USB DLL in 5000PVR

Document Summary

This document includes the descriptions of the functions in TFDLL.dll which is used with Altair.exe.

☞ Change History ☜

Date	Revision	Author	Description of Change
2003-05-28	1.0	Mutul	
2004-07-09	1.1	Mutul	UsbTaskFileSendPosi and UsbTaskFileReceivePosi are added.

☞ Table of Contents ☜

1. USB Function and Specification

1.1. Function in loader

TF5000 users can download the firmware through USB2.0 with TFDN_USB.exe

1.2. Functions in firmware

Sending / Receiving File .

2. Structure of communication block

This chapter explains the structure of communication block Host(PC) and Device(TF5000PVR) through USB2.0.

Device always waits for the host command and communicates with the host by block unit.

Offset	Field	Size	Description
0	Length	2	Total size of block(including length field)
2	CRC16	2	CRC16 of Cmd and Data in block
4	Cmd	4	Command (refer to command table)
8	Data	Length – 8	Contents of data

2.1. Basic structure of block

Maximum size of block is 0xffff

2.2. Request, Reply Cmd Table

Command and data requested by Host(PC)

Reply Cmd means reply for requested command.

Filename : Full path + name

TypeFile : Refer to TypeFile structure

Cmd	Value	Data	Size (Byte)	Reply Cmd	Description
USB_Fail	0x00000001	ErrorCode	4		Fail Refer to ErrorCode
USB_Success	0x00000002				Success(Complete)
USB_Cancel	0x00000003			USB_Fail USB_Success	Host requests job that is being processed. It means reset.
USB_CmdReady	0x00000100			USB_Fail USB_Success	Host requests that device should be ready to communicate.
USB_CmdReset	0x00000101			USB_Fail USB_Success	Hosts requests to reboot TF5000PVR.
USB_CmdTurbo	0x00000102	Mode	4	USB_Fail USB_Success	It makes for TF5000PVR to run Usb Task only. Turbo On : Mode > 0 Turbo Off : Mode = 0 Host has to send Turbo Off command after transfer is

					completed.
USB_CmdHddSize	0x00001000			USB_DataHddSize	Host requests HDD information.
USB_DataHddSize	0x00001001	TotalSize(KB) FreeSize(KB)	4 4		Device sends HDD information.
USB_CmdHddDir	0x00001002	Dir path		USB_DataHddDir	Host requests the full path information of directory.
USB_DataHddDir	0x00001003	File count TypeFile0 TypeFile1 .....2 . .	2	USB_Fail USB_Success	Device sends the full path information of directory. If it is not enough for one block, use USB_RE_HDD_DirEnd
USB_DataHddDirEnd	0x00001004				Device tells that directory path information is sent completely.
USB_CmdHddDel	0x00001005	Filename		USB_Fail USB_Success	Host requests to delete a file or a directory.
USB_CmdHddRename	0x00001006	namesize Filename(source) namesize Filename(target)	2 2	USB_Fail USB_Success	Host request to rename a file or a directory.
USB_CmdHddCreateDir	0x00001007	namesize Filename	2	USB_Fail USB_Success	Host requests to create directory(folder).
USB_CmdHddFileSend	0x00001008	Dir(direction) namesize Filename	1 2	USB_DataHddFileStart USB_Fail USB_Success	Host requests to send a file to a device. Dir means as follows. 0 : From Host to Device 1: From Device to Host
USB_DataHddFileStart	0x00001009	TypeFile		USB_Fail USB_Success	File information to be transferred. Please refer to the below procedure. 1.USB_DataHddFileStart 2.USB_DataHddFileData 3.USB_DataHddFileEnd
USB_DataHddFileData	0x0000100A	HighOffset LowOffset Data	4 4	USB_Fail USB_Success	Position and data of the file
USB_DataHddFileEnd	0x0000100B			USB_Fail USB_Success	File transfer is completed.

2.3. TypeFile structure

Offset	size(byte)	Name	Description
0	2	mjd	Date information of MJD type
2	1	hour	Hour
3	1	min	Minute
4	1	sec	Second
5	1	attr	Type of file 1 : Folder 2 : Normal file
6	4	high size	Size of file : Upper 4byte
10	4	low size	Size of file : Lower 4byte
14	95	name	Filename
109	1	reserved	NotUsed

