

Die Bibliothek SysLibCom.lib

Diese Bibliothek unterstützt die serielle Kommunikation mit einem Zielrechner. Wenn das Zielsystem die Funktionalität beinhaltet, können folgende Bibliotheksfunktionen verwendet werden, um einen seriellen Port zu öffnen und zu schließen, dessen Kommunikationsparameter zu setzen und Dateien über diesen Port zu schreiben oder zu lesen (die Abarbeitung erfolgt synchron):

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

SysComOpen

Diese Funktion dient dazu, einen seriellen Port zu öffnen.

Als Rückgabewert erhält man ein Handle auf den Port, das beim Aufruf der anderen Funktionen der Bibliothek übergeben wird. Wenn der Port nicht geöffnet werden kann, wird 0xFFFFFFFF als Handle zurückgegeben.

Input-Variable	Datentyp	Beschreibung
Port	PORTS;	spezifiziert die zu öffnende Schnittstelle (COM1,...); Port-Nummer, siehe SysComSetSettings: Enumeration PORTS

SysComSetSettings

Diese Funktion mit dem Übergabeparameter vom Typ POINTER TO COMSETTINGS dient dazu, für einen seriellen Port die Parameter wie Baudrate, Stopbits, Parity, Funktions-Timout, Buffer-Größe und Scan-Time zu setzen. Dies geschieht über die Struktur COMSETTINGS. Je nach verwendeter Steuerung kann man die Parameter einer Schnittstelle zur Laufzeit ändern, es muss aber einmal nach dem Öffnen der Schnittstelle gewährleistet sein, die Parameter zu übernehmen. Je nach verwendeter Steuerung kann es also notwendig sein, die Schnittstelle zu schließen und neu zu öffnen, um Parameter setzen zu können.

Als Rückgabewert erhält man ein TRUE, wenn die Parameter gesetzt werden konnten, ansonsten FALSE.

Input-Variable	Datentyp	Beschreibung
ComSettings	POINTER TO COMSETTINGS;	Zeiger auf die Struktur COMSETTINGS; der Operator ADR kann dabei zu Hilfe genommen werden (siehe unten, Beispiel)
dwHandle	DWORD	Der Rückgabewert der Funktion SysComOpen.

Struktur COMSETTINGS:

TYPE COMSETTINGS :
STRUCT

Port:PORTS; (* Port-Nummer, siehe unten: 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 der Schnittstelle in ms, Default = 0 *)
 dwBufferSize:DWORD; (* Puffergröße des internen Gerätepuffers, Default = 0 *)
 dwScan:DWORD; (* Poll-Zeit der seriellen Schnittstelle, sollte auf 0 gesetzt sein *)

END_STRUCT
END_TYPE

Enumeration PORTS:

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

Beispiel:

```
dwHandle: DWORD;
pt_comsettings: COMSETTINGS :=
  (port:=1,dwBaudRate:=9600,byStopBits:=1,byParity:=0,
  dwTimeout:=0,dwBufferSize:=8,dwScan:=0);
```

SysComSetSettingsEx

Diese Funktion mit dem Übergabeparameter vom Typ POINTER TO COMSETTINGSEX dient dazu, für einen seriellen Port alle Parameter (nicht nur Baudrate, Stopbits, Parity, Funktions-Timout, Buffer-Größe und Scan-Time, sondern auch Parameter wie Zeichengröße und die für den Hardwarehandshake relevanten Parameter DTR/RTS-Control, CTS/DSR Flusskontrolle und DSR-Sensitivität) zu setzen. Dies geschieht über die Struktur COMSETTINGSEX.

Als Rückgabewert erhält man ein TRUE, wenn die Parameter gesetzt werden konnten, ansonsten FALSE. Je nach verwendeter Steuerung kann man die Parameter einer Schnittstelle zur Laufzeit ändern, es muss aber einmal nach dem Öffnen der Schnittstelle gewährleistet sein, die Parameter zu übernehmen. Je nach verwendeter Steuerung kann es also notwendig sein, die Schnittstelle zu schließen und neu zu öffnen, um Parameter setzen zu können.

