



Ideazione Progettazione Sviluppo Elettronica Scientifica

LIBRERIA DINAMICA

FTD2XX.DLL

MANUALE D'USO



Pag. 1

Le informazioni contenute nel presente manuale sono proprietà di IPSES S.r.l. e devono essere considerate e trattate come confidenziali.

La presente pubblicazione può essere riprodotta, trasmessa, trascritta o tradotta in qualsiasi linguaggio umano o elettronico solamente dopo avere ottenuto l'autorizzazione scritta di IPSES S.r.l.

Le informazioni contenute nel presente manuale sono state accuratamente verificate e sono considerate valide alla data di pubblicazione del presente documento; IPSES non si assume comunque alcuna responsabilità di eventuali inaccurately. IPSES non potrà essere ritenuta responsabile per qualsiasi danno conseguente o incidentale accaduto dall'aver considerato accurata la presente documentazione.

Le informazioni contenute nel presente manuale possono subire variazioni senza preavviso e non rappresentano un impegno da parte di IPSES.

Il presente manuale in italiano è la versione originale.

Stampato in Italia

Copyright © 2007-10 IPSES S.r.l.

Tutti i diritti riservati.



INDICE

Revisioni Manuale	5
Architettura Driver D2XX	6
Installazione del driver D2XX.....	7
Variabili	10
Errori di stato.....	10
Costanti.....	10
Funzioni della DLL	12
FT_ListDevices.....	12
FT_Open.....	13
FT_OpenEx	14
FT_Close.....	15
FT_Read	15
FT_Write.....	16
FT_ResetDevice	16
FT_SetBaudRate	17
FT_SetDataCharacteristics	17
FT_SetFlowControl.....	18
FT_GetModemStatus.....	19
FT_Purge	19
FT_SetTimeouts	20
FT_GetQueueStatus	20
FT_GetStatus	20
FT_GetDeviceInfo	21
FT_ResetPort	22
FT_CreateDeviceInfoList.....	22
FT_GetDeviceInfoList	23
FT_GetDeviceInfoDetail	23
FT_GetDriverVersion.....	24



FT_GetLibraryVersion	25
FT_SetBitMode.....	25
FT_GetBitMode	26
CONTATTI	27
INFORMAZIONI PER IL SUPPORTO TECNICO	28
RAPPORTO PROBLEMATICHE	28
ENGINEERING PROBLEM REPORT.....	29



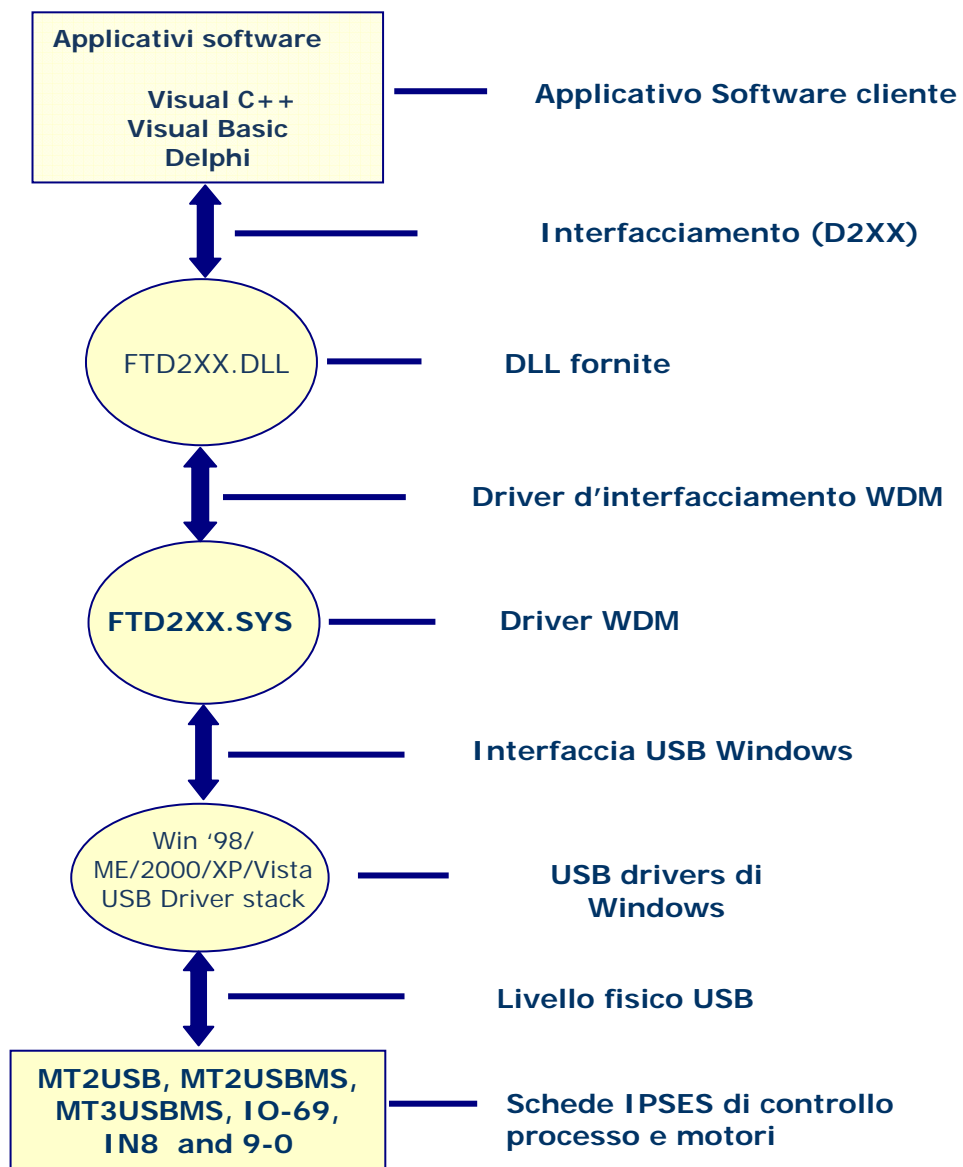
Revisioni Manuale

Revisione/ Data	Descrizione modifica	Autore
01.00.0000 Luglio, 2008	Rilascio prima versione	Pizzocolo G.
01.01.0000 Aprile, 2009	Aggiunta descrizione funzioni "FT_SetBitMode" e "FT_GetBitMode". Altre modifiche minori.	Rivolta A.
01.02.0000 Agosto, 2010	Aggiunto capitolo "Installazione del driver D2XX"	Zancanato A.



