The library SysLibCom.lib

This library supports the serial communication with a target system. If the target system provides the functionality, then the following library functions can be used to open or close a serial port and to read or write data via this port. (The execution is synchronous.):

- SysComOpen
- SysComSetSettings
- SysComSetSettingsEx
- SysComClose
- SysComRead
- SysComWrite
- SysComGetVersion2300

SysComOpen

This function serves to open a serial port.

The function returns a handle for the port, which can be passed on when calling other functions of the library. If the port cannot be opened, 0xFFFFFFFF will be returned as handle.

Input Variable	Data Type	Description
Port	PORTS;	specifies the port which should be opened (COM1,..); Port number see below: Enumeration PORTS (see SysComSetSettings)

SysComSetSettings

This function serves to set values like baudrate, stopbits, parity, function-timeout, buffer-size and scan-time for a serial port. The parameter value is of type POINTER TO COMSETTINGS; the structure COMSETTINGS is used.

As soon as the parameters could be set successfully, TRUE will be returned, otherwise FALSE.

Input Variable	Data Type	Description
ComSettings	POINTER TO COMSETTINGS;	Pointer to the structure COMSETTINGS; you can make use of the operator **ADR** (see below, example)
dwHandle	DWORD	Port handle, acquired by SysComOpen

The structure COMSETTINGS, which is also part of the library, is defined as follows:

```
TYPE COMSETTINGS :
STRUCT
    Port:PORTS;          Port number, see below: Enumeration PORTS
    dwBaudRate:DWORD;    4800, 9600, 19200, 38400, 57600, 115200
```

byStopBits:BYTE;	0 = ONESTOPBIT, 1=ONE5STOPBITS, 2=TWOSTOPBITS
byParity:BYTE;	0 = NOPARITY, 1 = ODDPARITY, 2 = EVENPARITY
dwTimeout:DWORD;	Timeout of the interface in ms, Default = 0
dwBufferSize:DWORD;	Buffer size of the internal device buffer, Default = 0
dwScan:DWORD;	Polling time of the serial interface; should be set to 0

END_STRUCT
END_TYPE

Enumeration PORTS:

TYPE PORTS : (COM1:=1, COM2, COM3, COM4, COM5, COM6, COM7, COM8);
END_TYPE

SysComSetSettingsEx

The function with the parameters of type POINTER TO COMSETTINGSEX is used to set all relevant parameters of a serial communication port. Not only the parameters of the above function are set, but also the parameters for flowcontrol and character size can be set with this function. This is performed by filling them into the structure COMSETTINGSEX.

The return value of the function is true if the parameters were successfully set and false if the parameters could not be applied to the communication port. It is hardware-dependent whether the settings can be changed more often than one time after opening a port. It may be necessary to close and reopen the port before setting the parameters new.

Input-Variable	Data type	Description
ComSettingsEx	POINTER TO COMSETTINGSEX;	Pointer to the structure COMSETTINGSEX; Use the **ADR** operator to determine an address (see below for an example)
dwHandle	DWORD	Port handle, acquired by SysComOpen

Struktur **COMSETTINGSEX**:

TYPE COMSETTINGSEX :
STRUCT

Size:INT;	(*The size in bytes of the structure. Use the sizeof() operator to fill in. Used for backward compatibility.*)
Port:PORTS;	(*Port number, see below: Enumeration PORTS *)
dwBaudRate:DWORD;	(* 4800, 9600, 19200, 38400, 57600, 115200 *)
byStopBits:BYTE;	(* 0 = ONESTOPBIT, 1=ONE5STOPBITS, 2=TWOSTOPBITS *)
byParity:BYTE;	(* 0 = NOPARITY, 1 = ODDPARITY, 2 = EVENPARITY *)
dwTimeout:DWORD;	(* Timeout of the port in ms, Default = 0 *)
dwBufferSize:DWORD;	(* Buffersize used by the driver, Default = 0 *)
dwScan:DWORD;	(* Poll-time of the serial driver. Should be set to 0. Only change if the documentation of the hardware-vendor tells so. *)
cByteSize : BYTE;	(*4...8: Character size in bits.*)

```
    fOutxCtsFlow :  BOOL;   (*Specifies whether the CTS (clear-to-send) signal is monitored for output
                            flow control. If this member is TRUE and CTS is turned off, output is
                            suspended until CTS is sent again. *)
    fDtrControl : BYTE;     (*0:Disables the DTR line when the device is opened and leaves it
                            disabled.
                            1:Enables the DTR line when the device is opened and leaves it on.
                            2:Enables DTR handshaking. *)
    fDsrSensitivity : BOOL; (*Specifies whether the communications driver is sensitive to the state of
                            the DSR signal. If this member is TRUE, the driver ignores any bytes
                            received, unless the DSR modem input line is high. *)
    fRtsControl : BYTE;     (*0: Disables the RTS line when the device is opened and leaves it
                            disabled.
                            1: Enables the RTS line when the device is opened and leaves it on.
                            2: Enables RTS handshaking. The driver raises the RTS line when the
                            "type-ahead" (input) buffer is less than one-half full and lowers the RTS
                            line when the buffer is more than three-quarters full.
                            3: Specifies the RTS line will be high if bytes are available for
                            transmission. After all buffered bytes have been sent, the RTS line will be
                            low. *)
    fOutxDsrFlow : BOOL;    (*Specifies whether the DSR (data-set-ready) signal is monitored for
                            output flow control. If this member is TRUE and DSR is turned off, output
                            is suspended until DSR is sent again. *)
END_STRUCT
END_TYPE
```

Enumeration **PORTS**:

```
TYPE PORTS : (COM1:=1, COM2, COM3, COM4, COM5, COM6, COM7, COM8);
END_TYPE
```

Example for the settings to perform a hardwarehandshake:

```
dwHandle: DWORD;
pt_comsettingsex:COMSETTINGSEX:=(Port:=COM1,
                                 dwBaudRate:=38400,
                                 byStopBits:=0,
                                 dwTimeout:=5000,
                                 cByteSize:=7,
                                 byParity := 2,
                                 fOutxCtsFlow := FALSE,
                                 fOutxDsrFlow:=TRUE,
                                 DtrControl := 2,
                                 fRtsControl := 2);
```

Implementation:

```
pt_comsettingsex.Size := sizeof(pt_comsettingsex);
SysComSetSettingsEx(dwHandle := Handle, ComSettingsExt :=
ADR(pt_comsettingsex));
Where Handle is the returnvalue of a call to SysComOpen(COM1).
```

SysComClose

This function of type BOOL closes the COM port. For that purpose the port handle, which has been got by SysComOpen, must be given as input parameter. The return value will be TRUE after a successful operation, otherwise FALSE

Input Variable	Data Type	Description
dwHandle	DWORD	Port handle, acquired by SysComOpen

SysComWrite

This function of type DWORD writes the data to that port which is defined by the handle got by SysComOpen. Besides the handle also the address from which the data should be taken, the number of data which should be written and the timeout of the function must be passed on.

The function will return the number of actually written bytes.

Input Variable	Data type	Description
dwHandle	DWORD	Port handle, acquired by SysComOpen
dwBufferAddress	DWORD	Address from which the data should be taken and written to the port; you can use the **ADR** operator to get this address
dwBytesToWrite	DWORD	Number of bytes, which should be written
dwTimeout	DWORD	Time in [ms], after which the function will return at the latest

SysComRead

This function of type DWORD reads the data of COM-PORT. The input parameters are the port handle got by SysComOpen, the number of expected bytes and the timeout of the function. Besides that the address to which the read data should be copied, will be passed on.

The function will return the number of actually read bytes.

Input Variable	Data Type	Description
dwHandle	DWORD	Port handle, acquired by SysComOpen
dwBufferAddress	DWORD	address, to which the read bytes should be copied after having been read from the port; (you can make use of the operator ADR to get this address)
dwBytesToRead	DWORD	Number of bytes, which should be read
dwTimeout	DWORD	Time in [ms], after that the function returns at the latest

SysComGetVersion2300

This function of the library SysLibCom.lib (type DWORD, always returns 0) is only used for an automatic internal version check and is not to be called explicitly in the application program.