Input-Variable	Datentyp	Beschreibung
ComSettingsEx	POINTER TO COMSETTINGSEX;	Zeiger auf die Struktur COMSETTINGSEX; der Operator ADR kann dabei zu Hilfe genommen werden (siehe unten, Beispiel)
dwHandle	DWORD	Der Rückgabewert der Funktion SysComOpen.

Struktur COMSETTINGSEX:

TYPE COMSETTINGSEX :
STRUCT

Size:INT; (* Die Größe dieser Struktur in Bytes, mit dem Sizeof-Operator hier eintragen. Für Rückwärtskompatibilität.*)

Port:PORTS; (* Port-Nummer, siehe unten: 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 der Schnittstelle in ms, Default = 0 *)

dwBufferSize:DWORD; (* Puffergröße des internen Gerätepuffers, Default = 0 *)

dwScan:DWORD; (* Poll-Zeit der seriellen Schnittstelle, sollte auf 0 gesetzt sein *)

cByteSize : BYTE; (*4...8: Zeichengröße in Anzahl Datenbits.*)

fOutxCtsFlow : BOOL; (*Gibt an, ob das CTS-Signal für Ausgabekontrolle benutzt wird. Wenn dieser Member TRUE ist, wird die Ausgabe auf der seriellen Schnittstelle solange verzögert bis CTS wieder auf High geht. Hardware-Handshake. *)

fDtrControl : BYTE; (*0:Schaltet die DTR Leitung ab und lässt sie abgeschaltet, solange die Schnittstelle geöffnet ist.
1: Schaltet die DTR Leitung ein und lässt sie eingeschaltet, solange die Schnittstelle geöffnet ist.
2:Aktiviert das DTR Handshake. *)

fDsrSensitivity : BOOL; (*Gibt an ob der Kommunikationstreiber bei eingeschaltetem DSR ankommende Zeichen ignoriert. Ist dieser Member auf FALSE, wird der Status der DSR-Leitung ignoriert.*)

fRtsControl : BYTE; (* 0:Schaltet die RTS Leitung ab solange die Schnittstelle geöffnet ist.
1:Schaltet die RTS Leitung ein, solange die Schnittstelle geöffnet ist.
2:Aktiviert das RTS Handshake. Der Treiber aktiviert die RTS-Leitung, wenn der Eingangspuffer weniger als halb voll ist und deaktiviert die Leitung wenn der der Puffer mehr als dreiviertel voll ist.
3:RTS wird vom Treiber immer eingeschaltet, wenn noch Zeichen für die Übertragung anstehen. Wenn alle Zeichen gesendet wurden, wird die RTS Leitung abgeschaltet.*)

fOutxDsrFlow : BOOL; (*Gibt an, ob die DSR-Leitung als Sendesperre benutzt wird. Wenn dieser Member auf TRUE ist, wird nur bei eingeschalteter DSR-Leitung gesendet.*)

END_STRUCT
END_TYPE

Enumeration PORTS:

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

Beispiel für die Einstellungen mit HW-Handshake.

```
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));
Dabei ist Handle der Rückgabewert der Funktion SysComOpen(COM1).
```

SysComClose

Diese Funktion vom Typ BOOL schließt den COM-Port, dazu ist der bei SysComOpen erhaltene Port-Handle zu übergeben. Als Rückgabewert erhält man mit TRUE oder FALSE Auskunft über den Erfolg der Funktion.

