur « suivant »



- Lorsque l'installation du driver est terminée, windows affiche l'écran suivant :



A titre de vérification vous pouvez aller dans le panneau de configuration , système, onglet Matériel, Gestionnaire de périphérique et vérifier ce qui suit :



En dernier lieu, lancez le programme **TCUSB4I\Win32\Programme de démonstration\DemoTCUSB4I.EXE** présent sur le CD-ROM pour vérifier que l'installation s'est bien déroulée et que le module est accessible. (voir partie Programme de démonstration)

- Point d'exclamation sur système/gestion de périphérique :

Un point d'exclamation jaune dans le gestionnaire de périphérique de Windows Xx indique qu'un matériel a été détecté mais son driver n'est pas correctement installé ou est inexistant. Pour résoudre ce problème il suffit de sélectionner les propriétés du périphérique, onglet pilote, choisir « mettre à jour le pilote » et recommencer les étapes de l'installation décrites auparavant.

- Installation interrompue :

Si au cours de l'installation un problème informatique survient, veuillez recommencer toutes les étapes de l'installation, et se référer aux autres cas de résolution de problèmes pour trouver une issue.

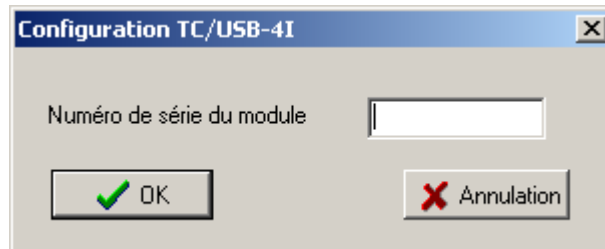
Une autre solution consiste à redémarrer le PC sans aucun module branché, puis de le relancer après avoir branché le module : Ce cycle de redémarrage force Windows à re-détecter les nouveaux périphériques, permettant à l'utilisateur de resaisir l'emplacement du pilote.

- Désinstallation du driver :

Si vous tenez à supprimer le driver , il suffit de rentrer dans le menu « ajout/suppression de programmes » et chercher la ligne :TC/USB-4I USB Drivers. Suivre la procédure indiquée par le logiciel.

