

Manual

USB Products

WERMA Signaltechnik GmbH + Co. KG

Version 2.4 – 2018



Content

1.	Introduction	3
2.	Technical information	4
2.1	System requirements	4
3.	Driver Installation	5
3.1	Automatic Driver Installation	5
3.2	Manual driver installation	5
4.	Operation with virtual COM Port	9
4.1	Settings	9
4.2	Serial Commands for the USB Terminal Element	9
4.2.1	Version	9
4.2.2	Name	9
4.2.3	Write	10
4.2.4	Read	11
4.2.5	Power-on state	11
4.3	Serial Commands for the 816 LED Beacon USB	12
4.3.1	Version	12
4.3.2	Name	12
4.3.3	Read	12
4.3.4	Write	13
4.3.5	Power-on colour	13
5.	Operation with Dynamic Link Library (DLL)	14
5.1	Introduction	14
5.2	Related files and executables	14
	Runtime files:	14
5.3	DLL Functions for the USB Terminal Element	14
5.3.1	Function Overview	14
5.3.2	Basic Functions	15
5.3.2.1	SetUsbDevice	15
5.3.2.2	GetDeviceSerial	15
5.3.2.3	GetDeviceLocation	15
5.3.2.4	SetDeviceStatus	16
5.3.2.5	SetStatusOnPower	16
5.3.2.6	GetDeviceStatus	17
5.3.2.7	GetFirmwareVersion	17
5.3.2.8	GetLibVersion	18
5.4	DLL Functions for the 816 Multicolour Beacon USB	19
5.4.1	Function Overview	19
5.4.2	Functions	19
5.4.2.1	GetRGBDevice	19
5.4.2.2	GetRGBDeviceSerial	20
5.4.2.3	GetRGBDeviceLocation	20
5.4.2.4	SetRGBStatus	21
5.4.2.5	SetRGBStatusOnPower	21
5.4.2.6	GetRGBStatus	22
5.4.2.7	GetRGBFirmwareVersion	22
5.4.2.8	GetLibVersion	23

1. Introduction

The ideal solution for applications which require a personal computer is a WERMA USB product. Developed in-house at WERMA, the innovative products **KombiSIGN 71/72** and the **816 LED beacon** (multicolour) allow direct PC actuation for fast and convenient commissioning. As well as actuation via a DLL (Dynamic Link Library), the element can also be simply commissioned by means of VCP (Virtual COM Port) actuation. This ensures a simple integration into any customer-specific software.

Our USB products are using the **FTDI FT230X** from the FTDI FT-X series chip for the USB to UART bridge, for which license-free drivers are provided by FTDI for all major OS.

The terminal element for KombiSIGN 71/72 and the 816 LED beacon (multicolour) with USB interface requires neither its own separate power supply nor any additional hardware.

WERMA Signaltechnik GmbH + Co.KG

D-78604 Rietheim-Weilheim

Fon: +49 (0)7424 / 9557-222

Fax: +49 (0)7424 / 9557-44

support@werma.com

www.werma.com

WERMA reserves the right to make technical improvements to the product and accepts no responsibility for mistakes or printing errors which may be contained in this documentation.

As of: 01/2018

310.640.008

2. Technical information

2.1 System requirements

- USB port (port must provide 500 mA, otherwise a USB hub with external power supply must be used)
- Windows 7, Windows 8, 8.1, Windows 10, Windows Server 2008 R2 and Windows server 2012 R2 (with FTDI-driver version 2.12.28; driver provided by WEMA)

With FTDI VCP-drivers

- Linux, Mac OS X, Windows CE (Version 4.2 and greater), Windows RT, Android
(For further information and drivers, see <http://www.ftdichip.com>)

With FTDI-driver version 2.08.24 (not recommended)

- Windows XP, Vista and Server 2003
(For further information and drivers, see <http://www.ftdichip.com>)