Input-Variable	Datentyp	Beschreibung
dwHandle	DWORD	aus SysComOpen erhaltenes Handle für den Port

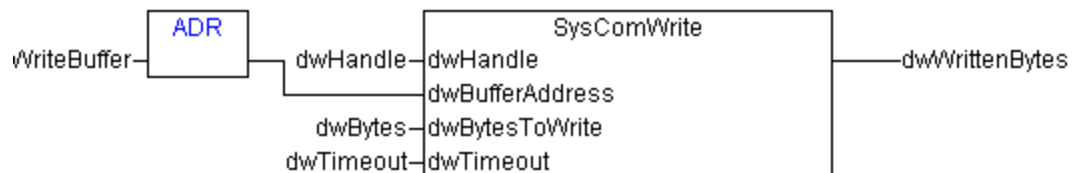
SysComWrite

Diese Funktion vom Typ DWORD schreibt die Daten an den mit dem aus SysComOpen erhaltenen Handle definierten Port. Außerdem wird die Adresse, aus der die Daten geschrieben werden sollen, die Anzahl der zu schreibenden Daten und der Timeout der Funktion übergeben. Als Rückgabewert wird die Anzahl der tatsächlich geschriebenen Bytes geliefert.

Input-Variable	Datentyp	Beschreibung
dwHandle	DWORD	aus SysOpenCom erhaltener Handle des Ports
dwBufferAddress	DWORD	Adresse, von der die Daten geholt werden sollen, um auf den Port geschrieben zu werden; kann mit Hilfe des Operators ADR ermittelt werden
dwBytesToWrite	DWORD	Anzahl der Bytes, die geschrieben werden sollen
dwTimeout	DWORD	Zeit in [ms], nach der die Funktion spätestens zurückkehrt

Beispiel:

```
dwHandle: DWORD;
WriteBuffer: ARRAY[0..5] OF BYTE:=0,1,2,4,5,6;
dwBytes: DWORD:=6;
dwTimeout: DWORD:=0;
dwWrittenBytes: DWORD;
```



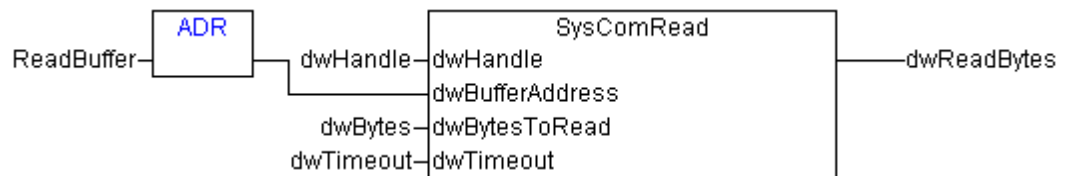
SysComRead

Diese Funktion vom Typ DWORD liest Daten vom COM-PORT. Die Eingangswerte sind der mit SysComOpen erhaltene Port-Handle, die Anzahl der erwarteten Bytes und die Timeout-Zeit der Funktion. Außerdem wird die Adresse, auf die die gelesenen Daten kopiert werden sollen, übergeben. Als Rückgabewert erhält man die Anzahl der tatsächlich gelesenen Bytes.

Input-Variable	Datentyp	Beschreibung
dwHandle	DWORD	aus SysOpenCom erhaltener Handle des Ports
dwBufferAddress	DWORD	Adresse, auf die die gelesenen Bytes kopiert werden sollen, nachdem sie vom Port gelesen wurden; kann mit Hilfe des Operators ADR ermittelt werden
dwBytesToRead	DWORD	Anzahl der Bytes, die gelesen werden sollen
dwTimeout	DWORD	Zeit in [ms], nach der die Funktion spätestens zurückkehrt

Beispiel:

```
dwHandle: DWORD;
ReadBuffer: ARRAY[0..5] OF BYTE;
dwBytes: DWORD:=6;
dwTimeout: DWORD:=0;
dwReadBytes: DWORD;
```



SysComGetVersion2300

Diese Funktion der Bibliothek SysLibCom.lib (Typ DWORD, Rückgabewert immer 0) dient ausschließlich einem automatischen internen Versionsabgleich und muss vom Anwender nicht aufgerufen werden.