

die zeichen lesen
die zeichen setzen

*reading the signs
setting the signs*



LINK882.DLL
1.0.0.0
ddm device driver

LINK882.DLL

DLL Driver
Reference Manual
Rev. 1.0 (Mai 2010)

ddm
hopt+schuler

D-78628 Rottweil
Königsberger Straße 12
Tel. (++49) 741 / 2607-0
Fax (++49) 741 / 1 33 98
ddm@hopt-schuler.com
www.hopt-schuler.com

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing.

No part of this document may be reproduced or transmitted in any form or by any means, for any purpose, without the prior written permission of “ddm hopt+schuler”.

ddm hopt+schuler shall have no liability for any errors or damages of any kind resulting from the use of this document.

History

Date	Rev	Note
10.05.2010	1.00	Initial draft

Table of Content

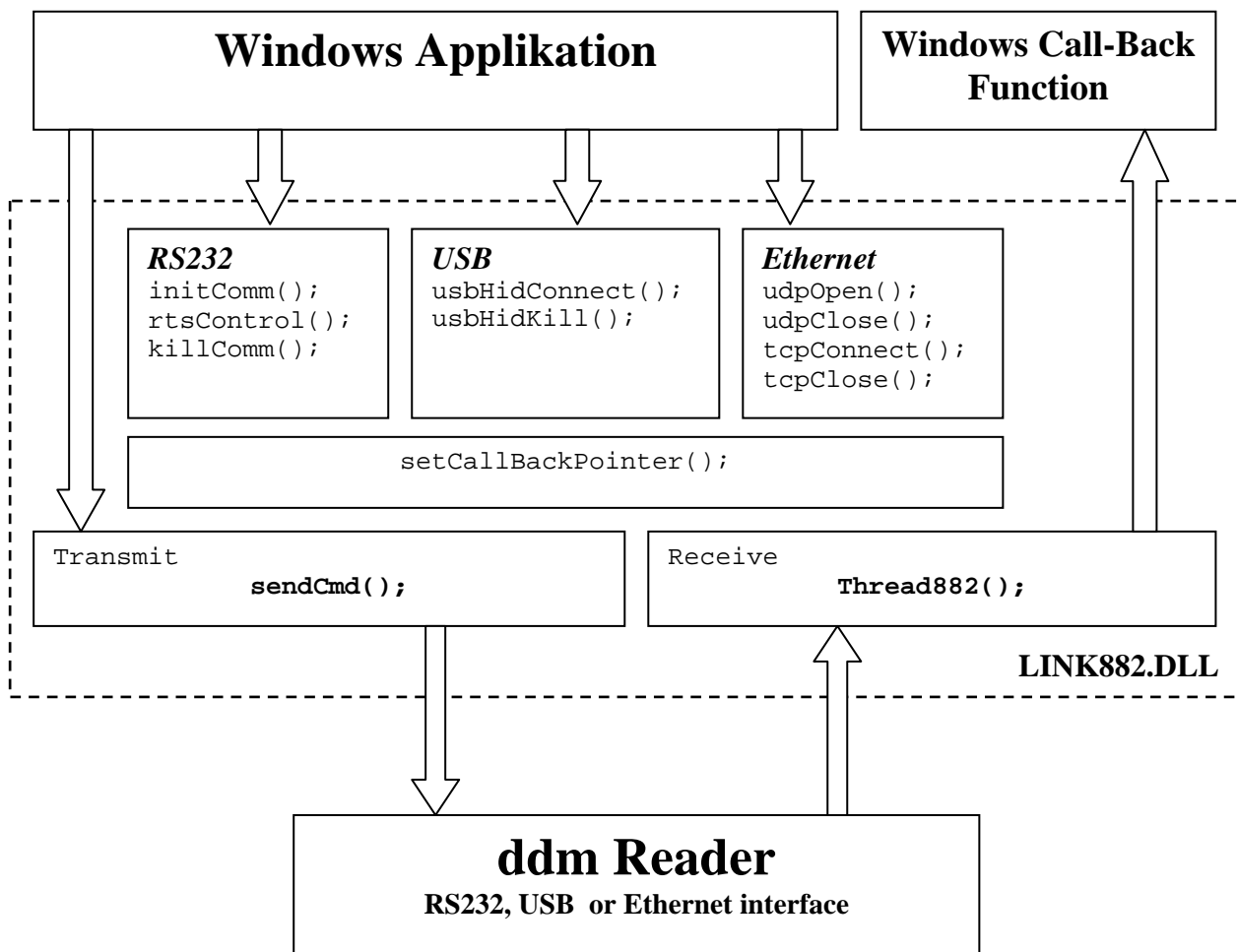
1	INTRODUCTION	4
2	FUNCTIONS	5
2.1	Serial Interface	5
2.1.1	initComm.....	5
2.1.2	killComm.....	5
2.1.3	rtsControl.....	5
2.2	USB Interface	5
2.2.1	usbHidConnect	5
2.2.2	usbHidKill	6
2.3	Ethernet Interface	6
2.3.1	tcpConnect.....	6
2.3.2	tcpClose.....	6
2.3.3	udpConnect.....	6
2.3.4	udpClose.....	7
2.4	Common Functions	7
2.4.1	getDLLVersion.....	7
2.4.2	setCallBackPointer	7
2.4.3	sendCmd.....	7

1 INTRODUCTION

LINK882.DLL is a common library for all ddm hopt+schuler readers. This library supports the following interfaces;

- RS232
- USB
- Ethernet (TCP/IP and UDP)

It has been developed with Visual C++ under “Visual Studio.NET 2003” environment.