3. Driver Installation

3.1 Automatic Driver Installation

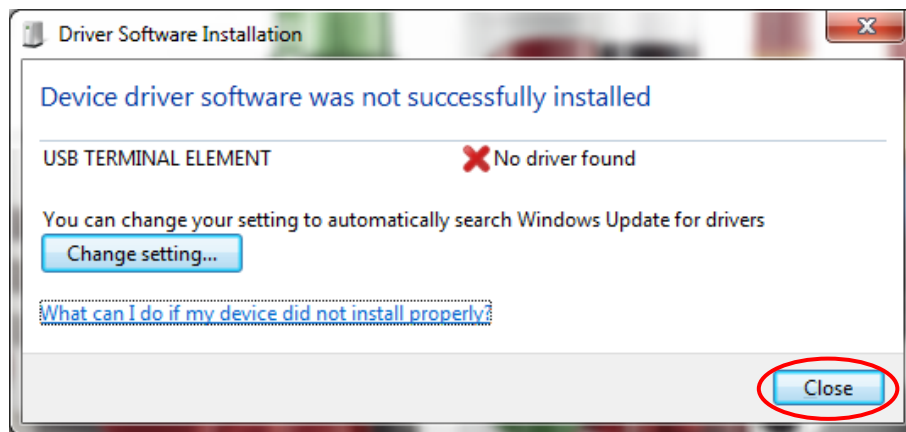
1. Download the install package: www.werma.com/download
2. Unpack the ZIP file, we need the subfolder "drivers".
3. On Windows based Systems, please run the file Setup_USB_Driver_Win_7_8_10.exe.
This setup installs the drivers for Windows 7, Windows 8, 8.1 and Windows 10.

Note: You must have administrator rights to install the drivers.

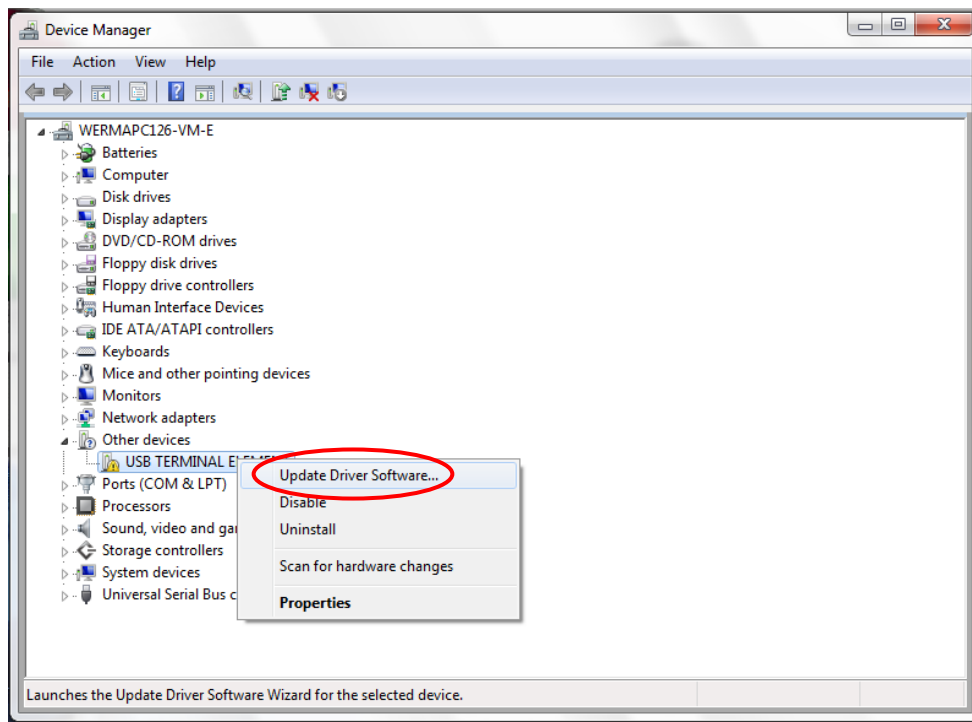
3.2 Manual driver installation

1. Download the install package: www.werma.com/download
2. Unpack the ZIP file, we need the subfolder "drivers" for the manual driver installation