Architettura Driver D2XX

La libreria dinamica per Windows FTD2XX.DLL permette la scrittura di applicativi software per l'interfacciamento con i dispositivi elettronici realizzati da IPSES S.r.l. L'architettura della FTD2XX.DLL è composta da un driver WDM di Windows, che comunica con il dispositivo attraverso USB, e da DLL, che interfaccia il software applicativo (scritto in VC++, C++ Builder, Delphi, VB etc.) con il driver WDM. La libreria FTD2XX.DLL contiene un insieme di funzioni elementari, di semplice utilizzo, per il dialogo con le schede **MT2USB, MT2USBMS, MT3USBMS, IO-69, IN8 e 9-0**.



Installazione del driver D2XX

Si consiglia di avviare l'installazione del driver D2XX dal collegamento disponibile sul CD fornito da IPSES Srl con il dispositivo.

In questo modo l'installazione sarà completamente automatizzata e il dispositivo sarà riconosciuto automaticamente quando collegato.

Non sarà quindi necessario seguire le procedure descritte in questo capitolo.

Il *driver* USB D2XX fornito da IPSES Srl è certificato per i più recenti sistemi operativi Microsoft:

- Microsoft Windows 2000 family
- Microsoft Windows XP family, x86
- Microsoft Windows Server 2003 family, x86
- Microsoft Windows Server 2003 family, x64
- Microsoft Windows XP family, x64
- Microsoft Windows Vista family, x86
- Microsoft Windows Vista family, x64
- Windows Server 2008 family, x86
- Windows Server 2008 family, x64
- Windows 7
- Windows 7 x64
- Windows Server 2008 Release 2 family, x64

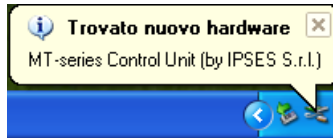


Se NON si esegue la procedura di installazione automatica consigliata da CD, e il PC è connesso ad internet è possibile seguire la procedura di installazione automatica con *Windows Update*, altrimenti è necessario procedere con l'installazione manuale.

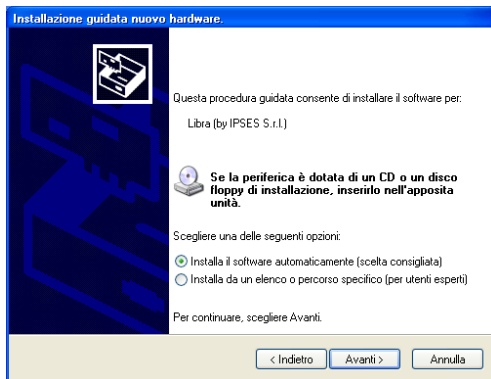
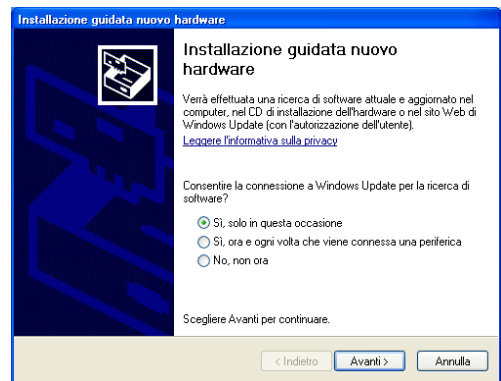


Procedura automatica con Windows Update

- 1) Collegare con il cavo USB il dispositivo IPSES al PC. Il sistema operativo *Windows* rileva la presenza di un dispositivo con un messaggio simile a questo:

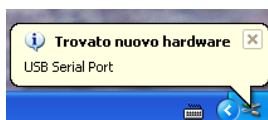
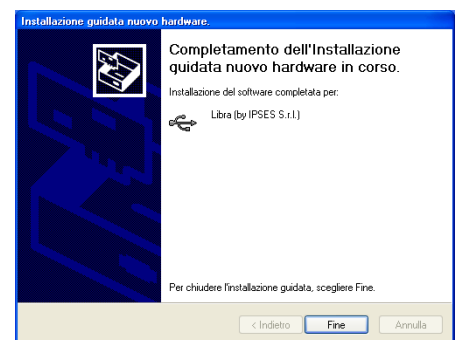


- 2) Nella successiva finestra "Installazione guidata nuovo hardware" scegliere "Sì, solo in questa occasione" e quindi "Avanti".



- 3) Successivamente, scegliere "Installa il software automaticamente (Scelta Consigliata)" e "Avanti". Quindi Attendere il termine della ricerca e dell'installazione.

- 4) L'avvenuta installazione è segnalata dal messaggio di completamento dell'aggiornamento guidato *hardware* in corso. Per terminare, scegliere "Fine".

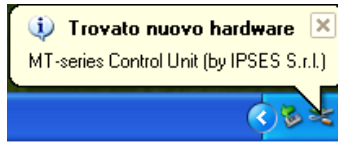


- 5) Terminata l'installazione dell'*hardware* descritta sopra, viene rilevata la nuova periferica "USB Serial Port". Ripercorrere di nuovo quanto sopra dal punto 2).