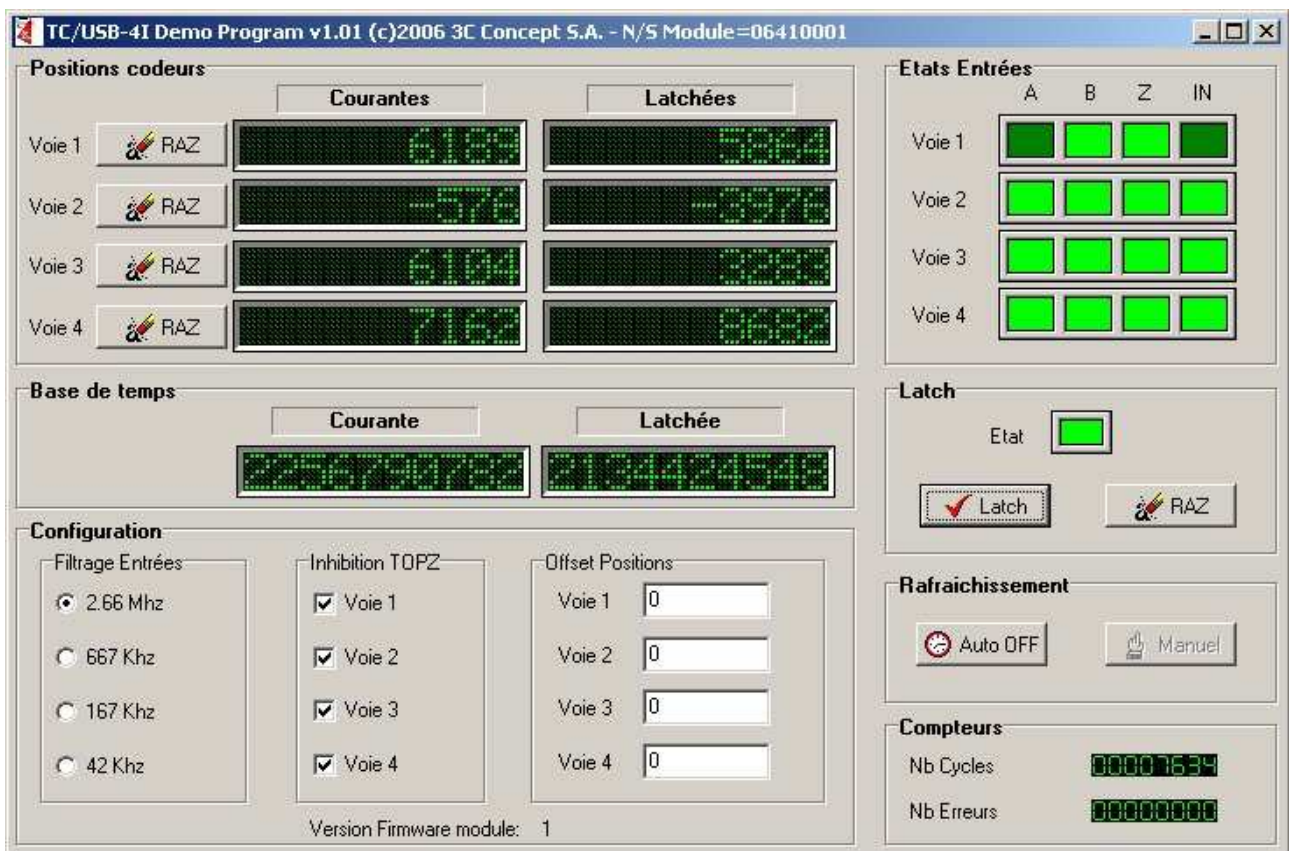
Programme de démonstration

Le programme de démonstration est présent sur le CD-ROM livré avec le module USB, il se nomme « TCUSB4I\Win32\Programme de demonstartion\DemoTCUSB4I.exe».



Lors du lancement du programme de démonstration, celui-ci demande le numéro de série du module TC/USB-4I. Si vous n'utilisez qu'un seul module, vous n'êtes pas obligé de rentrer son numéro de série, il suffit de cliquer sur OK et le programme détecte le premier module branché.

NB : Veillez à ce que le module **soit alimenté par son alimentation principale**, le numéro de série est affiché au dos du module



Le programme de démonstration fait appel à toutes les fonctions présentes dans la DLL fournie avec le CD-ROM.

Position codeurs :

Cette zone nous informe sur les données issues du couple module / codeurs : La position courante de chaque voie, ainsi que la position latchée. A noter qu'il est possible de remettre à zéro les positions courantes dans cette fenêtre.

Etats des entrées :

Cette zone nous donne une image des signaux (A,B,Z) issus des codeurs, sur toutes les voies. La case « IN » nous indique l'état de l'entrée auxiliaire.

Base de Temps :

Cette zone nous informe sur la valeur de la base de temps ainsi que la base de temps latchée.

Latch :

Cette zone nous permet de commander un latch sur les 4 voies ainsi que sur la base de temps. Remarquez que l'on peut remettre à zéro toutes les informations latchées. De plus la case « Etat » nous indique si une opération de latch a été effectuée.

Configuration :

Cette zone permet de paramétrer le fonctionnement du module : Choisir la fréquence du filtrage d'entrée, l'inhibition du TopZ (la remise à zéro des positions en fonctions du Top 0), ainsi que de définir une valeur d'offset sur les positions

Rafraîchissement :

En fonctionnement normal le programme de démonstration interroge le module en permanence , le bouton « Auto OFF » arrête le rafraîchissement automatique, et permet de rafraîchir manuellement les données issues du couple module/codeurs grâce au bouton « Manuel ».

Compteurs :

Cette zone permet de vérifier qu'il n'y a eu aucune erreur lors des échanges de données entre le module et le PC. Indique le nombre de cycle de lecture effectué.

la ligne « version Firmware module » permet de connaître la version du logiciel embarqué dans le module.

Utilisation de la DLL TC/USB-4I

Une DLL destinée à l'utilisation du module dans un environnement Windows 32 bits (2000/XP) est fournie avec le module. Cette DLL permet à l'utilisateur d'appeler les procédures et fonctions de gestion du module depuis un langage quelconque supportant l'appel aux procédures externes.

Pour un compilateur PASCAL (DELPHI):

Utilisez le fichier TCUSB4I.INC et la DLL TCUSB4I.DLL

Pour un compilateur C :

Utilisez le fichier TCUSB4I.H et la DLL TCUSB4I.DLL

Pour le langage Basic:

Utilisez le fichier TCUSB4I.BAS et la DLL TCUSB4I.DLL

Shared object sous Linux:

Voir dans le répertoire « linux » du disque fourni, un fichier texte explique la démarche à suivre.