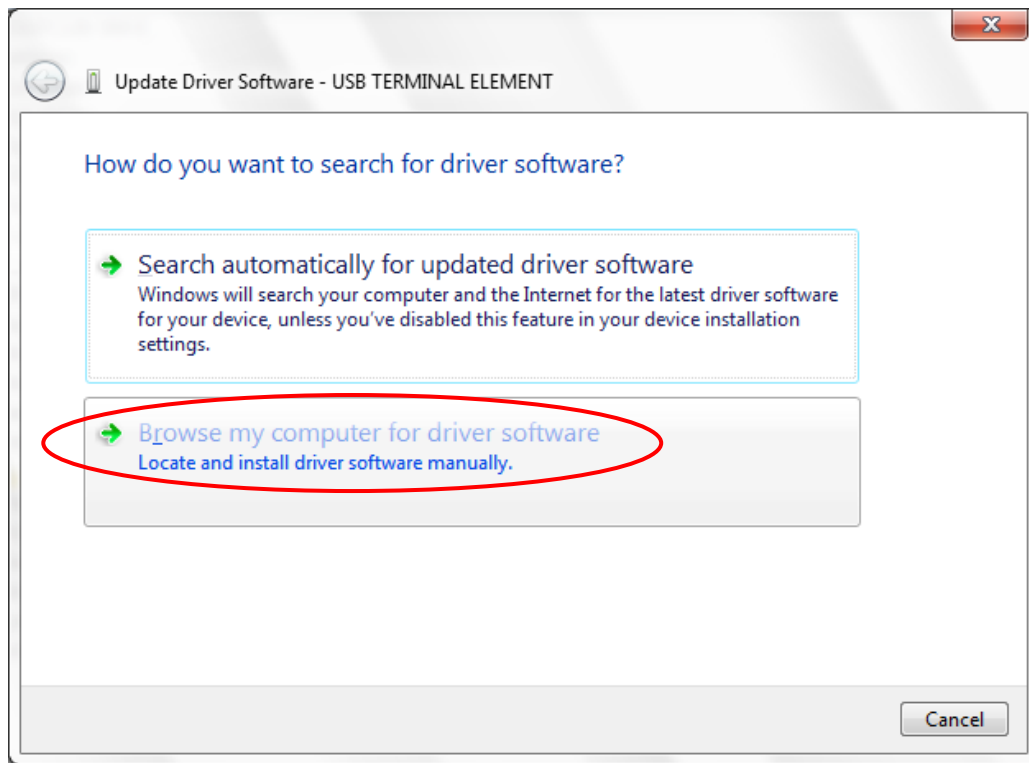
Note: Manual driver installation is only required if the USB product is not recognised. Windows will open the following window:



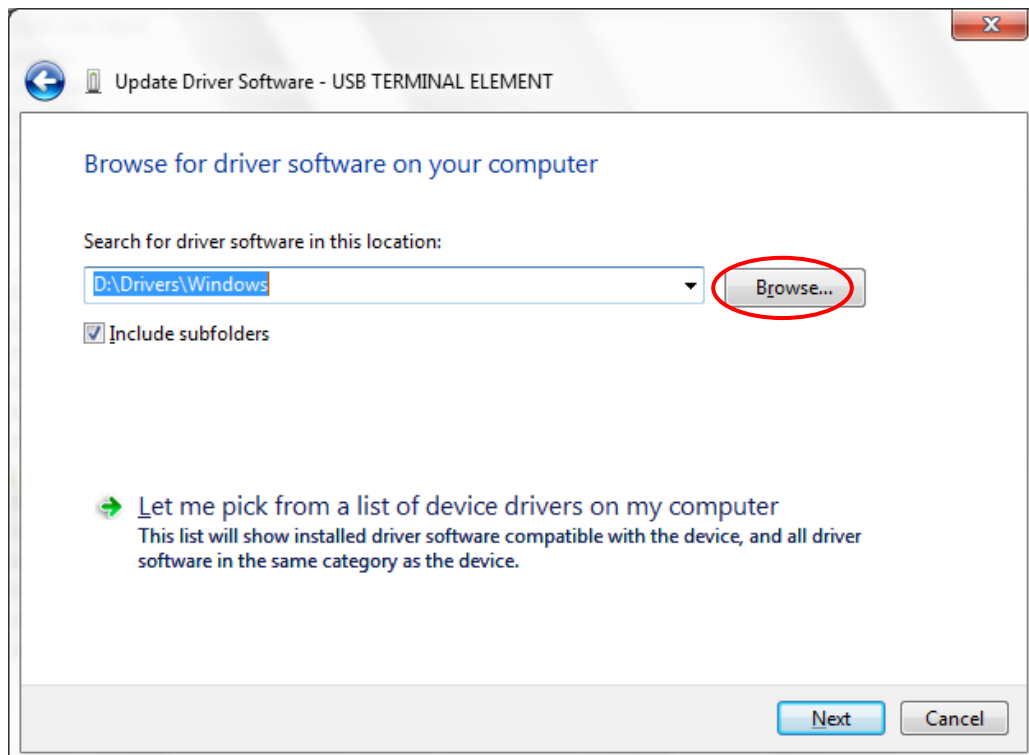
Close this window and open the Device Manager (Computer/Properties/Device Manager). Right click on USB TERMINAL ELEMENT / WERMA MULTICOLOUR BEACON and then click on "Update Driver Software".