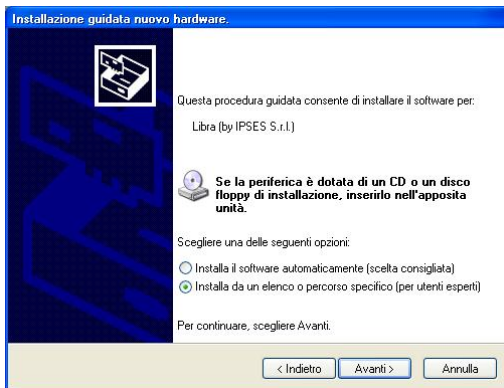
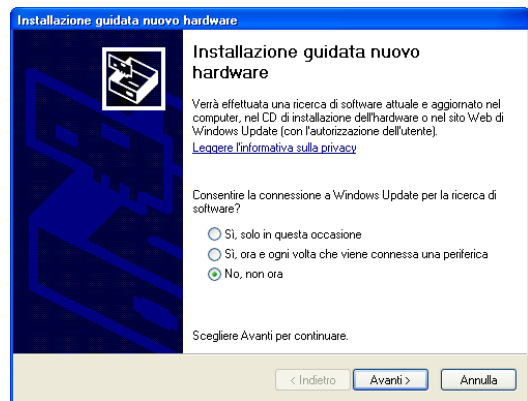


Procedura Installazione driver manuale

- 1) Collegare con il cavo USB la il dispositivo IPSES al PC. Il sistema operativo *Windows* rileva la presenza di un dispositivo con un messaggio simile a questo:



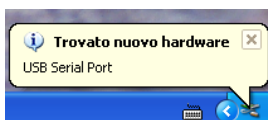
- 2) Nella successiva finestra "Installazione guidata nuovo hardware" scegliere "No, non ora" e quindi "Avanti".



- 3) Successivamente, scegliere "Installa da un elenco o percorso specifico (per utenti esperti)" e "Avanti". Quindi selezionare la cartella "driver" dal CD fornito da IPSES.



- 4) L'avvenuta installazione è segnalata dal messaggio di completamento dell'aggiornamento guidato *hardware* in corso. Per terminare, scegliere "Fine".



- 5) Terminata l'installazione dell'*hardware* descritta sopra, viene rilevata la nuova periferica "USB Serial Port". Ripercorrere di nuovo quanto sopra dal punto 2).



Variabili

UCHAR unsigned char (1 byte).
PUCHAR pointer to unsigned char (4 bytes).
PCHAR pointer to char (1 byte).
DWORD unsigned long (4 bytes).
FT_HANDLE **DWORD**.

Errori di stato

FT_STATUS (DWORD)

FT_OK	0
FT_INVALID_HANDLE	1
FT_DEVICE_NOT_FOUND	2
FT_DEVICE_NOT_OPENED	3
FT_IO_ERROR	4
FT_INSUFFICIENT_RESOURCES	5
FT_INVALID_PARAMETER	6
FT_INVALID_BAUD_RATE	7
FT_DEVICE_NOT_OPENED_FOR_ERASE	8
FT_DEVICE_NOT_OPENED_FOR_WRITE	9
FT_FAILED_TO_WRITE_DEVICE	10
FT_EEPROM_READ_FAILED	11
FT_EEPROM_WRITE_FAILED	12
FT_EEPROM_ERASE_FAILED	13
FT_EEPROM_NOT_PRESENT	14
FT_EEPROM_NOT_PROGRAMMED	15
FT_INVALID_ARGS	16
FT_OTHER_ERROR	17

Costanti

```
FT_DEVICE_LIST_INFO_NODE: typedef struct _ft_device_list_info_node
{
    DWORD      Flags;
    DWORD      Type;
    DWORD      ID;
    DWORD      LocID;
    char       SerialNumber[16];
    char       Description[64];
    FT_HANDLE  ftHandle;
} FT_DEVICE_LIST_INFO_NODE;
```



FTD2XX.DLL – MANUALE D'USO

FT_LIST_NUMBER_ONLY	0x80000000
FT_LIST_BY_INDEX	0x40000000
FT_LIST_ALL	0x20000000
FT_OPEN_BY_SERIAL_NUMBER	1
FT_OPEN_BY_DESCRIPTION	2
FT_OPEN_BY_LOCATION	4
FT_BITS_8	8
FT_BITS_7	7
FT_STOP_BIT_1	0
FT_STOP_BIT_2	2
FT_PARITY_NONE	0
FT_PARITY_ODD	1
FT_PARITY_EVEN	2
FT_PARITY_MARK	3
FT_PARITY_SPACE	4
FT_FLOW_NONE	0x0000
FT_FLOW_RTS_CTS	0x0100
FT_FLOW_DTR_DSR	0x0200
FT_FLOW_XON_XOFF	0x0400
FT_PURGE_RX	1
FT_PURGE_TX	2



Funzioni della DLL

FT_ListDevices

Descrizione

Ottiene informazioni riguardo i dispositivi contemporaneamente connessi. Questa funzione può informare circa il numero di schede, il loro serial number, la stringa di descrizione e l'IDs dei dispositivi connessi.

Sintassi

FT_STATUS **FT_ListDevices** (PVOID *pvArg1*, PVOID *pvArg2*, DWORD *dwFlags*)

Parametri

<i>pvArg1</i>	dipende dal valore di <i>dwFlags</i> (vedi nota)
<i>pvArg2</i>	dipende dal valore di <i>dwFlags</i> (vedi nota)
<i>dwFlags</i>	determina il formato e i parametri restituiti dalla funzione (vedi nota)

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota

Questa funzione può essere usata in diverse configurazioni per ottenere differenti tipi di informazioni.

Un metodo più potente per ottenere tutte le informazioni disponibili sul dispositivo consiste nell'utilizzare le funzioni **FT_CreateDeviceInfoList**, **FT_GetDeviceInfoList** e **FT_GetDeviceInfoDetail**.

Nella sua forma più semplice, questa funzione può essere usata per ottenere il numero di schede contemporaneamente connesse. Se il parametro *dwFlags* è configurato come *FT_LIST_NUMBER_ONLY*, il parametro *pvArg1* è interpretato come un puntatore a una variabile di tipo *DWORD* nella quale verrà memorizzato il numero di dispositivi contemporaneamente connessi.

Questa funzione può essere usata anche per ottenere informazioni sui serial number o sulle descrizioni delle schede connesse.

Se il parametro *dwFlags* è configurato come *FT_OPEN_BY_SERIAL_NUMBER*, verrà restituita la stringa contenente il *serial number*; se *dwFlags* è configurato come *FT_OPEN_BY_DESCRIPTION*, verrà restituita la stringa della descrizione del prodotto; se *dwFlags* è configurato come *FT_OPEN_BY_LOCATION*, verrà restituito l'ID di locazione. Se non è configurato nessuno di questi valori, di default verrà restituita la stringa dei serial number.

FT_ListDevices può essere usata per ottenere informazioni relative ad singolo dispositivo.

