

**DMXManager.dll**  
Version 2

The functions description of dynamic library for usage DMX Manager controller with the self designed software.

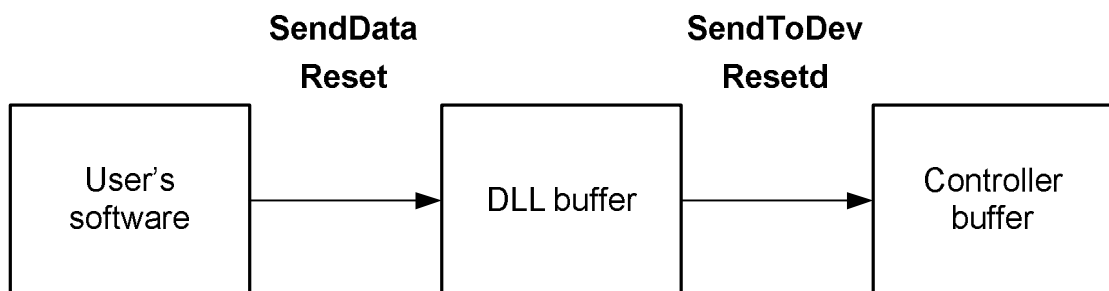
**1. General provisions**

The given document describes features of usage the DMXManager.dll library in the own software. The description of functions, their arguments, and also returned values is resulted.

The library was tested successfully and are used today with software developed in C#, C Builder, Delphi.

All features and the conditionalities were considered allowing to use the given library in the applications developed in various development environments. So, the data in library is transferred not by data arrays, but by data pairs (channel number; channel value). For synchronous data transfer of all 512 channels in the controller, in library the program buffer is realised.

Data transfer process is carried out as follows:



function SendData are transferred data in DLL program buffer. After transmission of demanded quantity of channels (1. 512) in the buffer, function SendToDev is unitary called, which transfers the data directly to the controller.

Procedure of cleaning of the buffer is similarly realised. Function Reset clears DLL program buffer, and function Resetd clears the controller buffer.

In the current version of library You may use several controllers. For this purpose in some functions the additional argument is entered. Also function CheckDev for quantifying of on-line controllers and function GetIndex for definition the controller index under its serial number.

The library is developed in the Delphi environment, in the given key the description of functions will be resulted.

**Please, use library ONLY after successful installation of the controller drivers!**

## 2. The list of functions.

SendData,  
SendToDev,  
Reset,  
Resetd,  
Start,  
CheckDev,  
GetIndex;

### **function SendData(ch\_num: word; data: byte; dev: word): Integer;**

***destination:***

transfers the data in DLL program buffer of the controller with an index dev.  
It is called for each pair of the data (channel\_number; channel\_value),  
where

*channel\_number* – channel number in which it is necessary to set value *channel\_value*.  
range (0..511).  
*channel\_value* – transferred value. range (0..255).

***usage:***

```
...
var data: array[0..511] of byte;
i: word; k: word;
...
for i:= 0 to 511 do
  if SendData(i, data[i], k) = -1 break;
...
```

***result:***

[0..511] – number of current transmitted channel in DLL program buffer of the controller with an index dev - in case of successful transmission  
-1 – error: drivers are not installed, the port is used by other software.

### **function SendToDev(dev: word): Integer;**

***destination:***

data transfer, from DLL program buffer with an index dev to the controller with an index dev.

***usage:***

```
SendToDev(k);
```

***result:***

0 – successful buffer cleaning;  
-1 –function error: drivers are not installed, the port is used by other software.  
-2 –error of connection with the controller: the controller is disconnected, drivers are not installed, the controller is used by other software.

**function Reset(dev: word): Integer;*****destination:***

clears DLL program buffer with an index dev.

***usage:***

Reset(k);

***result:***

0 – successful buffer cleaning;  
-1 – error: drivers are not installed, the port is used by other software.

**function Resetd(dev: word): Integer;*****destination:***

clears the controller buffer with an index dev (set values of all channels in 0).

***usage:***

Resetd(k);

***result:***

0 – successful buffer cleaning;  
-1 –function error: drivers are not installed, the port is used by other software.  
-2 – error of connection with the controller: the controller is disconnected, drivers are not installed, the controller is used by other software.

**function Start: Integer;*****destination:***

called once at start of the software for controller(s) initialization.

***usage:***

Start;

***result:***

0 – successfull initialization;  
-1 – initialization error: all controllers are disconnected, drivers are not installed, the controller is used by other software.

**function GetIndex(dev: word): string;*****destination:***

returns the controller serial number by its index dev.

Because the index of controllers in system is dynamic and depends on many factors (sequence of connection of controllers, both concerning time, and concerning ports, etc.), but its serial number always the same, so current index of the controller is defining by means of the given function.

***usage:***

var str:string;

...

str:=GetIndex(k);

**result:**

str – serial number of checked controller.

**function CheckDev: Integer;**

**destination:**

check quantity of the on-line controllers in system.

**usage:**

var m: word;

...

m:=CheckDev;

**result:**

m – quantity of connected controllers.

Thank for DMX Manager controller usage!!!

---

For correspondence please use:  
e-mail: [dmxmanager@mail.ru](mailto:dmxmanager@mail.ru)  
ICQ: 238896486