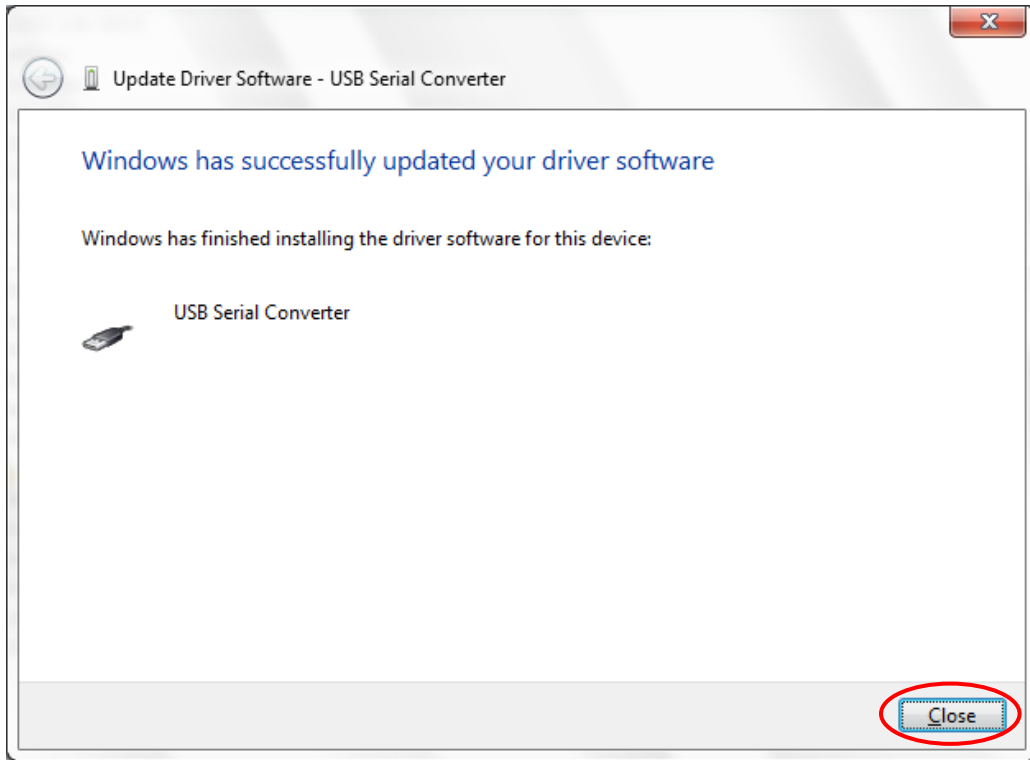
Select "Browse my computer for driver software".



Specify the location of the driver by clicking "Browse". Select the "Driver" folder from the unpacked ZIP file and confirm by clicking "Next".



The driver for the USB products has been successfully installed. Click "Close" to finish the update.



Note: The USB product is now ready for operation.

4. Operation with virtual COM Port

4.1 Settings

For the USB serial port you need the following settings:

Baud rate	9600 bits per second
Data bits	8
Stop bit	1
Parity	None
Flow control	No

Note: Every command has to end with a return <CR> and/or a line feed <LF>.

The arrow (→) represents the return value of the product. On error it would be "Command Error". All return values are finalized by <CR><LF>.

4.2 Serial Commands for the USB Terminal Element

4.2.1 Version

Type	Command	Description
Version	V	<p>Information about the version.</p> <p>V<CR></p> <p>Example: V<CR> → Version:2.1.0<CR><LF></p>

4.2.2 Name

Type	Command	Description
Name	N	<p>Information about the name.</p> <p>N<CR></p> <p>Example: N<CR> → USB TERMINAL ELEMENT<CR><LF></p>

4.2.3 Write

Type	Command	Description
Write	WRXXXXX	<p>Sets the output, X could be:</p> <p>0 → Off</p> <p>1 → On</p> <p>2 → Blinking mode 1</p> <p>3 → Blinking mode 2</p> <p>X → Don't care, no change on that tier</p>