In questo caso se *dwFlags* è configurato come *FT_LIST_BY_INDEX* in AND logico con *FT_OPEN_BY_SERIAL_NUMBER* o *FT_OPEN_BY_DESCRIPTION*, il parametro *pvArg1* è interpretato come indice del dispositivo e *pvArg2* come puntatore ad un buffer di dimensioni adatte a memorizzare la stringa restituita. L'indicizzazione è di tipo zero-based (comincia da 0): se viene passato un indirizzo non valido la funzione genera l'errore di stato *FT_DEVICE_NOT_FOUND*.



FT_ListDevices può essere usata per ottenere informazioni relative a tutti i dispositivi connessi.

In quest'altro caso se *dwFlags* è configurato come *FT_LIST_ALL* in AND logico con *FT_OPEN_BY_SERIAL_NUMBER* o *FT_OPEN_BY_DESCRIPTOR*, il parametro *pvArg1* è interpretato come puntatore ad un array di puntatori ai buffer, di dimensioni adatte a memorizzare la stringa restituita da ciascun dispositivo, mentre il parametro *pvArg2* è interpretato come puntatore ad una variabile di tipo *DWORD* in cui viene memorizzato il numero di dispositivi contemporaneamente connessi.

L'array di puntatori ai buffer, *pvArg1*, deve terminare con un valore *NULL*, quindi il numero di elementi dell'array deve essere maggiore di una unità rispetto al numero di dispositivi contemporaneamente connessi.

Se *dwFlags* è configurato come *FT_LIST_BY_INDEX* in AND logico con *FT_OPEN_BY_LOCATION* la funzione restituisce l'ID di locazione di un singolo dispositivo. Il parametro *pvArg1* è interpretato come indice del dispositivo mentre *pvArg2* come puntatore ad una variabile di tipo *long* nella quale verrà memorizzato l'ID di locazione. L'indicizzazione è di tipo zero-based (comincia da 0): se viene passato un indirizzo non valido la funzione genera l'errore di stato *FT_DEVICE_NOT_FOUND*.

Se *dwFlags* è configurato come *FT_LIST_ALL* in AND logico con *FT_OPEN_BY_LOCATION* la funzione restituisce l'ID di locazione di tutti i dispositivi contemporaneamente connessi. In questo caso il parametro *pvArg1* è interpretato come un puntatore ad un array di variabili di tipo *long* nelle quali verranno memorizzati gli ID di locazione, mentre *pvArg2* è interpretato come un puntatore a una variabile di tipo *DWORD* nella quale verrà memorizzato il numero di dispositivi connessi.

FT_Open

Descrizione

Apri la comunicazione con il dispositivo e restituisce un *Handle* che verrà usato per i seguenti accessi.

Sintassi

FT_STATUS **FT_Open** (int *iDevice*, FT_HANDLE **ftHandle*)

Parametri

<i>iDevice</i>	Indice del dispositivo da connettere. Gli indici sono di tipo <i>zero-based</i> .
<i>ftHandle</i>	Puntatore ad una variabile di tipo FT_HANDLE nella quale viene memorizzato l' <i>handle</i> della connessione. Questo <i>handle</i> deve essere usato per i successivi accessi al dispositivo.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota

Sebbene questa funzione possa essere usata per aprire connessioni multiple settando *iDevice* 0, 1, 2 ecc., essa non è in grado di aprire la connessione con uno specifico dispositivo. Per aprire una connessione con uno specifico dispositivo, usare la funzione



FT_OpenEx. Grazie a **FT_OpenEx** è possibile stabilire la connessione con un dispositivo tramite il suo *serial number* o la sua descrizione.

FT_OpenEx

Descrizione

Aprire la comunicazione con il dispositivo specificato e restituisce un handle che dovrà essere usato per i seguenti accessi. Il dispositivo può essere specificato tramite il suo *serial number*, la sua descrizione (*device description*) o la sua locazione (*ID location*). Questa funzione può anche essere utilizzata per aprire più connessioni simultaneamente. Anche in questo caso i dispositivi possono essere specificati tramite i loro *serial number*, i descrittori (*device descriptor*) o ID di locazione (gli ID di locazione derivano dalla posizione fisica di un dispositivo sul bus USB). Gli ID di locazione per i dispositivi connessi a un sistema possono essere ottenuti invocando le funzioni **FT_GetDeviceInfoList** o **FT_ListDevices** con gli opportuni flags configurati.

Sintassi

FT_STATUS **FT_OpenEx** (PVOID *pvArg1*, DWORD *dwFlags*, FT_HANDLE **ftHandle*)

Parametri

<i>pvArg1</i>	Puntatore ad un argomento il cui tipo dipende dalla configurazione del parametro <i>dwFlags</i> . Normalmente è interpretato come un puntatore ad una stringa terminata con un elemento NULL.
<i>dwFlags</i>	<i>FT_OPEN_BY_SERIAL_NUMBER</i> , <i>FT_OPEN_BY_DESCRIPTION</i> o <i>FT_OPEN_BY_LOCATION</i> .
<i>ftHandle</i>	Puntatore ad una variabile di tipo FT_HANDLE nella quale viene memorizzato l' <i>handle</i> della connessione. Questo <i>handle</i> deve essere usato per i successivi accessi al dispositivo.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota

Il tipo del parametro *pvArg1* è funzione di *dwFlags*: se *dwFlags* è configurato come *FT_OPEN_BY_SERIAL_NUMBER*, *pvArg1* deve essere un puntatore ad una stringa (null-terminated) che contiene il *serial number* del dispositivo; se *dwFlags* è configurato come *FT_OPEN_BY_DESCRIPTION*, *pvArg1* deve essere un puntatore ad una stringa (null-terminated) che contiene la descrizione del dispositivo; se *dwFlags* è configurato come *FT_OPEN_BY_LOCATION*, *pvArg1* deve essere una variabile di tipo long che contiene la locazione ID del dispositivo.

Il parametro *ftHandle* è un puntatore ad una variabile di tipo FT_HANDLE che memorizza l'*handle* di comunicazione con il dispositivo. Questo *handle* dovrà essere utilizzato per i futuri accessi alla scheda.



FT_Close

Descrizione

Termina la comunicazione con un dispositivo aperto e libera le sue risorse.