Files supporting the development of a windows application;

Header: LINK882.h

Header: protCommon.h

C# Class: classLink882.cs

2 FUNCTIONS

2.1 Serial Interface

2.1.1 initComm

C, C++	<code>bool initComm(int port,long baud,long timeout);</code>
C#	<code>bool initComm(int port,long baud,long timeout);</code>
Description	This function initializes and opens the specified communication port.
Return value	Returns true or false
Parameters	Port: port number (1,2,3 ...) Baud: Baud rate (38400, 115200, ...) Timeout: Timeout in ms (1000, 2000, ...)

2.1.2 killComm

C, C++	<code>bool killComm(void);</code>
C#	<code>bool killComm();</code>
Description	This function closes the last opened comport.
Return value	Returns true or false
Parameters	No parameter

2.1.3 rtsControl

C, C++	<code>bool rtsControl(int rts);</code>
C#	<code>bool rtsControl(int rts);</code>
Description	This function sets the RTS pin of comport.
Return value	Returns true or false
Parameters	rts: 1 or 0

2.2 USB Interface

2.2.1 usbHidConnect

C, C++	<code>unsigned char usbHidConnect(unsigned char* name);</code>
C#	<code>byte usbHidConnect([Out()] byte[] name);</code>
Description	This function opens the USB interface.
Return value	Returns 0 or error code stat_USB_DEVICE_ERROR
Parameters	name: "DDM-866" or "DDM-882"

2.2.2 usbHidKill

C, C++	<code>void usbHidKill(void);</code>
C#	<code>void usbHidKill();</code>
Description	This function closes the USB interface.
Return value	
Parameters	

2.3 Ethernet Interface

2.3.1 tcpConnect

C, C++	<code>unsigned char tcpConnect(char *add, int port);</code>
C#	<code>byte tcpConnect([Out()] byte[] add, int port);</code>
Description	This function opens the TCP/IP interface.
Return value	Returns 0 or error code stat_ERROR
Parameters	add: IP address "172.20.2.170" port: Port number "default: 5000"

2.3.2 tcpClose

C, C++	<code>unsigned char tcpClose(void);</code>
C#	<code>byte tcpClose();</code>
Description	This function closes the TCP/IP interface.
Return value	Returns 0 or error code stat_ERROR
Parameters	

2.3.3 udpConnect

C, C++	<code>unsigned char udpOpen(int port, char * dest_addr, int listen_only);</code>
C#	<code>byte udpOpen(int port, [Out()] byte[] dest_addr, int listen_only);</code>
Description	This function opens the UDP interface.
Return value	Returns 0 or error code stat_ERROR
Parameters	port: Port number "default: 5000" dest_addr: IP address "172.20.2.170" listen_only: 0 for RX/TX, 1 for only RX

2.3.4 udpClose

C, C++	<code>unsigned char udpClose(void);</code>
C#	<code>byte udpClose();</code>
Description	This function closes the UDP interface.
Return value	Returns 0 or error code stat_ERROR
Parameters	

2.4 Common Functions

2.4.1 getDLLVersion

C, C++	<code>bool getDLLVersion(unsigned int *ver);</code>
C#	<code>bool getDLLVersion([Out()] byte[] ver);</code>
Description	This function gets the DLL version.
Return value	Returns true or false
Parameters	ver: DLL version (ex. 100 = V 1.00)

2.4.2 setCallBackPointer

C, C++	<code>bool setCallBackPointer(EventFunction evFunction);</code>
C#	<code>bool setCallBackPointer(EventFunction evFunction);</code>
Description	This function sets the call-back function pointer. In case of data receive, this call-back function will be called.
Return value	Returns true or false
Parameters	EventFunction(int addr,int status, IntPtr data, int len); addr: Network address status: Status byte data: Data pointer len: Data length

2.4.3 sendCmd

C, C++	<code>unsigned char sendCmd(unsigned char add, unsigned char cmd,unsigned char* snd,unsigned int len);</code>
C#	<code>byte sendCmd(int add, int cmd,[Out()] byte[] snd, int len);</code>
Description	This function sends the given command and data.
Return value	Returns 0 or error code
Parameters	add: Network address cmd: Command snd: Data to send len: Length of data