WRXXXXX<CR>

Return

Tier 5

Tier 4

Tier 3

Tier 2

Tier 1

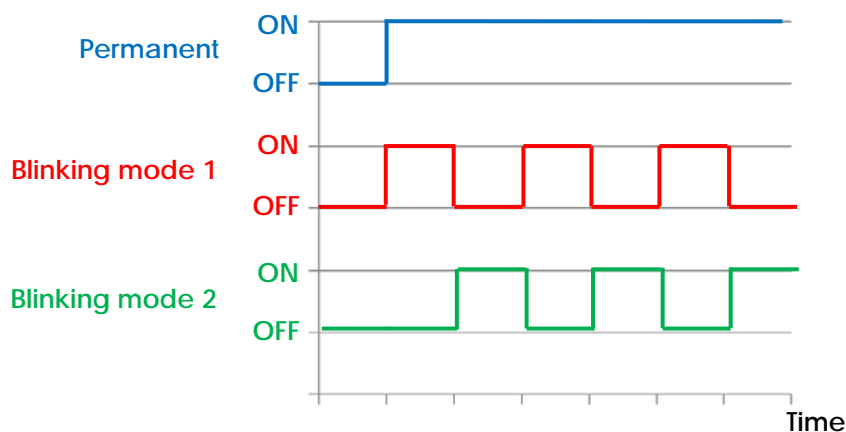
Write

Example:

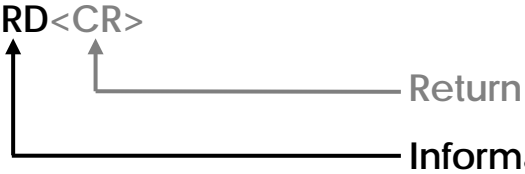
WR120X3<CR> → OK<CR><LF>

(Tier 1: On; Tier 2: Blinking mode 1; Tier 3: Off; Tier 4: Don't care; Tier 5: Blinking mode 2)

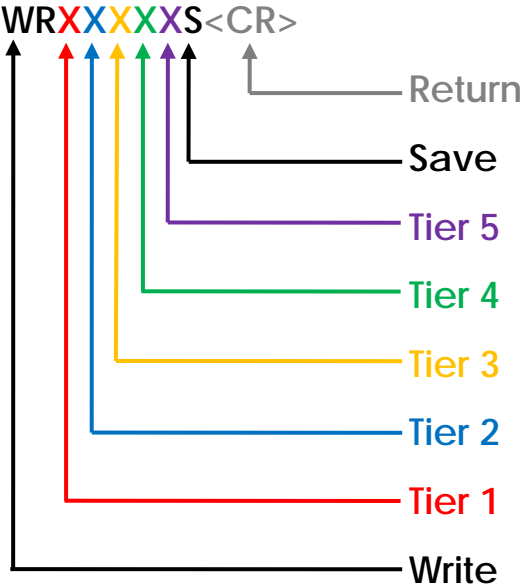
Note: Blinking mode 1 and Blinking mode 2 differ by blinking alternately (see diagram).



4.2.4 Read

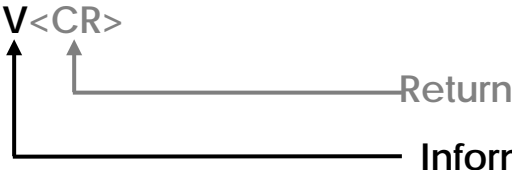
Type	Command	Description
Read	RD	<p>Information of current status.</p> <p>RD<CR></p>  <p>Example: RD<CR> → 02131<CR><LF> (Tier 1: Off; Tier 2: Blinking mode 1; Tier 3: On; Tier 4: Blinking mode 2; Tier 5: On)</p>

4.2.5 Power-on state

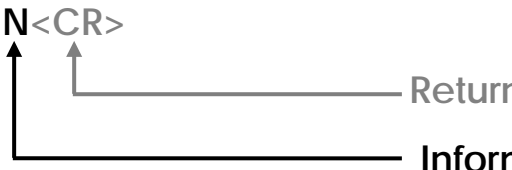
Type	Command	Description
Default output state on power	WRXXXXXS	<p>Sets the output, X could be:</p> <p>0 → Off 1 → On 2 → Blinking mode 1 3 → Blinking mode 2</p> <p>WRXXXXXS<CR></p>  <p>Example: WR12003S<CR> → OK<CR><LF> (Tier 1: On; Tier 2: Blinking mode 1; Tier 3: Off; Tier 4: Off; Tier 5: Blinking mode 2)</p>