Sintassi

FT_STATUS **FT_Close** (FT_HANDLE *ftHandle*)

Parametri

ftHandle Handle del dispositivo.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

FT_Read

Descrizione

Legge una stringa dal dispositivo.

Sintassi

FT_STATUS **FT_Read** (FT_HANDLE *ftHandle*, LPVOID *lpBuffer*, DWORD *dwBytesToRead*, LPDWORD *lpdwBytesReturned*)

Parametri

<i>ftHandle</i>	Handle del dispositivo.
<i>lpBuffer</i>	Puntatore ad un buffer in cui memorizzare i dati ricevuti dal dispositivo.
<i>DwBytesToRead</i>	Numero di bytes da leggere dal dispositivo.
<i>lpdwBytesReturned</i>	Puntatore ad una variabile di tipo DWORD nella quale viene memorizzato il numero di bytes effettivamente letti dal dispositivo.

Valore Restituito

FT_OK se terminata con successo, FT_IO_ERROR in caso di errore.

Nota

FT_Read restituisce sempre il numero di bytes letti, scrivendolo in *lpdwBytesReturned*. Questa funzione non termina la sua esecuzione finché non vengono letti *dwBytesToRead* bytes e vengono posti nel buffer. Il numero di bytes presente nella coda di ricezione può essere determinato chiamando la funzione **FT_GetStatus** o **FT_GetQueueStatus**, e quindi passato come parametro *dwBytesToRead* a **FT_Read** cosicché quest'ultima funzione legga tutta la coda del dispositivo e termini immediatamente.

Se precedentemente è stato impostato un timeout di lettura con la funzione **FT_SetTimeouts**, **FT_Read** terminerà allo scadere di questo tempo o quando *dwBytesToRead* bytes saranno stati letti, dipende da quale delle due condizioni si verifica prima. Se si verifica il timeout, **FT_Read** legge i dati disponibili nel buffer di ricezione e restituisce FT_OK come valore di stato.



Un metodo di controllo per la gestione del buffer dovrebbe usare lo stato restituito dalla funzione e il valore memorizzato in *lpdwBytesReturned*. Se il valore restituito è FT_OK, e *lpdwBytesReturned* è uguale a *dwBytesToRead* allora **FT_Read** è effettivamente terminata con successo. Se il valore restituito è FT_OK, e *lpdwBytesReturned* è minore di *dwBytesToRead* allora si è verificato un timeout e la lettura è stata completata solo in parte. Nota che se si verifica un timeout e nessun dato viene letto il valore restituito è comunque FT_OK.

Lo stato FT_IO_ERROR indica un errore critico dovuto all'assegnazione errata dei parametri della funzione o ad un problema hardware legato alla connessione USB.

FT_Write

Descrizione

Scrivere una stringa nel dispositivo.

Sintassi

FT_STATUS **FT_Write** (FT_HANDLE *ftHandle*, LPVOID *lpBuffer*, DWORD *dwBytesToWrite*, LPDWORD *lpdwBytesWritten*)

Parametri

<i>ftHandle</i>	Handle del dispositivo.
<i>lpBuffer</i>	Puntatore ad un buffer che contiene i dati da scrivere nel dispositivo.
<i>DwBytesToWrite</i>	Numero di bytes da scrivere nel dispositivo.
<i>lpdwBytesWritten</i>	Puntatore ad una variabile di tipo DWORD in cui viene memorizzato il numero di bytes effettivamente scritti nel dispositivo.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

FT_ResetDevice

Descrizione

Questa funzione invia un comando di reset al dispositivo.

Sintassi

FT_STATUS **FT_ResetDevice** (FT_HANDLE *ftHandle*)

Parametri

<i>ftHandle</i>	Handle del dispositivo.
-----------------	-------------------------

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.



FT_SetBaudRate

Descrizione

Questa funzione imposta il *baud rate* per il dispositivo.

Sintassi

FT_STATUS **FT_SetBaudRate** (FT_HANDLE *ftHandle*, DWORD *dwBaudRate*)

Parametri

ftHandle Handle del dispositivo.
dwBaudRate Valore del baud rate.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Per comunicare con i dispositivi forniti da **IPSES** S.r.l è necessario impostare il parametro *dwBaudRate* della funzione **FT_SetBaudRate** come mostrato nella seguente tabella:

Device	<i>dwBaudRate</i>
MT2USB	9600
MT2USBMS	9600
MT3USBMS	9600
IO-69	9600
9-0	19200

FT_SetDataCharacteristics

Descrizione

Questa funzione configura le caratteristiche del pacchetto dati per il dispositivo.

Sintassi

FT_STATUS **FT_SetDataCharacteristics** (FT_HANDLE *ftHandle*, UCHAR *uWordLength*, UCHAR *uStopBits*, UCHAR *uParity*)

Parametri

ftHandle Handle del dispositivo.
uWordLength Numero di bits per parola. Se la word ha dimensione 8 bits deve essere impostato *FT_BITS_8*, se la word ha dimensione 7 bits deve essere impostato *FT_BITS_7*.
uStopBits Numero di *bits* di stop. Se è richiesto un bit di stop deve essere configurato come *FT_STOP_BITS_1*, se sono richiesti due *bits* di stop deve essere configurato come *FT_STOP_BITS_2*.



uParity Numero di *bits* di parità. Può essere configurato come *FT_PARITY_NONE* (nessun *bit* di parità) oppure come *FT_PARITY_ODD* (*bit* di parità dispari) oppure come *FT_PARITY_EVEN* (*bit* di parità pari) oppure come *FT_PARITY_MARK* (*bit* di parità sempre alto) oppure come *FT_PARITY_SPACE* (*bit* di parità sempre basso).

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Per comunicare con i dispositivi forniti da **IPSES** S.r.l è necessario impostare i parametri della funzione **FT_SetDataCharacteristics** come mostrato nella seguente tabella:

Device	<i>uWordLength</i>	<i>uStopBits</i>	<i>uParity</i>
MT2USB	8	1	0
MT2USBMS	8	1	0
MT3USBMS	8	1	0
IO-69	8	1	0
9-0	8	1	0

FT_SetFlowControl

Descrizione

Questa funzione imposta il controllo di flusso dei dati per il dispositivo.

