

WAGO-I/O-PRO 32 Bibliothek

GSM_SMS_01.lib

Überblick

Diese Bibliothek enthält einen Baustein zum SMS-Versand und Empfang über ein GSM-Modem.

Es wird der PDU-Mode des Modems unterstützt.

Weiterhin enthält diese Bibliothek einen Baustein zum Parsen eines Strings.

Durch die Kombination dieser beiden Bausteine kann ein einfaches RPC-Interface in einem programmierbaren Controller realisiert werden.

Inhalt

GSM_SMS	3
PARSE_SMS_MESSAGE	7
typGSM_RECEIVE_SMS (Datentyp).....	8
typGSM_SEND_SMS (Datentyp).....	9
typGSM_NUMBER (Datentyp)	10
typGSM_TIME_STAMP (Datentyp)	11
GSM_ERROR (Enum)	12
typMSG_PARSE (Datentyp).....	13
typGSM_BUFFER (Datentyp intern)	14
typGSM_PDU (Datentyp intern)	15

GSM_SMS

WAGO-I/O-PRO 32 Elemente der Bibliothek		
Kategorie:	Kommunikationsbausteine	
Name:	GSM_SMS	
Typ:	Funktion <input type="checkbox"/>	Funktionsblock <input checked="" type="checkbox"/> Programm <input type="checkbox"/>
Name der Bibliothek:	GSM_SMS_01.lib	
Verwendete Bibliotheken		
Anwendbar für:	Programmierbare Koppler (nicht für 750-812/814/815/816)	
Eingangsparameter:	Datentyp:	Kommentar:
xLISTEN	BOOL	Durch ein TRUE an diesem Eingangsparameter wird der Treiber in den Zustand "Listen" versetzt. Nur im Zustand "Listen" können Nachrichten empfangen werden.
bCOM_PORT_NR	BYTE	Nr. der verwendeten ser. Schnittstelle 1 -> interne Serviceschnittstelle 2 -> 1.gestecktes Modul 750-65x 3 -> 2.gestecktes Modul 750-65x u.s.w.
cbBAUDRATE	COM_BAUDRATE	je nach benutzter Schnittstelle siehe SerComm.lib
cpPARITY	COM_PARITY	je nach benutzter Schnittstelle siehe SerComm.lib
csSTOPBITS	COM_STOPBITS	je nach benutzter Schnittstelle siehe SerComm.lib
cfFLOW_CONTROL	COM_FLOW_CONTROL	je nach benutzter Schnittstelle siehe SerComm.lib
GSM_MODEM_INT	STRING	Initialisierungsstring für das verwendete Modem. Dieser String wird beim Verbindungsaufbau als erstes an das Modem gesendet. Folgende Zeichen in diesem String haben eine Sonderbedeutung : / => 1 sec Pause das Zeichen / wird nicht gesendet. => wird durch Return (0x0D) ersetzt
xDELETE_AFTER_READ	BOOL	Bei einem TRUE an diesem Eingang werden eingehende Nachrichten nach dem Empfang im Treiber vom Treiber aus dem Speicher des GSM-Modems automatisch gelöscht und somit ein Speicherüberlauf im Modem verhindert.
tACK_TIME_OUT	TIME	max. erlaubte Zeit zum Übermitteln einer Nachricht an das SMSC. Wenn innerhalb dieser Zeit keine Übernahmequittung vom SMSC gesendet wird, dann wird die Übertragung abgebrochen.