4.3 Serial Commands for the 816 LED Beacon USB

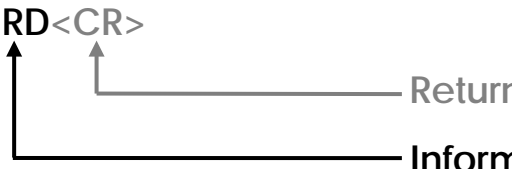
4.3.1 Version

Type	Command	Description
Version	V	<p>Information about the version.</p> <p>V<CR></p>  <p>Return</p> <p>Information of the version</p> <p>Example: V<CR> → Version:2.0.0<CR><LF></p>

4.3.2 Name

Type	Command	Description
Name	N	<p>Information about the name.</p> <p>N<CR></p>  <p>Return</p> <p>Information of the name</p> <p>Example: N<CR> → WERMA MULTICOLOUR BEACON<CR><LF></p>

4.3.3 Read

Type	Command	Description
Read	RD	<p>Information of current status.</p> <p>RD<CR></p>  <p>Return</p> <p>Information of status</p> <p>Example: RD<CR> → 64 64 00<CR><LF> (R = 100%, G = 100%, B = 0%);</p>

4.3.4 Write

Type	Command	Description
Write	WR 00 RR GG BB	<p>Sets the colour of the beacon, RR/GG/BB could be: 00_{hex} to 64_{hex} (101 different values) 00 = 0% 64 = 100% 0C = 12%</p> <p>WR 00 XX XX XX <CR></p> <p>Return Value blue Value green Value red Command</p> <p>Example: WR 00 64 64 00 <CR> → OK <CR> <LF> (R = 100%, G = 100%, B = 0%);</p>

4.3.5 Power-on colour

Type	Command	Description
Default output state on power	WR 00 RR GG BBS	<p>WR 00 XX XX XXS <CR></p> <p>Return Save Value blue Value green Value red Command</p> <p>Example: WR 00 64 64 00S <CR> → OK <CR> <LF> (R = 100%, G = 100%, B = 0%);</p>

5. Operation with Dynamic Link Library (DLL)

5.1 Introduction

The user controls the USB module by using a specified dynamic link library (DLL) file, included to the ZIP file. The DLL contains several functions to set and retrieve the light module status.

The control over the light module is generally carried out by a customer application. The control over the module from the user application is achieved via a specific DLL which shares all the commonly needed functions. The DLL library is a standard windows functions library which can be easily used with the most common application development tools such as Visual Basic, Visual C++ etc.

5.2 Related files and executables

Runtime files:

The DLL Library should reside in a file located in the same folder with user project .EXE file. This file communicates with the USB module.

slma.dll	Contains functions which provide control over the USB module
slma_x64.dll	Use this DLL for 64 bit systems: Contains functions which provide control over the USB module

5.3 DLL Functions for the USB Terminal Element

5.3.1 Function Overview

Basic Functions

SetUsbDevice	Returns the number of devices currently found in the system
GetDeviceSerial	Gets the specified device serial code
GetDeviceLocation ¹	Gets the Location ID of the USB port, where the specified device is connected
SetDeviceStatus	Sets the specified device status (ON, OFF, Blinking mode 1, Blinking mode 2) for all outputs
SetStatusOnPower	Sets the specified device power-on status (ON, OFF, Blinking mode 1, Blinking mode 2) for all outputs
GetDeviceStatus	Gets the specified device output status (ON, OFF, Blinking mode 1, Blinking mode 2) for all outputs
GetFirmwareVersion	Gets the specified device firmware version
GetLibVersion	Gets the currently used library version

¹ Please note that Linux, Mac OS X and Windows CE and also USB 3.0-ports do not support location IDs.

5.3.2 Basic Functions

5.3.2.1 SetUsbDevice

int SetUsbDevice()

SetUsbDevice retrieves the devices number currently found in the system.

Parameters

None

Return Value

The amount of devices currently found in the system. When no devices are found, zero will be returned.