Sintassi

FT_STATUS **FT_SetFlowControl** (FT_HANDLE *ftHandle*, USHORT *usFlowControl*, UCHAR *uXon*, UCHAR *uXoff*)

Parametri

ftHandle Handle del dispositivo.
usFlowControl Configura il tipo di controllo di flusso. Questo può essere impostato come FT_FLOW_NONE (nessun controllo di flusso), FT_FLOW_RTS_CTS (controllo di flusso hardware di tipo RTS/CTS), FT_FLOW_DTR_DSR (controllo di flusso hardware di tipo DTR/DSR) o FT_FLOW_XON_XOFF (controllo di flusso software di tipo XON/XOFF).
uXon Carattere usato come segnale Xon. Usare solo se il controllo di flusso è configurato FT_FLOW_XON_XOFF.
uXoff Carattere usato come segnale Xoff. Usare solo se il controllo di flusso è configurato FT_FLOW_XON_XOFF.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.



FTD2XX.DLL – MANUALE D'USO

Per comunicare con i dispositivi forniti da **IPSES** S.r.l è necessario impostare i parametri della funzione **FT_SetFlowControl** come mostrato nella seguente tabella:

Device	<i>usFlowControl</i>	<i>uXon</i>	<i>uXoff</i>
MT2USB	NONE	0	0
MT2USBMS	NONE	0	0
MT3USBMS	NONE	0	0
IO-69	NONE	0	0
9-0	NONE	0	0

FT_GetModemStatus

Descrizione

Restituisce lo stato del modem e della linea di comunicazione lato dispositivo.

Sintassi

FT_STATUS **FT_GetModemStatus** (FT_HANDLE *ftHandle*, LPDWORD *lpdwModemStatus*)

Parametri

FtHandle Handle del dispositivo.
lpdwModemStatus Puntatore ad una variabile di tipo DWORD che riceve lo stato del modem dal dispositivo. Gli stati della linea di comunicazione sono mappati come segue:
CTS = 0x10
DSR = 0x20
RI = 0x40
DCD = 0x80

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

FT_Purge

Descrizione

Questa funzione svuota i buffer di ricezione e trasmissione del dispositivo.

Sintassi

FT_STATUS **FT_Purge** (FT_HANDLE *ftHandle*, ULONG *uMask*)

Parametri

FtHandle Handle del dispositivo.
uMask Qualsiasi combinazione di *FT_PURGE_RX* e *FT_PURGE_TX*.



Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

FT_SetTimeouts

Descrizione

Questa funzione imposta i valori di timeout per la lettura e la scrittura del dispositivo.

Sintassi

FT_STATUS **FT_SetTimeouts** (FT_HANDLE *ftHandle*, DWORD *dwReadTimeout*, DWORD *dwWriteTimeout*)

Parametri

<i>ftHandle</i>	Handle del dispositivo.
<i>dwReadTimeout</i>	Timeout di lettura espresso in millisecondi.
<i>dwWriteTimeout</i>	Timeout di scrittura espresso in millisecondi.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

FT_GetQueueStatus

Descrizione

Restituisce il numero di *bytes* della coda di ricezione.

Sintassi

FT_STATUS **FT_GetQueueStatus** (FT_HANDLE *ftHandle*, LPDWORD *lpdwAmountInRxQueue*)

Parametri

<i>ftHandle</i>	Handle del dispositivo.
<i>lpdwAmountInRxQueue</i>	Puntatore ad una variabile di tipo DWORD che memorizza il numero di bytes presenti nella coda di ricezione in attesa di essere letti.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

FT_GetStatus

Descrizione



Restituisce lo stato del dispositivo inclusi il numero di caratteri presenti nella coda di ricezione, il numero di caratteri presenti nella coda di trasmissione e lo stato dell'evento in corso

Sintassi

FT_STATUS **FT_GetStatus** (FT_HANDLE *ftHandle*, LPDWORD *lpdwAmountInRxQueue*, LPDWORD *lpdwAmountInTxQueue*, LPDWORD *lpdwEventstatus*)

Parametri

<i>ftHandle</i>	Handle del dispositivo.
<i>lpdwAmountInRxQueue</i>	Puntatore ad una variabile di tipo DWORD che memorizza il numero di caratteri presenti nella coda di ricezione.
<i>lpdwAmountInTxQueue</i>	Puntatore ad una variabile di tipo DWORD che memorizza il numero di caratteri presenti nella coda di trasmissione.
<i>lpdwEventstatus</i>	Puntatore ad una variabile di tipo DWORD che memorizza lo stato attuale dell'interfaccia.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

FT_GetDeviceInfo

Descrizione

Restituisce le informazioni sul dispositivo se la connessione è aperta.

Sintassi

FT_STATUS **FT_GetDeviceInfo** (FT_HANDLE *ftHandle*, FT_DEVICE **pftType*, LPDWORD *lpdwID*, PCHAR *pcSerialNumber*, PCHAR *pcDescription*, PVOID *pvDummy*)

Parametri

<i>ftHandle</i>	Handle del dispositivo.
<i>pftType</i>	Puntatore ad una variabile di tipo <i>unsigned long</i> che memorizza il tipo di dispositivo.
<i>lpdwId</i>	Puntatore ad una variabile di tipo <i>unsigned long</i> che memorizza l'ID del dispositivo.
<i>pcSerialNumber</i>	Puntatore ad un buffer che memorizza il <i>serial number</i> del dispositivo come una stringa terminata con il carattere NULL.
<i>pcDescription</i>	Puntatore ad un buffer che memorizza la descrizione del dispositivo come una stringa terminata con il carattere NULL.
<i>pvDummy</i>	Riservato per usi futuri – deve essere settato come NULL.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota



Questa funzione è usata per ricavare le informazioni circa il tipo di dispositivo, il suo ID, la sua descrizione e il suo *serial number*. L'ID del dispositivo è codificato come un tipo DWORD – la *word* più significativa contiene il *vendor ID (VID)*, mentre la *word* meno significativa contiene il *product ID (PID)*. Ad esempio, se la funzione restituisce l'ID 0x04036001 questo corrisponde a VID_0403&PID_6001.

FT_ResetPort

Descrizione

Invia un comando di reset alla porta.