Ein-/Ausgangsparameter:	Datentyp:	Kommentar:
xNEW_SMS_RECEIVED	BOOL	Dieser Durchgangsparameter wird vom Treiber autom. nach jedem Eintreffen einer Nachricht gesetzt. Ein Anwenderprogramm sollte diesen Parameter auf TRUE prüfen und anschl. löschen um bei einem erneuten TRUE eine neue Nachricht erkennen zu können.
utSMS_MESSAGE	typGSM_SEND_SMS	Struktur in die die Daten einer zu sendenden Nachricht eingetragen werden können. Die Struktur (Msg) enthält folgende Elemente (siehe auch Strukturbeschreibung) : Msg. SMSC_NO. International Msg. SMSC_NO. Number Msg. Dest_No. International Msg. Dest_No. Number Msg. Msg Ist das Element Msg. SMSC_NO. Number bei einem Sendeauftrag leer, dann wird das im GSM-Modem gespeicherte SMSC verwendet.
xSEND_SMS	BOOL	Durch einmaliges setzen dieses Parameters wird ein Sendeauftrag angestossen. Nach Beendigung des Auftrages wird dieser Parameter autom. vom Treiber gelöscht.

Ausgangsparameter:	Datentyp:	Kommentar:																								
utRECEIVED_SMS	typGSM_RECEIVE_SMS	In dieser Struktur wird eine eintreffende Nachricht eingetragen. Diese Struktur enthält neben der eigentlichen Nachricht auch den Zeitstempel sowie die Absendernummer und die Nummer des verwendeten SMSC.																								
RECEIVE_ERROR	GSM_ERROR	GSM_NO_ERROR := 0, GSM_PARSE_ERROR := 1, GSM_SIO_TIMEOUT_ERROR := 2, GSM_DATA_TIMEOUT_ERROR := 3, GSM_DATA_VALID_ERROR := 4, GSM_DATA_PDU_ERROR := 5, GSM_ACK_TIMEOUT_ERROR := 6, GSM_SEND_ERROR := 7																								
SEND_ERROR	GSM_ERROR	GSM_NO_ERROR := 0, GSM_PARSE_ERROR := 1, GSM_SIO_TIMEOUT_ERROR := 2, GSM_DATA_TIMEOUT_ERROR := 3, GSM_DATA_VALID_ERROR := 4, GSM_DATA_PDU_ERROR := 5, GSM_ACK_TIMEOUT_ERROR := 6, GSM_SEND_ERROR := 7																								
STATUS	STRING	Dieser Parameter enthält nach einem Sendevorgang den vom SMSC gesendeten Quittierungsstring.																								
Grafische Darstellung:																										
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;"> <p style="text-align: center;">GSM_SMS</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px solid black; padding-right: 5px;">xLISTEN</td> <td style="width: 50%; padding-left: 5px;">utRECEIVED_SMS</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">bCOM_PORT_NR</td> <td style="padding-left: 5px;">RECEIVE_ERROR</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">cbBAUDRATE</td> <td style="padding-left: 5px;">SEND_ERROR</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">cpPARITY</td> <td style="padding-left: 5px;">STATUS</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">csSTOPBITS</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">cfFLOW_CONTROL</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">GSM_MODEM_INIT</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">xDELETE_AFTER_READ</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">tACK_TIME_OUT</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">xNEW_SMS_RECEIVED</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">utSMS_MESSAGE</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">xSEND_SMS</td> <td></td> </tr> </table> </div>			xLISTEN	utRECEIVED_SMS	bCOM_PORT_NR	RECEIVE_ERROR	cbBAUDRATE	SEND_ERROR	cpPARITY	STATUS	csSTOPBITS		cfFLOW_CONTROL		GSM_MODEM_INIT		xDELETE_AFTER_READ		tACK_TIME_OUT		xNEW_SMS_RECEIVED		utSMS_MESSAGE		xSEND_SMS	
xLISTEN	utRECEIVED_SMS																									
bCOM_PORT_NR	RECEIVE_ERROR																									
cbBAUDRATE	SEND_ERROR																									
cpPARITY	STATUS																									
csSTOPBITS																										
cfFLOW_CONTROL																										
GSM_MODEM_INIT																										
xDELETE_AFTER_READ																										
tACK_TIME_OUT																										
xNEW_SMS_RECEIVED																										
utSMS_MESSAGE																										
xSEND_SMS																										

Funktionsbeschreibung:

Dieser Baustein ermöglicht sowohl den Empfang als auch das Versenden von Kurzmitteilungen.

