

DMX4ALL PC-Interface

Programmierung über DMX4ALL.DLL

- DEUTSCH -



Inhaltsverzeichnis

DMX4ALL PC-INTERFACE PROGRAMMIERUNG	3
ALLGEMEINES.....	3
Beispiel: DMX-Kanäle setzen.....	4
Beispiel: DMX-Interface suchen und verbinden.....	4
BESCHREIBUNG DER API-FUNKTIONEN	5
CreateInterfaceList	5
GetInterfaceDetail	5
OpenPort.....	6
ClosePort	6
CheckPort	6
SetUniverse	7
GetProductID	7
FindInterface	8
GetComParameters.....	8
GetInfo	9
SetComParameters	9
GetDmx	10
GetBlackOut.....	12
SetBlackOut	12
GetNrOfChannelsOut.....	13
GetNrOfChannelsIn	13
SetMinChannels	14
SetNrOfChannels	14
RunLightScene	15
GetLightScene	15
SetPinout	16
SetMerge	16
ERROR-CODES	17
VERSIONS-HISTORIE	18

DMX4ALL PC-Interface Programmierung

Allgemeines

Die DMX4ALL-DLL stellt unterschiedliche Funktionen bereit, die für die Ansteuerung von DMX4ALL-PC-Interfaces vorgesehen sind. Die weitere Beschreibung richtet sich ausschließlich an Programmierer, die die Ansteuerung von DMX4ALL-Interfaces in eigene Software implementieren möchten.

Es werden zwei Funktionsgruppen von der DLL bereitgestellt, die sich durch ihre Definition unterscheiden. Die Funktionen mit „__“ beginnend sind mit `__stdcall` aufzurufen. Funktionen ohne „__“ in der Bezeichnung besitzen keine definierte Aufrufkonvention und sind aus Kompatibilitätsgründen weiterhin enthalten.

Beispiel: DMX-Kanäle setzen

```
#include "DMX4ALLinit.h"
#include "DMX4ALLdefs.h"

void Demofunction(void)
{
    // DLL laden
    Dmx4allDllOpen();

    // DMX4ALL-PC-Interface suchen und verbinden
    Dmx4allFindInterface();

    // DMX-Ausgangskanal 1 auf 255 setzen
    Dmx4allSetDmxCh(1, 255);

    // Verbindung schließen
    Dmx4allClearComPort();
}
```

Beispiel: DMX-Interface suchen und verbinden

```
#include "DMX4ALLinit.h"
#include "DMX4ALLdefs.h"

void Demofunction(void)
{
    // DLL laden
    Dmx4allDllOpen();

    // Interface-Liste erstellen
    DWORD NrOfFoundInterfaces=0;
    Dmx4allCreateInterfaceList(&NrOfFoundInterfaces);

    // Prüfen ob mindestens ein Interface vorhanden
    if(NrOfFoundInterfaces!=0)
    {
        char Description[50];
        int PortIndex;
        for(int i=0;i<NrOfFoundInterfaces;i++)
        {
            // Description vom Interface mit Index i lesen
            Dmx4allGetInterfaceDetail(i,&PortIndex,Description,50);

            // Verbindung zum Interface Herstellen
            Dmx4allSetComPort(PortIndex)

        }
    }

    ...

    // Verbindung schließen
    Dmx4allClearComPort();
}
```

Beschreibung der API-Funktionen

CreateInterfaceList

Beschreibung

Erstellt eine Liste mit den angeschlossenen DMX4ALL-Interfaces

Definition

```
bool Dmx4allCreateInterfaceList (DWORD* numDevs) ;
```

Parameter

<numDevs> Pointer auf ein DWORD, Rückgabe der Anzahl der gefundenen DMX4ALL Interfaces

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache kann dann über *LastError()* ermittelt werden.