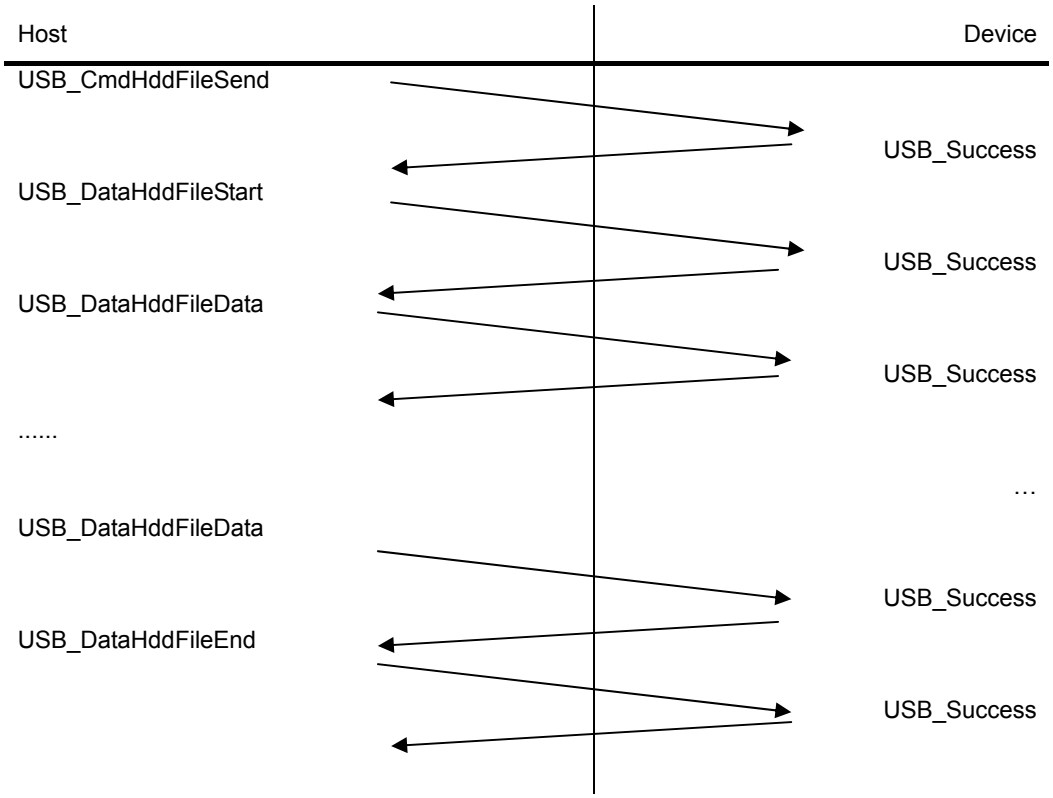
110	4	WindowsAttr	File attribute in Windows system(Host)

2.4. ErrorCode

Code	Name	Description
0x0001	CRC Error	CRC error
0x0002	Unknown Cmd	Unknown command
0x0003	Invalid Cmd	Invalid command
0x0004	Unknown Error	Unknown command
0x0005	Size Error	The size of block is invalid
0x0006	Run Error	Unknown error while running
0x0007	Memory Error	Memory is full

3. FLOWCHART

Flowchart that is transferring to host from device



4. TYPE

Data types are defined as follows.

```
typedef unsigned char   byte;
typedef unsigned short  word;
typedef unsigned long   dword;
```

5. TOOLS

5.1. GetCrc16

CRC-16 value can be calculated using the following polynomial with initial value 0.

$P(x) = x^{16} + x^{15} + x^2 + x^0$

```
word GetCrc16
(
        void    *data,                          In
        dword   size                            In
);
```

Parameters:

Parameter	Description
data	Data pointer to get CRC-16
size	Size of data
Return value	CRC-16 value

5.2. Get16bit

It reads 16-bit data in fixed address.

```
dword Get16bit
(
        void    *addr                           In
);
```

Parameters:

Parameter	Description
addr	Address of data to be read
Return value	16-bit Value after reading

example) addr → 0010 0101 0101 0011 0110 1011 0101 1101 0101 0000 1111…$_2$

return 값 = 0000 0000 0000 0000 **0010 0101 0101 0011**$_2$

5.3. Put16bit

It writes 16-bit data in fixed address.

```
void Put16bit
(
```

```
        void    *addr,                          Out
        word    data                            In
);
```

Parameters:

Parameter	Description
addr	Address of data to be written
data	16-bit Value to be written

예) Previous Execution :

addr → 0010 0101 0101 0011 0110 1011 0101 1101 0101 0000 1111…$_2$

data = **0000 0111 1101 0000**$_2$

After Execution:

addr → **0000 0111 1101 0000** 0110 1011 0101 1101 0101 0000 1111…$_2$

5.4. Get24bit

It reads 24-bit data in fixed address

```
dword Get24bit
(
        void    *addr                           In
);
```

Parameters:

Parameter	Description
addr	Address of data to be read
Return value	24-bit value after reading

예) addr → 0010 0101 0101 0011 0110 1011 0101 1101 0101 0000 1111…$_2$

return Value = 0000 0000 **0010 0101 0101 0011 0110 1011**$_2$

5.5. Put24bit

It writes 24-bit data in fixed address.

```
void Put24bit
(
        void    *addr,                          Out
        dword   data                            In
);
```