Nachrichteneingang:

Durch setzen des Parameters xLISTEN auf TRUE wird die parametrisierte Schnittstelle geöffnet, der Initialisierungsstring an das GSM-Modem gesendet und der Treiber im fehlerfreien Fall in den Zustand „Listen“ versetzt.

Ankommende Nachrichten werden nun vom Treiber in der Ausgangsstruktur utRECEIVED_SMS eingetragen und der Durchgangsparameter xNEW_SMS_RECEIVED gesetzt.

Neue Nachrichten überschreiben alte Nachrichten sofort nach dem Eintreffen.

Der Durchgangsparameter xNEW_SMS_RECEIVED sollte daher vom Anwender abgefragt werden und bei einem TRUE die eingetroffene Nachricht sofort vom Anwender in einen eigenen Buffer übernommen werden. Anschließend sollte xNEW_SMS_RECEIVED gelöscht werden um das neuerliche Eintreffen einer Nachricht zu erkennen.

Nachrichtenversand:

Zunächst muß die Struktur utSMS_MESSAGE mit den zu sendenden Informationen gefüllt werden.

Anschließend kann durch einmaliges setzen des Parameters xSEND_SMS der Sendevorgang ausgelöst werden.

Das GSM-Modem wird nun versuchen eine Verbindung zum SMSC aufzubauen und die Nachricht zu übertragen.

Ist das Element *Msg*. SMSC_NO. Number leer, so wird die im Modem gespeicherte SMSC-Nummer verwendet.

Nach Beendigung des Sendevorganges wird der Parameter xSEND_SMS automatisch vom Treiber gelöscht.

Der Sendevorgang wird durch die Zeit tACK_TIME_OUT überwacht. Sollte nach der Überwachungszeit keine Quittung vom SMSC eingetroffen sein, dann wird der Sendeauftrag abgebrochen.

Es sollte daher nach einem Sendeauftrag der Parameter SEND_ERROR geprüft werden.

Der Baustein muß vom Anwender zyklisch aufgerufen werden.

PARSE_SMS_MESSAGE

WAGO-I/O-PRO 32 Elemente der Bibliothek		
Kategorie:	Kommunikationsbausteine	
Name:	PARSE_SMS_MESSAGE	
Typ:	Funktion <input checked="" type="checkbox"/>	Funktionsblock <input type="checkbox"/> Programm <input type="checkbox"/>
Name der Bibliothek:	GSM_SMS_01.lib	
Verwendete Bibliotheken		
Anwendbar für:	Programmierbare Koppler	
Eingangsparameter:	Datentyp:	Kommentar:
MSG	STRING(GSM_SMS_MSG_MAX_LENGTH)	Nachricht die zerlegt werden soll
TRENNER	BYTE := 16#2C	Trennzeichen zwischen den Komponenten (default 16#2C => Komma).
Datentyp Rückgabewert:	Kommentar:	
typMSG_PARSE	Struktur in der der zerlegte Eingangsstring abgelegt wird.	
Grafische Darstellung:		
		
Funktionsbeschreibung:		
<p>Diese Funktion zerlegt einen String in einzelne Komponenten. Die einzelnen Komponenten müssen im String durch das Zeichen das mit TRENNER vorgegeben wird, getrennt sein. Der zerlegte String wird in einer Struktur vom Typ typMSG_PARSE zurückgegeben. ASCII-Zeichen werden als Großbuchstaben in der Rückgabestruktur abgelegt (UPPERCASE).</p>		

typGSM_RECEIVE_SMS (Datentyp)