Sintassi

FT_STATUS **FT_ResetPort** (FT_HANDLE *ftHandle*)

Parametri

ftHandle Handle del dispositivo.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota

Questa funzione è usata per tentare di ripristinare la porta USB dopo un errore. Non equivale a disconnettere e riconnettere fisicamente il dispositivo.

FT_CreateDeviceInfoList

Descrizione

Questa funzione crea una lista di informazioni di dispositivo e restituisce il numero di dispositivi, basati su *driver* D2XX, connessi al sistema. La lista contiene informazioni sia per i dispositivi con comunicazione aperta, sia per quelli la cui comunicazione non è aperta.

Sintassi

FT_STATUS **FT_CreateDeviceInfoList** (LPDWORD *lpdwNumDevs*)

Parametri

lpdwNumDevs Puntatore ad una variabile di tipo *unsigned long* in cui viene memorizzato il numero di dispositivi connessi.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota

Un'applicazione può sfruttare questa funzione per conoscere il numero di dispositivi connessi al sistema. Successivamente può quindi allocare una dimensione di memoria necessaria a contenere la lista di informazioni per tutti i *devices* e ricavare la stessa



usando le funzioni **FT_GetDeviceInfoList** o **FT_GetDeviceInfoDetail**. Se i dispositivi connessi al sistema cambiano, la lista di informazioni non verrà aggiornata finché **FT_CreateDeviceInfoList** non verrà nuovamente invocata.

FT_GetDeviceInfoList

Descrizione

Questa funzione restituisce una lista di informazioni del dispositivo e il numero di *devices* D2XX che compone la lista.

Sintassi

FT_STATUS **FT_GetDeviceInfo** (FT_DEVICE_LIST_INFO_NODE **pDest*, LPDWORD *lpdwNumDevs*)

Parametri

**pDest* Puntatore ad un array di struttura di tipo FT_DEVICE_LIST_INFO_NODE.
lpdwNumDevs Puntatore al numero di elementi nell'array.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Note

Questa funzione può essere invocata solo dopo aver chiamato **FT_CreateDeviceInfoList**. Se i dispositivi connessi al sistema cambiano, la lista di informazioni non verrà aggiornata finché **FT_CreateDeviceInfoList** non verrà nuovamente invocata.

L'informazione di *location ID* non verrà restituita per i dispositivi che sono già aperti quando **FT_CreateDeviceInfoList** viene invocata.

Le informazioni non sono disponibili per i dispositivi che risultano già aperti in altri processi: in questo caso il parametro *Flags* della **FT_DEVICE_LIST_INFO_NODE** indicherà che il dispositivo risulta aperto ma i restanti campi risulteranno vuoti.

L'array di **FT_DEVICE_LIST_INFO_NODES** contiene tutti i dati disponibili per ogni dispositivo. L'area di memoria per la lista deve essere allocata dall'applicazione: il numero di dispositivi restituito da **FT_CreateDeviceInfoList** può essere usato a questo scopo.

Quando si programma in Visual Basic, LabVIEW o altri linguaggi simili, **FT_GetDeviceInfoDetail** potrebbe essere richiesta al posto di questa funzione.

FT_GetDeviceInfoDetail

Descrizione

Questa funzione restituisce un punto d'ingresso alla lista di informazioni del dispositivo.

Sintassi

FT_STATUS **FT_GetDeviceInfoDetail** (DWORD *dwIndex*, LPDWORD *lpdwFlags*, LPDWORD *lpdwType*, LPDWORD *lpdwID*, LPDWORD *lpdwLocId*, PCHAR



pcSerialNumber, PCHAR *pcDescription*,
FT_HANDLE **ftHandle*)

Parametri

<i>dwIndex</i>	Indice del punto d'ingresso alla lista di informazioni del dispositivo.
<i>lpdwFlags</i>	Puntatore ad una variabile di tipo <i>unsigned long</i> che contiene il valore del flag.
<i>lpdwType</i>	Puntatore ad una variabile di tipo <i>unsigned long</i> che contiene il <i>type</i> del dispositivo.
<i>lpdwID</i>	Puntatore ad una variabile di tipo <i>unsigned long</i> che contiene l'ID del dispositivo.
<i>lpdwLocId</i>	Puntatore ad una variabile di tipo <i>unsigned long</i> che contiene la <i>location ID</i> .
<i>pcSerialNumber</i>	Puntatore ad un buffer che contiene il <i>serial number</i> come stringa terminata con il carattere NULL.
<i>pcDescription</i>	Puntatore ad un buffer che contiene la descrizione del dispositivo come stringa terminata con il carattere NULL.
<i>*ftHandle</i>	Puntatore ad una variabile di tipo FT_HANDLE che contiene l' <i>handle</i> del dispositivo.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota

Questa funzione deve essere invocata solo dopo la chiamata di **FT_CreateDeviceInfoList**. Se i dispositivi connessi al sistema cambiano, la lista di informazioni non verrà aggiornata finché **FT_CreateDeviceInfoList** non verrà nuovamente invocata.

Il valore dell'indice comincia da zero.

La variabile flag è costituita da 4 *bytes* contenente diversi dati: la dimensione della singola informazione è il *bit*. Il bit 0 (bit meno significativo) di questo numero indica se la porta USB è aperta (1) o chiusa (0). Il bit 1 indica se il dispositivo è enumerato come USB *high-speed* (2) o USB *full-speed*(0). I restanti *bits* (2 - 31) sono riservati.

L'informazione di *location ID* non verrà restituita per i dispositivi che sono già aperti quando **FT_CreateDeviceInfoList** viene invocata.

Usare **FT_CreateDeviceInfoList** per ottenere tutta la *info list* come un array di strutture **FT_DEVICE_LIST_INFO_NODE**.

FT_GetDriverVersion

Descrizione

Questa funzione restituisce il numero della versione del driver D2XX.

Sintassi

FT_STATUS **FT_GetDriverVersion** (FT_HANDLE *ftHandle*, LPDWORD *lpdwDriverVersion*)

Parametri



ftHandle Handle del dispositivo.
lpdwDriverVersion Puntatore alla variabile che memorizza la versione del *driver*.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota

La descrizione della versione è costituita da tre componenti: *major*, *minor* e *build* contenute in un campo di 4 *bytes* (di tipo *unsigned long*). Il byte 0 (meno significativo) individua la *build*, il byte1 individua il *minor*, e il byte 2 il *major*. Il byte 3 è settato a zero. Per esempio, la versione del driver "3.01.02" è rappresentata come 0x00030102. La comunicazione con il dispositivo deve essere aperta prima di invocare questa funzione.