Parameters:

Parameter	Description
addr	Address to be written
data	24-bit value to be written

예) Previous Execution:

addr → 0010 0101 0101 0011 0110 1011 0101 1101 0101 0000 1111…$_2$

data = 0000 0000 **1100 0011 0000 0111 1101 0000**$_2$

After Execution:

addr → **1100 0011 0000 0111 1101 0000** 0101 1101 0101 0000 1111…$_2$

5.6. Get32bit.

It reads 32-bit data in fixed address.

```
dword Get32bit
(
        void    *addr                                   In
);
```

Parameters:

Parameter	Description
addr	Address of data to be read
Return value	32-bit value after reading

예) addr → 0010 0101 0101 0011 0110 1011 0101 1101 0101 0000 1111…$_2$

return 값 = **0010 0101 0101 0011 0110 1011 0101 1101$_2$**

5.7. Put32bit

It writes 32-bit data in fixed address.

```
void Put32bit
(
        void    *addr,                                  Out
        dword   data                                    In
);
```

Parameters:

Parameter	Description
addr	Address to be written
data	32-bit value to be written

예) Previous Execution:

addr → 0010 0101 0101 0011 0110 1011 0101 1101 0101 0000 1111…$_2$

data = **1111 0101 1100 0011 0000 0111 1101 0000**$_2$

After Execution

addr → **1111 0101 1100 0011 0000 0111 1101 0000** 0101 0000 1111…$_2$

5.8. ExtractMjd

It gets the date information (year/month/day) from MJD (Modified Julian Date) format

```
byte ExtractMjd
(
        word    mjd,            In
        word    *year,          Out
        word    *month,         Out
        word    *day,           Out
        word    *weekDay        Out
);
```

Parameters:

Parameter	Description
mjd	MJD Value
year	Year
month	Month
day	Date
weekDay	A day of week(0=Monday, 1=Tuesday, …, 6=Sunday)
Return value	0 = Not a Mjd format 1 = Complete

5.9. MakeMjd

It gets MJD from the date information of year/month/day .

```
word MakeMjd
(
        word    year,           In
        byte    month,          In
        byte    day             In
);
```

Parameters:

Parameter	Description
year	Year
month	Month
day	Day
Return value	0 = Can not support the year, month or day. 0x3AAC..0xFFFF = MJD 값

6. DEVICE

Only one program can open some device in windows system.

6.1. InitDevice

It initializes memory to be used in TFDLL.

It has to be called only one time.

```
void InitDevice
(
        void                                    -
);
```

6.2. DeviceArrival

It opens the device if the device can be opened.

```
BOOL DeviceArrival
(
        char    * pDeviceName,                  In
);
```

Parameters:

Parameter	Description
pDeviceName	device name
Return value	1 - Success 0 - Fail

6.3. DeviceRemoval

It closes the device.

```
BOOL DeviceRemoval
(
        char    *pDeviceName,                   In
);
```

Parameters:

Parameter	Description
pDeviceName	device name
Return value	1 - Success

	0 - Fail

6.4. GetDeviceVidPid

It gets the vendor identification and product identification.

```
BOOL GetDeviceVidPid
(
        dword    *pVid,                    Out
        dword    *pPid,                    Out
);
```

Parameters:

Parameter	Description
pVid	Vendor ID
pPid	Product ID
Return value	1 - Sucsess 0 - Fail

6.5. StartPnpNotice

If device is connected with windows system , windows OS informs with WM_DEVICECHANGE message

```
BOOL StartPnpNotice
(
        HWND    hwnd,                      In
);
```

Parameters:

Parameter	Description
hwnd	it is windows handle to be received message when a device is connected with windows system.
Return value	1 - The device is connected and opened. No need to call DeviceArrival 0 - The device is not connected.

6.6. TryDeviceOpen

This function can be used to open periodically when the device is connected but can not be opened due to using in other program.

```
BOOL TryDeviceOpen
(
        void                                    -
);
```

Parameters:

Parameter	Description
Return value	1 - Open Success 0 - Open Fail

6.7. UsbWrite

It writes some data in device.

It returns fail if it does not complete within 10sec.

```
BOOL UsbWrite
(
        byte    *pBuf,                          In
        dword   size,                           In
);
```

Parameters:

Parameter	Description
pBuf	Pointer of data to be written
size	Size of data to be written (range : 0x1 ~ 0x10000)
Return value	1 – Success 0 - Fail

6.8. UsbRead

It reads some data from the device.