(TCUSB4I\Linux\librairie)

ⓘ Attention : Toutes les fonctions ci-dessous travaillent sur une image mémoire actualisée par la fonction TCUSB4I_Refresh. Cette fonction doit être appelée en tête de toute fonction.

Description des fonctions :

TCUSB4I_OpenN(Nmodule, NumSerie)

Paramètre : Nmodule: entier signé sur 32 bits.

Numserie : Pointeur sur un Dword non signé sur 32 bits contenant le numéro de série.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Initialise le driver, affecte le numéro de série du module au numéro de module spécifié (1 à 8) et remet à 0 les sorties. si le numéro de série=0 alors elle ouvre le premier module détecté et renvoi son numéro de série.

A utiliser avant tout autre appel à une fonction TCUSB4I_XXX.

Fermer la session par TCUSB4I_Close .

TCUSB4I_Open(NModule,NumSerie)

Paramètre : Nmodule: entier signé sur 32 bits.

Numserie : Pointeur sur une chaîne de caractère contenant le numéro de série.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Initialise le driver, affecte le numéro de série du module au numéro de module spécifié (1 à 8) et remet à 0 les sorties. si le numéro de série est vide alors elle ouvre le premier module détecté et renvoi son numéro de série.

A utiliser avant tout autre appel à une fonction TCUSB4I_XXX.

Fermer la session par TCUSB4I_Close .

L'utilisation des pointeurs sur les chaînes de caractères n'étant pas toujours aisée à mettre en œuvre en fonction du système de développement, il est préférable d'utiliser la fonction TCUSB4I_OpenN

TCUSB4I_Close(NModule)

Paramètre : Nmodule: entier signé sur 32 bits.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Termine la session et ferme le driver (indispensable à la fin du programme)

TCUSB4I_Refresh(NModule)

Paramètre : Nmodule: entier signé sur 32 bits.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Actualise les données provenant du module (positions courantes, positions latchées, base de temps, base de temps latchée, états des entrées, états des lignes A/B/Z de chaque voie) du module indiqué

A utiliser avant tout appel aux fonctions de lecture des données

TCUSB4I_Pos(Nmodule,Voie,Data) (1)

Paramètre : Nmodule: entier signé sur 32 bits.

Voie: entier signé sur 32 bits.

Data : Pointeur sur un entier signé sur 32 bits.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Renvoie dans Data, la position courante de la voie du module indiqué

TCUSB4I_PosLatch(Nmodule,Voie,Data) (1)

Paramètre : Nmodule: entier signé sur 32 bits.

Voie: entier signé sur 32 bits.

Data : Pointeur sur un entier signé sur 32 bits.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Renvoie dans Data, la position mémorisée de la voie du module indiqué

TCUSB4I_Time(Nmodule,Data) (1)

Paramètre : Nmodule: entier signé sur 32 bits.

Data : Pointeur sur un Dword non signé sur 32 bits.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Renvoie dans Data, la valeur courante de la base de temps du module indiqué (Rappel : base de temps cadencée @3.2MHz)

TCUSB4I_TimeLatch(Nmodule,Data) (1)

Paramètre : Nmodule: entier signé sur 32 bits.

Data : Pointeur sur un Dword non signé sur 32 bits.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Renvoie dans Data, la valeur mémorisée de la base de temps du module indiqué

TCUSB4I_InputLE(Nmodule,Data) (1)

Paramètre : Nmodule: entier signé sur 32 bits.

Data : Pointeur sur un byte non signé sur 8 bits.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Renvoie dans Data, l'état du latch du module indiquée (0 pas de mémorisation; 1 mémorisation)

TCUSB4I_Status(Nmodule,Voie,Data) (1)

Paramètre : Nmodule: entier signé sur 32 bits.

Voie: entier signé sur 32 bits.

Data : Pointeur sur un byte non signé sur 8 bits.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Renvoi dans Data, l'état des entrées de la voie indiquée (A,B,Z,IN)

TCUSB4I_Version(Nmodule,Data1,Data2) (1)

Paramètre : Nmodule: entier signé sur 32 bits.

Data1 : Pointeur sur un byte non signé sur 8 bits.

Data2 : Pointeur sur un byte non signé sur 8 bits.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Renvoi dans Data1, la version du firmware du module indiqué.

Renvoi dans Data2, la version de la DLL utilisée par le programme.

TCUSB4I_GetConfig(Nmodule,Data) (1)

Paramètre : Nmodule: entier signé sur 32 bits.