WAGO-I/O-PRO 32 Elemente der Bibliothek		
Kategorie:		
Name:	typGSM_RECEIVE_SMS	
Typ:	Datentyp <input checked="" type="checkbox"/>	Enumeration <input type="checkbox"/>
Name der Bibliothek:	GSM_SMS_01.lib	
Struktur:		
Element:	Datentyp:	Beschreibung:
Valid	BOOL	Ein TRUE zeigt an das diese Nachricht gültig ist.
SMSC_No	typGSM_NUMBER	Nummer des SMSC über das diese Nachricht vesendet wurde,
Sender_No	typGSM_NUMBER	Nummer des Absenders
Prot_ID	BYTE	
Data_Coding	BYTE	
Time_Stamp	typGSM_TIME_STAMP	Zeitstempel des SMSC
Msg	STRING(GSM_SMS_MSG_MAX_LENGTH)	Nachrichtentext
Deklaration:		
<pre> TYPE typGSM_RECEIVE_SMS : STRUCT Valid : BOOL; SMSC_No : typGSM_NUMBER; Sender_No : typGSM_NUMBER; Prot_ID : BYTE; Data_Coding : BYTE; Time_Stamp : typGSM_TIME_STAMP; Msg : STRING (GSM_SMS_MSG_MAX_LENGTH) ; END_STRUCT END_TYPE </pre>		
Datentypbeschreibung: Dieser Datentyp beschreibt den Aufbau einer eintreffenden Nachricht.		

typGSM_SEND_SMS (Datentyp)

WAGO-I/O-PRO 32 Elemente der Bibliothek		
Kategorie:		
Name:	typGSM_SEND_SMS	
Typ:	Datentyp <input checked="" type="checkbox"/>	Enumeration <input type="checkbox"/>
Name der Bibliothek:	GSM_SMS_01.lib	
Struktur:		
Element:	Datentyp:	Beschreibung:
SMSC_No	typGSM_NUMBER	Nummer des SMSC über das diese Nachricht versendet werden soll.
Dest_No	typGSM_NUMBER	Nummer des Empfängers
Msg	STRING(GSM_SMS_MSG_MAX_LENGTH)	Nachrichtentext
Deklaration:		
<pre> TYPE typGSM_SEND_SMS : STRUCT SMSC_No : typGSM_NUMBER; Dest_No : typGSM_NUMBER; Msg : STRING(GSM_SMS_MSG_MAX_LENGTH); END_STRUCT END_TYPE </pre>		
Datentypbeschreibung: Dieser Datentyp beschreibt den Aufbau einer zu versendenden Nachricht.		

typGSM_NUMBER (Datentyp)

WAGO-I/O-PRO 32 Elemente der Bibliothek		
Kategorie:		
Name:	typGSM_NUMBER	
Typ:	Datentyp <input checked="" type="checkbox"/>	Enumeration <input type="checkbox"/>
Name der Bibliothek:	GSM_SMS_01.lib	
Struktur:		
Element:	Datentyp:	Beschreibung:
International	BOOL	Gibt an ob die Nummer im internationalen Format zu interpretieren ist, also z.B. mit 49 für Deutschland beginnt.
Number	STRING(GSM_PHONE_NO_MAX_LENGTH)	Telefonnummer
Deklaration:		
<pre> TYPE typGSM_NUMBER : STRUCT International : BOOL := FALSE; Number : STRING(GSM_PHONE_NO_MAX_LENGTH) := ''; END_STRUCT END_TYPE </pre>		
Datentypbeschreibung: Dieser Datentyp beschreibt den Aufbau einer Telefonnummer für den SMS-Versand.		

typGSM_TIME_STAMP (Datentyp)