GetInterfaceDetail

Beschreibung

Liefert detailinformationen über das DMX4ALL-Interface aus der Liste an Position *index*

Definition

```
bool Dmx4allGetInterfaceDetail (DWORD index, DWORD* PortIndex, char* pDescription, int Length) ;
```

Parameter

<index> 0..numDevs: Index in die Interface-Liste
<PortIndex> Rückgabe des Ports für SetPort()
<pDescription> Rückgabe der Interfacebeschreibung
<Length> Größe des CHAR-Arrays für Description

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache kann dann über *LastError()* ermittelt werden.

OpenPort

Beschreibung

Öffnet die Verbindung zum DMX4ALL-Interface

Definition

```
bool Dmx4allOpenPort (int Port) ;
```

Parameter

<Port> Zu öffnender Port. 1...255 = COM-Port; 256 oder größer = USB

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

ClosePort

Beschreibung

Schließt die Verbindung mit dem DMX4ALL-Interface

Definition

```
bool Dmx4allClosePort () ;
```

Parameter

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

CheckPort

Beschreibung

Überprüft, ob an dem angegebene Port ein DMX4ALL-Interface angeschlossen ist

Definition

```
bool Dmx4allCheckPort (int Port) ;
```

Parameter

<Port> Zu überprüfender Port. 1...255 = COM-Ports; 256 und größer = USB

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

SetUniverse

Beschreibung

Wählt das DMX-Universum aus, auf das die nächsten Funktionen angewendet werden sollen

Definition

```
bool Dmx4allSetUniverse(int Universe);
```

Parameter

<Universe> 0...15 Index des aktuellen DMX-Universums

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

GetProductID

Beschreibung

Liefert die ID das angeschlossene DMX4ALL-Interface

Definition

```
bool Dmx4allGetProductID(int* pProdID);
```

Parameter

<pProdID>	2	Mini-USB-DMX-Interface
	3	PC-DMX-Interface V3
	4	PC-DMX-Interface V4
	5	Easy-Light-Control
	6	USB-DMX STAGE-PROFI
	7	LAN-DMX STAGE-PROFI
	8	DMX-Player S
	9	DMX-Player XL
	10	DMX-Player XS
	11	DMX-Player M
	20	DMX-IN STAGE-PROFI

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

FindInterface

Beschreibung

Sucht ein DMX4ALL-Interface am USB und anschließend an den COM-Ports. Zu dem zuerst gefundenen Interface wird die Verbindung hergestellt.

Definition

```
bool Dmx4allFindInterface(int MaxPort = 255);
```

Parameter

<MaxPort> Gibt an, bis zu welchem COM-Port ein Interface gesucht werden soll

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache kann dann über *LastError()* ermittelt werden.

GetComParameters

Beschreibung

Gibt die verwendeten Parameter zurück

Definition

```
bool Dmx4allGetComParameters(int* pPort, DWORD* pBaudrate);
```

Parameter

<pPort> Zeiger auf Integerwert der den verwendeten COM-Port angibt
<pBaudrate> Zeiger auf Integerwert der die verwendete Baudrate angibt

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache kann dann über *LastError()* ermittelt werden.

GetInfo

Beschreibung

Liefert Informationen über das angeschlossene DMX4ALL-Interface

Definition

```
bool Dmx4allGetInfo(VERSION_INFO* pVersionInfo);
```

Parameter

<pVersionInfo>

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

SetComParameters

Beschreibung

Definition

```
bool Dmx4allSetComParameters(UINT Baudrate);
```

Parameter

<Baudrate> Zu verwendende Baudrate (19200 / 38400)

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

GetDmx

Beschreibung

Definition

```
bool Dmx4allGetDmx( int FirstChannel,  
                    int NrOfBytes,  
                    BYTE* pData);
```

Parameter

<FirstChannel>
<NrOfBytes>
<pData>

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

GetInDmx

Beschreibung

Definition

```
bool Dmx4allGetInDmx( int FirstChannel,  
                      int NrOfBytes,  
                      BYTE* pData);
```

Parameter

<FirstChannel>
<NrOfBytes>
<pData>

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

SetDmxCh

Beschreibung

Überträgt einen DMX-Wert an das DMX4ALL-Interface, der auf dem DMX-Ausgang ausgegeben wird.