Data : Pointeur sur un byte non signé sur 8 bits.

Retour : la fonction renvoi un entier signé sur 32 bits correspondant au code du défaut (voir liste des défauts ci dessous).

Renvoi dans Data, la configuration du module (Filtrage, Inhibition)

TCUSB4I_Latch(NModule)

Paramètre : Nmodule: entier signé sur 32 bits.

Retour : aucun.

Commande de mémorisation (latch) du module indiqué

TCUSB4I_RAZLatch(NModule)

Paramètre : Nmodule: entier signé sur 32 bits.

Retour : aucun.

Commande de mise à zéro de la mémorisation (latch) du module indiqué

TCUSB4I_RAZPos(Nmodule,Data)

Paramètre : Nmodule: entier signé sur 32 bits.

Data : byte non signé sur 8 bits.

Retour : aucun.

Commande de mise à zéro des positions courantes en fonction du paramètre Data (Data=1 RAZ de la Voie 1, Data=2 RAZ de la Voie 2, Data=4 RAZ de la Voie 3, Data=8 RAZ de la Voie 4 , Data=15 RAZ de toutes les voies)

TCUSB4I_setREF(Nmodule,Data)

Paramètre : Nmodule: entier signé sur 32 bits.

Data : byte non signé sur 8 bits.

Retour : aucun.

Commande d'inhibition du signal Z en fonction du paramètre Data (Data=1 inhibition du signal Z sur Voie 1, Data=2 inhibition du signal Z sur Voie 2, Data=4 inhibition du signal Z sur Voie 3, Data=8 inhibition du signal Z sur Voie 4 , Data=15 inhibition du signal Z sur toutes les voies)

TCUSB4I_FreqNum(Nmodule,Data)

Paramètre : Nmodule: entier signé sur 32 bits.
Data : byte non signé sur 8 bits.
Retour : aucun.

Sélection de la fréquence du filtre numérique d'entrée en fonction du paramètre Data (Data=1 → 2.67MHz, Data=2 → 667kHz, Data=4 → 167kHz, Data=8 → 42kHz)

TCUSB4I_SetPos(Nmodule,Voie,Data)

Paramètre : Nmodule: entier signé sur 32 bits.
Voie: entier signé sur 32 bits.
Data : entier signé sur 32 bits.
Retour : aucun.

Commande d'Offset sur la position de la voie indiquée en fonction du paramètre Data (Data=valeur de l'offset)

TCUSB4I_ResetPort(NModule)

Paramètre : Nmodule: entier signé sur 32 bits.
Retour : aucun.

Réinitialise le port USB du module : en cas de perte de communication avec le module, cette commande tente de reprendre le dialogue. (Cette fonction n'est pas équivalente au fait de débrancher/ re-brancher le module)

TCUSB4I_CyclePort(NModule)

Paramètre : Nmodule: entier signé sur 32 bits.
Retour : aucun.

Déconnecte et reconnecte le port USB du module. Cette fonction est équivalente au fait de débrancher/ re-brancher physiquement le module. Cette commande peut être utilisée suite à une erreur sérieuse, pour tenter de rétablir la communication module/PC.

(1) Voir la mise en garde concernant l'usage de la fonction TCUSB4I_Refresh ci-dessus

Liste des défauts:

N°défaut	Désignation
0	Pas d'erreur
1	Numéro de série incorrect
2	N° de module incorrect
3	Pas de module TC/USB-4I détecté
4	Module non initialisé
5	N° de voie incorrect
6	Erreur écriture
7	Erreur lecture
8	Erreur données reçues
9	Erreur à la fermeture
10	Erreur octet fin de trame

Problème sur l'ouverture du module avec la fonction TCUSB4I_Open

L'appel à la fonction TCUSB4I_Open se fait en passant comme paramètre un numéro de module (de 1 à 8) et un numéro de série,

Le numéro de série est passé en tant que pointeur sur la chaîne de caractère de type zéro terminal contenant le numéro de série, Cette chaîne peut être vide lors de l'appel à TCUSB4I_Open auquel cas, le premier module trouvé sera ouvert et la fonction retournera le numéro de série du module trouvé.

Sous ATEasy, lors de la déclaration de la procédure TCUSB4I_Open, la chaîne contenant le numéro de série doit être de type String parameter Var avec une dimension de 1. Lors de l'appel à cette procédure dans le programme la variable doit être de type String parameter None avec une dimension de 1.