FT_GetLibraryVersion

Descrizione

Questa funzione restituisce il numero della versione della DLL D2XX.

Sintassi

FT_STATUS **FT_GetLibraryVersion** (LPDWORD *lpdwDLLVersion*)

Parametri

lpdwDLLVersion Puntatore alla variabile che contiene il valore della versione della DLL.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota

Il numero della versione è costituito da tre elementi, *major*, *minor* e *build*, contenuti in una variabile a 4 *bytes* (di tipo *insigne long*). Il byte 0 (meno significativo) contiene l'informazione di *build*, il byte 1 il campo *minor*, e il byte 2 il campo *major*. Il byte 3 è settato a zero. Per esempio, la versione del driver "3.01.02" è codificata come 0x00030102. Questa funzione non necessita di un *handle*, quindi può essere chiamata senza che il dispositivo sia stato precedentemente aperto.

FT_SetBitMode

Descrizione

Abilita la modalità *CBUS Bit Bang Mode* sui dispositivi MT2USB e MT2USBMS.

Sintassi

FT_STATUS **FT_SetBitmode** (FT_HANDLE *ftHandle*, UCHAR *ucMask*, UCHAR *ucMode*)

Parametri



<i>ftHandle</i>	<i>Handle</i> del dispositivo.
<i>ucMask</i>	Maschera per la configurazione dei <i>pin</i> . Il <i>nibble</i> superiore stabilisce la direzione dei <i>pins</i> : se il bit vale 0 il <i>pin</i> corrispondente sarà un ingresso, se il bit vale 1 il <i>pin</i> corrispondente sarà un'uscita. Il <i>nibble</i> inferiore controlla lo stato dei pin settati come uscite: alto se il bit è 1, basso se il bit è 0.
<i>ucMode</i>	Modalità di configurazione. Può essere una delle seguenti: 0x00 = <i>Reset</i> 0x20 = <i>CBUS Bit Bang Mode</i>

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota

Questa funzione è disponibile solo per i dispositivi **MT2USB** e **MT2USBMS**.
La modalità CBUS Bit Bang permette di controllare direttamente quattro pin fisici (da C3 a C0) del chip USB montato sulla scheda. Per i dispositivi forniti da **IPSES** S.r.l. sono resi disponibili solamente i pin C3 e C2, mentre i pin C1 e C0 sono riservati. Quindi il parametro *ucMask* dovrà essere configurato come "XX00YY00", dove X varrà 0 (ingresso) o 1 (uscita) e Y individuerà lo stato dei pin. Per maggiori informazioni rivolgersi a **IPSES** S.r.l.

FT_GetBitMode

Descrizione

Restituisce il valore attuale dello stato del *CBUS Bit Bang Mode* dei dispositivi MT2USB e MT2USBMS.

Sintassi

FT_STATUS **FT_GetBitmode** (FT_HANDLE *ftHandle*, PCHAR *pucMode*)

Parametri

<i>ftHandle</i>	<i>Handle</i> del dispositivo.
<i>pucMode</i>	Puntatore ad una variabile di tipo <i>unsigned char</i> in cui viene memorizzato il valore istantaneo dello stato del bus. Il <i>nibble</i> inferiore contiene il valore corrente dello stato dei <i>pins</i> , sia di quelli configurati come ingressi, sia di quelli configurati come uscite.

Valore Restituito

FT_OK se terminata con successo, altrimenti restituisce un codice di errore di stato FT.

Nota

Questa funzione è disponibile solo per i dispositivi **MT2USB** e **MT2USBMS**.
Per maggiori informazioni rivolgersi a **IPSES** S.r.l.



CONTATTI

La **IPSES s.r.l.** si occupa dell'ideazione e della commercializzazione di strumenti elettronici e scientifici. La **progettazione personalizzata** consente di rispondere alle diverse esigenze di chi ricerca sistemi *embedded* dedicati ad applicazioni specifiche.

L'**IPSES** si avvale di uno staff con pluriennale esperienza nel settore. L'aggiornamento continuo e l'evoluzione costante rendono l'**IPSES** un'azienda all'avanguardia, capace di unire il dinamismo di una giovane impresa con la professionalità e l'affidabilità di personale qualificato.

IPSES S.r.l.

Sede operativa e centro di sviluppo:

via Trieste, 48
20020 Cesate (MI)
Italy



tel. +39 02 99068453

fax +39 02 700403170

e-mail: info@ipses.com

http://www.ipses.com



INFORMAZIONI PER IL SUPPORTO TECNICO

I nostri tecnici possono essere contattati ai seguenti recapiti:

Telephone : ++39 02 99068453
Fax : ++39 02 700403170
Email : support@ipses.com

RAPPORTO PROBLEMATICHE

Il modulo nella seguente pagina permette di raccogliere i dati necessari ad una corretta ricerca del problema eventualmente evidenziatosi.



ENGINEERING PROBLEM REPORT

Problem describer

Name		IPSES s.r.l. Via Trieste, 48 Cesate (MI) Italy Fax ++39 02/700403170 e-mail <i>support@ipses.com</i>
Company		
Date	Tel.	

Product

Name	Version	Serial No.
------	---------	------------

Report Type (bug, change request or technical problem)

Major bug	<input type="checkbox"/>	Urgency:	
Minor bug	<input type="checkbox"/>	High	<input type="checkbox"/>
Change request	<input type="checkbox"/>	Medium	<input type="checkbox"/>
Technical problem	<input type="checkbox"/>	Low	<input type="checkbox"/>

Problem Description

Reproduction of Problem

IPSES s.r.l. Action notes

Received by	Date	Report No.	Action
-------------	------	------------	--------



Code FTD2XX.DLL

IPSES S.r.l.
Via Trieste, 48
20020 CESATE (MI) - ITALY
Tel. (+39) 02/99068453
Fax (+39) 02/700403170
e-mail: info@ipses.com
support@ipses.com