5.3.2.2 GetDeviceSerial

int GetDeviceSerial([in] int DevIdx, [out] int * TypeCode, [out] int * BatchCode, [out] int * SerialCode)

GetDeviceSerial gets the specified device serial code. This code is unique and can be used to distinguish between various devices currently installed in the system.

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

TypeCode

Pointer to a variable that will contain the device type code.

BatchCode

Pointer to a variable that will contain the device batch code.

SerialCode

Pointer to a variable that will contain the device serial code.

Return Value

1 if successful, otherwise the return value is -1.

5.3.2.3 GetDeviceLocation

int GetDeviceLocation ([in] int DevIdx, [out] int * Loc)

GetDeviceLoation gets the Location ID of the USB port, where the device is connected. The Location ID is unique and can be used to distinguish between various devices currently installed in the system.*

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

Loc

Pointer to a variable that will contain the Location ID of the USB-Port.

Return Value

1 if successful, otherwise the return value is -1.

* Please note that Linux, Mac OS X and Windows CE and also USB 3.0-ports do not support location IDs

5.3.2.4 SetDeviceStatus

int SetDeviceStatus ([in] int DevIdx, [in] char Pos1, [in] char Pos2, [in] char Pos3, [in] char Pos4, [in] char Pos5)

SetDeviceStatus sets the status for all outputs.

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

Pos1, Pos2, Pos3, Pos4, Pos5

Sets port 1, 2, 3, 4 and 5 to OFF, ON, Blinking mode 1 or Blinking mode 2, depending on value.

Macro	Value as Char	Value in ASCII	Description
OFF	0	0x30	Port off
ON	1	0x31	Port on
Blinking mode 1	2	0x32	Blinking mode 1
Blinking mode 2	3	0x33	Blinking mode 2
Don't Care	X	0x58	No change on that port

Return Value

1 if successful, otherwise the return value is -1.

5.3.2.5 SetStatusOnPower

int SetStatusOnPower ([in] int DevIdx, [in] char Pos1, [in] char Pos2, [in] char Pos3, [in] char Pos4, [in] char Pos5)

SetDeviceStatus sets the status for all outputs.

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

Pos1, Pos2, Pos3, Pos4, Pos5

Sets port 1, 2, 3, 4 and 5 to OFF, ON, Blinking mode 1 or Blinking mode 2, depending on value.

Macro	Value as Char	Value in ASCII	Description
OFF	0	0x30	Port off
ON	1	0x31	Port on
Blinking mode 1	2	0x32	Blinking mode 1
Blinking mode 2	3	0x33	Blinking mode 2

Return Value

1 if successful, otherwise the return value is -1.

5.3.2.6 GetDeviceStatus

int GetDeviceStatus ([in] int DevIdx, [out] char * Pos1, [out] char * Pos2, [out] char * Pos3, [out] char * Pos4, [out] char * Pos5)

GetDeviceStatus gets the status for all outputs.

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

Pos1, Pos2, Pos3, Pos4, Pos5

Pointer to a variable that will contain the status of the output.

Macro	Value as Char	Value in ASCII	Description
OFF	0	0x30	Port off
ON	1	0x31	Port on
Blinking mode 1	2	0x32	Blinking mode 1
Blinking mode 2	3	0x33	Blinking mode 2

Return Value

1 if successful, otherwise the return value is -1.

5.3.2.7 GetFirmwareVersion

int GetFirmwareVersion ([in] int DevIdx, [out] int * Major, [out] int * Minor, [out] int * Build)

GetFirmwareVersion gets the specified device firmware version.

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

Major

Pointer to a variable that will contain the device major firmware version value.

Minor

Pointer to a variable that will contain the device minor firmware version value.

Build

Pointer to a variable that will contain the device build firmware version value.

Return Value

1 if successful, otherwise the return value is -1.

5.3.2.8 GetLibVersion

int GetLibVersion ([out] int * Major, [out] int * Minor)

GetLibVersion gets the currently used library version.

Parameters

Major

Pointer to a variable that will contain the library major version value.

Minor

Pointer to a variable that will contain the library minor version value.

Return Value

1 if successful, otherwise the return value is -1.