It returns fail if it does not complete within 15sec.

```
dword UsbRead
(
        byte    *pBuf,                          In
        dword   size,                           In
);
```

Parameters:

Parameter	Description
pBuf	Pointer of data to be read
size	Size of data to be read (range : 0x1 ~ 0x10000)
Return value	Real size of read data

6.9. UsbCommandSend

It sends the predefined command with data to a device

```
BOOL UsbCommandSend
(
        dword   cmd,                    In
        byte    *pBuf,                  In
        dword   size,                   In
);
```

Parameters:

Parameter	Description
cmd	Command to be sent
pBuf	Pointer of data to be sent with command
size	Size of data to be sent
Return value	1 - Success 0 - Fail

7. FILE TASK

It is the task to transfer file to or from PC

7.1. UsbTaskFileSend

It sends a file from PC to TF5000PVR and creates the task to send file.

If the task completes sending the file, it sends PostMessage(hWnd, WM_COMMNAD,wParam,bSuccess);.

BSuccess(iParam) 1- Success

0 - Fail

```
BOOL UsbTaskFileSend
(
        char    *pDestName,             In
        char    *pSrcName,              In
        HWND    hWnd,                   In
        WPARAM wParam,                  In
);
```

Parameters:

Parameter	Description
pDestName	filename (full path) to be written in TF5000PVR
pSrcName	Filename to be sent
hWnd	Windows handle to receive the message when the job is completed or canceled
wParam	wParam value to be received with WM_COMMAND
Return value	1 - Success to start sending 0 - Fail to start sending

7.2. UsbTaskFileSendPosi

It sends a file from PC to TF5000PVR from the specific position of a file. It may be used to resume sending the file in case of some interruption.

If the task completes sending the file, it sends PostMessage(hWnd, WM_COMMNAD,wParam,bSuccess);.

BSuccess(iParam) 1- Success

0 - Fail

```
BOOL UsbTaskFileSendPosi
(
        char    *pDestName,             In
        char    *pSrcName,              In
        HWND    hWnd,                   In
        WPARAM wParam,                  In
        dword   highPosi,               In
        dword   lowPosi                 In
);
```

Parameters:

Parameter	의미
pDestName	filename (full path) to be written in TF5000PVR
pSrcName	filename to be sent
hWnd	Windows handle to receive the message when the job is completed or canceled.
wParam	wParam value to be received with WM_COMMAND
highPosi	high-order word of filesize
lowPosi	low-order doubleword of filesize
Return value	1 - Success to start sending 0 - Fail to start sending

7.3. UsbTaskFileReceive

It receives a file from TF5000PVR and creates a task to receive file.

If the task completes receiving the file, it sends PostMessage(hWnd,WM_COMMAND,wParam,bSuccess);

BSuccess(iParam) 1- Success

0 - Fail

```
BOOL UsbTaskFileReceive
(
        char    *pDestName,                     In
        char    *pSrcName,                      In
        HWND    hWnd,                           In
        WPARAM wParam,                          In
);
```

Parameters:

Parameter	Description
pDestName	Filename to be written in PC
pSrcName	filename (full path) to be read in TF5000PVR
hWnd	Windows handle to receive the message when the job is completed or canceled.
wParam	wParam value to be received with WM_COMMAND
Return value	1 - Success to start receiving 0 - Fail to start receiving

7.4. UsbTaskFileReceivePosi

It receives a file from TF5000PVR from the specific position of a file. It may be used to resume receiving the file in case of some interruption.

If the task completes sending the file, it sends PostMessage(hWnd, WM_COMMNAD,wParam,bSuccess);.

BSuccess(iParam) 1- Success

0 - Fail

```
BOOL UsbTaskFileReceivePosi
(
        char    *pDestName,                     In
```

```
        char    *pSrcName,                  In
        HWND    hWnd,                       In
        WPARAM  wParam,                     In
        dword   highPosi,                   In
        dword   lowPosi                     In
);
```

Parameters:

Parameter	의미
pDestName	Filename to be written in PC
pSrcName	filename (full path) to be read in TF5000PVR
hWnd	Windows handle to receive the message when the job is completed or canceled.
wParam	wParam value to be received with WM_COMMAND
highPosi	high-order word of filesize
lowPosi	low-order doubleword of filesize
Return value	1 - Success to start receiving 0 - Fail to start receiving

7.5. UsbTaskGetPercent

It gets the percentage of process using UsbTaskFileSend() or UsbTaskFileReceive()

```
word UsbTaskGetPercent
(
        void                                -
);
```

Parameters:

Parameter	Description
Return value	The current percentage of processing (0~ 100)

7.6. UsbTaskCancel

It cancels the task that is created by UsbTaskFileSend() or UsbTaskFileReceive()

```
BOOL UsbTaskCancel
(
        void                                -
);
```

Parameters:

Parameter	Description
Return value	1 – Cancel success 0 – Cancel Fail