WAGO-I/O-PRO 32 Elemente der Bibliothek		
Kategorie:		
Name:	typGSM_NUMBER	
Typ:	Datentyp <input checked="" type="checkbox"/>	Enumeration <input type="checkbox"/>
Name der Bibliothek:	GSM_SMS_01.lib	
Struktur:		
Element:	Datentyp:	Beschreibung:
Year	BYTE	Jahr BCD-Codiert
Month	BYTE	Monat BCD-Codiert
Day	BYTE	Tag BCD-Codiert
Hour	BYTE	Stunde BCD-Codiert
Minute	BYTE	Minute BCD-Codiert
Timezone	BYTE	Zeitzone 15min / digit
Deklaration:		
<pre> TYPE typGSM_TIME_STAMP : STRUCT Year : BYTE; Month : BYTE; Day : BYTE; Hour : BYTE; Minute : BYTE; Second : BYTE; Timezone : BYTE; END_STRUCT END_TYPE </pre>		

GSM_ERROR (Enum)

WAGO-I/O-PRO 32 Elemente der Bibliothek		
Kategorie:		
Name:	GSM_ERROR	
Typ:	Datentyp <input type="checkbox"/>	Enumeration <input checked="" type="checkbox"/>
Name der Bibliothek:	GSM_SMS_01.lib	
Deklaration:		
<pre> TYPE GSM_ERROR : (GSM_NO_ERROR := 0, GSM_PARSE_ERROR := 1, GSM_SIO_TIMEOUT_ERROR := 2, GSM_DATA_TIMEOUT_ERROR := 3, GSM_DATA_VALID_ERROR := 4, GSM_DATA_PDU_ERROR := 5, GSM_ACK_TIMEOUT_ERROR := 6, GSM_SEND_ERROR := 7); END_TYPE </pre>		

typMSG_PARSE (Datentyp)

WAGO-I/O-PRO 32 Elemente der Bibliothek		
Kategorie:		
Name:	typMSG_PARSE	
Typ:	Datentyp <input checked="" type="checkbox"/>	Enumeration <input type="checkbox"/>
Name der Bibliothek:	GSM_SMS_01.lib	
Struktur:		
Element:	Datentyp:	Beschreibung:
Count	INT	Anzahl gültiger Parameter in der Struktur
Parameter	ARRAY[1..GSM_PARSE_MAX_PARAMETER] OF STRING	Array mit den Einzelkomponenten
Deklaration:		
<pre> TYPE typMSG_PARSE : STRUCT Count : INT; Parameter : ARRAY[1..GSM_PARSE_MAX_PARAMETER] OF STRING; END_STRUCT END_TYPE </pre>		

typGSM_BUFFER (Datentyp intern)

WAGO-I/O-PRO 32 Elemente der Bibliothek		
Kategorie:		
Name:	typGSM_BUFFER	
Typ:	Datentyp <input checked="" type="checkbox"/>	Enumeration <input type="checkbox"/>
Name der Bibliothek:	GSM_SMS_01.lib	
Struktur:		
Element:	Datentyp:	Beschreibung:
Index	INT	Länge der aktuellen Response
Data	ARRAY[0..GSM_RESPONSE_MAX_LENGTH] OF BYTE	Empfangsbuffer für die Response
Deklaration:		
<pre> TYPE typGSM_BUFFER : STRUCT Index : INT; Data : ARRAY[0..GSM_RESPONSE_MAX_LENGTH] OF BYTE; END_STRUCT END_TYPE </pre>		
Datentypbeschreibung: Dieser Datentyp wird nur Bausteinintern verwendet		

typGSM_PDU (Datentyp intern)

WAGO-I/O-PRO 32 Elemente der Bibliothek			
Kategorie:			
Name:	typGSM_PDU		
Typ:	Datentyp <input checked="" type="checkbox"/>	Enumeration <input type="checkbox"/>	
Name der Bibliothek:	GSM_SMS_01.lib		
Deklaration:			
<pre> TYPE typGSM_PDU : STRUCT Length : BYTE; strLenBody : STRING(10); Data : ARRAY[0..GSM_PDU_MSG_MAX_LENGTH] OF BYTE; END_STRUCT END_TYPE </pre>			
Datentypbeschreibung: Dieser Datentyp beschreibt den Aufbau einer PDU. Dieser Datentyp wird nur Bausteinintern verwendet.			