5.4 DLL Functions for the 816 Multicolour Beacon USB

5.4.1 Function Overview

GetRGBDevice	Gets the number of beacons currently found in the system
GetRGBDeviceSerial	Gets the specified device serial code
GetRGBDeviceLocation ¹	Gets the Location ID of the USB port, where the specified device is connected
SetRGBStatus	Sets the specified device colour
SetRGBStatusOnPower	Sets the specified device power-on colour
GetRGBStatus	Gets the specified device colour
GetRGBFirmwareVersion	Gets the specified device firmware version
GetLibVersion	Gets the currently used library version

¹ Please note that Linux, Mac OS X and Windows CE and also USB 3.0-ports do not support location IDs.

5.4.2 Functions

5.4.2.1 GetRGBDevice

int GetRGBDevice()

GetRGBDevice retrieves the devices number currently found in the system.

Parameters

None

Return Value

The amount of devices currently found in the system. When no devices are found, zero will be returned.

5.4.2.2 GetRGBDeviceSerial

```
int GetRGBDeviceSerial ( [in] int DevIdx, [out] int * TypeCode, [out] int * BatchCode, [out] int * SerialCode)
```

GetDeviceSerial gets the specified device serial code. This code is unique and can be used to distinguish between various devices currently installed in the system.

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

TypeCode

Pointer to a variable that will contain the device type code.

BatchCode

Pointer to a variable that will contain the device batch code.

SerialCode

Pointer to a variable that will contain the device serial code.

Return Value

1 if successful, otherwise the return value is -1.

5.4.2.3 GetRGBDeviceLocation

```
int GetRGBDeviceLocation ( [in] int DevIdx, [out] int * Loc)
```

GetDeviceLoation gets the Location ID of the USB port, where the device is connected. The Location ID is unique and can be used to distinguish between various devices currently installed in the system.*

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

Loc

Pointer to a variable that will contain the Location ID of the USB-Port.

Return Value

1 if successful, otherwise the return value is -1.

* Please note that Linux, Mac OS X and Windows CE and also USB 3.0-ports do not support location IDs

5.4.2.4 SetRGBStatus

int SetRGBStatus ([in] int DevIdx, [in] int * Red, [in] int * Green, [in] int * Blue)

SetRGBStatus sets the colour of the specified device

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

Red, Green, Blue

Sets the device colour, depending on values.

Value as Int	Description
0 (0x00)	Colour 0%
100 (0x64)	Colour 100%
12 (0x0C)	Colour 12%

Return Value

1 if successful, otherwise the return value is -1.

5.4.2.5 SetRGBStatusOnPower

int SetRGBStatus ([in] int DevIdx, [in] int * Red, [in] int * Green, [in] int * Blue)

SetRGBStatus sets the power-on colour of the specified device

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

Red, Green, Blue

Sets the device colour, depending on values.

Value as Int	Description
0 (0x00)	Colour 0%
100 (0x64)	Colour 100%
12 (0x0C)	Colour 12%

Return Value

1 if successful, otherwise the return value is -1.

5.4.2.6 GetRGBStatus

int GetRGBStatus ([in] int DevIdx, [out] int * Red, [out] int * Green, [out] int * Blue)

GetDeviceStatus gets the current colour of the specified device

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

Red, Green, Blue

Sets the device colour, depending on values.

Value as Int	Description
0 (0x00)	Colour 0%
100 (0x64)	Colour 100%
12 (0x0C)	Colour 12%

Return Value

1 if successful, otherwise the return value is -1.

5.4.2.7 GetRGBFirmwareVersion

int GetRGBFirmwareVersion ([in] int DevIdx, [out] int * Major, [out] int * Minor, [out] int * Build)

GetRGBFirmwareVersion gets the specified device firmware version.

Parameters

DevIdx

Index to the desire device currently available in the system. The first index is 1.

Major

Pointer to a variable that will contain the device major firmware version value.

Minor

Pointer to a variable that will contain the device minor firmware version value.

Build

Pointer to a variable that will contain the device build firmware version value.

Return Value

1 if successful, otherwise the return value is -1.

5.4.2.8 GetLibVersion

int GetLibVersion ([out] int * Major, [out] int * Minor)

GetLibVersion gets the currently used library version.

Parameters

Major

Pointer to a variable that will contain the library major version value.

Minor

Pointer to a variable that will contain the library minor version value.

Return Value

1 if successful, otherwise the return value is -1.



WERMA Signaltechnik GmbH + Co.KG
D-78604 Rietheim-Weilheim
Fon: +49 (0)7424 / 9557-222
Fax: +49 (0)7424 / 9557-44
support@werma.com
www.werma.com