Definition

```
bool Dmx4allSetDmxCh(int Kanal, BYTE Data);
```

Parameter

<Kanal>	DMX-Kanal 1...512
<Data>	DMX-Wert 0...255

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.

Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache kann dann über *LastError()* ermittelt werden.

SetDMX

Beschreibung

Überträgt DMX-Werte in das DMX4ALL-Interface, die dann auf dem DMX-Ausgang ausgegeben werden.

Definition

```
bool Dmx4allSetDmx( int FirstChannel,  
                    int NrOfBytes,  
                    BYTE* pData);
```

Parameter

<FirstChannel>	Gibt den ersten zu schreibenden DMX-Kanal an
<NrOfBytes>	Gibt die Anzahl der zu schreibenden Kanäle an
<pData>	Zeiger auf ein BYTE-Array der Größe NrOfBytes das die DMX-Werte enthält

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.

Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache kann dann über *LastError()* ermittelt werden.

GetBlackOut

Beschreibung

Liefert den aktuellen Status über die BlackOut-Einstellung.

Definition

```
bool Dmx4allgetBlackOut ()
```

Parameter

Rückgabewert

TRUE BlackOut ein

FALSE BlackOut aus

SetBlackOut

Beschreibung

Definition

Parameter

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.

Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache kann dann über *LastError()* ermittelt werden.

GetNrOfChannelsOut

Beschreibung

Definition

Parameter

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

GetNrOfChannelsIn

Beschreibung

Definition

Parameter

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

SetMinChannels

Beschreibung

Definition

Parameter

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache kann dann über *LastError()* ermittelt werden.

SetNrOfChannels

Beschreibung

Definition

Parameter

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache kann dann über *LastError()* ermittelt werden.

RunLightScene

Beschreibung

Definition

Parameter

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

GetLightScene

Beschreibung

Definition

Parameter

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

SetPinout

Beschreibung

Definition

Parameter

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über `LastError()` ermittelt werden.

SetMerge

Beschreibung

Definition

Parameter

Rückgabewert

Wurde die Funktion erfolgreich ausgeführt, so wird TRUE zurückgegeben.
Im Fehlerfall wird FALSE zurückgegeben. Die genaue Fehlerursache
kann dann über *LastError()* ermittelt werden.

Error-Codes

Bei fehlerhaft ausgeführten Funktionen wird FALSE zurückgegeben. Einen Fehlercode der den aufgetretenen Fehler spezifiziert wird von der Funktion LastError() bereitgestellt.

Die Fehlercodes sind in der folgenden Tabelle beschrieben:

[illegible]

Versions-Historie

DMX4ALL-DLL V1.1.0

- Erste DLL-Version

DMX4ALL-DLL V1.2.0

- Funktionen mit `__stdcall` implementiert

DMX4ALL-DLL V1.5.0

- CString Variablen gegen CHAR-Arrays ersetzt
- Multi-Universe Unterstützung



Dipl.-Ing. (FH) Markus Siwek
Sophienstr. 8
D-44791 Bochum
Germany

© Copyright 2008 Markus Siwek

Alle Rechte vorbehalten. Kein Teil dieses Handbuches darf in irgendeiner Form (Fotokopie, Druck, Mikrofilm oder in einem anderen Verfahren) ohne ausdrückliche schriftliche Genehmigung reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Alle in diesem Handbuch enthaltenen Informationen wurden mit größter Sorgfalt und nach bestem Wissen zusammengestellt. Dennoch sind Fehler nicht ganz auszuschließen.

Aus diesem Grund sehe ich mich dazu veranlasst, darauf hinzuweisen, dass ich weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen kann. Dieses Dokument enthält keine zugesicherten Eigenschaften. Die Anleitung und die Eigenschaften können jederzeit und ohne vorherige Ankündigung geändert werden.

Der Einsatz der Geräte in Anwendungen, wo die Sicherheit von Personen von deren Funktion abhängt, ist untersagt.