WAGO-I/O-*PRO* 32 Libraries

Serial_Interface_01.lib

The library “Serial_Interface_01.lib” contains the function block SERIAL_INTERFACE. This function block allows sending and receiving of data using a WAGO serial communication module, 750-65x. The function block supports transparent data communication. The library is applicable for all controllers, except 750-812/814/815/816.

Content

Serial_Interface_01.lib 3

SERIAL_INTERFACE.......... 3
typRING_BUFFER (Data type) 8

Serial_Interface_01.lib

SERIAL_INTERFACE

WAGO-I/O-*PRO* 32 Library Elements			
Category:	Communication		
Name:	SERIAL_INTERFACE		
Type:	Function	Function block **X**	Program
Library name:	Serial_Interface_01.lib		
Used libraries:	SerComm.Lib		
Applicable to:	Controller 750-8xx (except 750-812/814/815/816)		

Input parameter:	**Data type:**	**Comments:**
xOPEN_COM_PORT	BOOL	Set True to open the comm port
bCOM_PORT_NR	BYTE	Port number of serial interface module 16#01 => Service Port 16#02 => 1. Module 750-65X 16#03 => 2. Module 750-65X etc. (up to 64 750-65X modules)

cbBAUDRATE	COM_ BAUDRATE	(*--- NOTE !! ----------------------------------*)
		not all baudrates are supported by all modules
		see the manual of your used interface for supported baudrates
		
		BAUD_110 := 11,
		BAUD_300 := 30,
		BAUD_600 := 60,
		BAUD_1200 := 120,
		BAUD_2400 := 240,
		BAUD_4800 := 480,
		BAUD_9600 := 960,
		BAUD_12000 := 1200,
		BAUD_14400 := 1440,
		BAUD_19200 := 1920,
		BAUD_28800 := 2880,
		BAUD_38400 := 3840,
		BAUD_57600 := 5760,
		BAUD_62500 := 6250,
		BAUD_115200 := 11520,
		
		BAUDRATE_TERMINAL_DEFAULT := 16#7FFF

cbsBYTESIZE	COM_
	BYTESIZE
		BS_7 :=7,
		BS_8 :=8,
		BYTESIZE_TERMINAL_DEFAULT := 16#7FFF

WAGO-I/O-*PRO* 32 Library Elements		

cpPARITY	COM_
	PARITY
		PARITY_NO:=0,
		PARITY_ODD:=1,
		PARITY_EVEN:=2,
		PARITY_TERMINAL_DEFAULT := 16#7FFF

csSTOPBITS	COM_
	STOPBITS
		STOPBITS_1 := 1,
		STOPBITS_2 := 2,
		STOPBITS_TERMINAL_DEFAULT := 16#7FFF

cfFLOW_CONTROL	COM_ FLOW_CONTROL	(*--- NOTE !! ------------------------------------*) not all settings are supported by all modules see the manual of your used interface for supported settings :::::: WITH CONTINOUS SEND :::::::::::: NO_FLOW_CONTROL :=0, (* RS 232 / fullduplex / without handshake / with continous send *) XON_XOFF :=1, (* RS 285 / halfduplex / with XON & XOFF / with continous send *) RTS_CTS :=2, (* RS 232 / fullduplex / with hardware handshake / with continous send *) FULLDUPLEX :=3, (* RS 422 / fullduplex / without handshake / with continous send *) HALFDUPLEX :=4, (* RS 485 / halfduplex / without handshake / with continous send *) RS232_FULLDUPLEX_XON_XOFF := 5, (* RS 232 / fullduplex / with XON & XOFF / with continous send *) RS422_FULLDUPLEX_XON_XOFF := 6, (* RS 422 / fullduplex / with XON & XOFF / with continous send *)

WAGO-I/O-*PRO* 32 Library Elements		

		(*:::::: WITHOUT CONTINOUS SEND ::::::
		NCS_NO_FLOW_CONTROL :=20, (* RS 232 / fullduplex / without handshake / without continous send *)
		NCS_XON_XOFF :=21, (* RS 285 / halfduplex / with XON & XOFF / without continous send *)
		NCS_RTS_CTS :=22, (* RS 232 / fullduplex / with hardware handshake / without continous send *)
		NCS_FULLDUPLEX :=23, (* RS 422 / fullduplex / without handshake / without continous send *)
		NCS_HALFDUPLEX :=24, (* RS 485 / halfduplex / without handshake / without continous send *)
		NCS_RS232_FULLDUPLEX_XON_XOFF := 25 (* RS 232 / fullduplex / with XON & XOFF / without continous send *)
		NCS_RS422_FULLDUPLEX_XON_XOFF := 26 (* RS 422 / fullduplex / with XON & XOFF / without continous send *)
		FLOW_CONTROL_TERMINAL_DEFAULT := 16#7FFF (* take the settings inside module *)

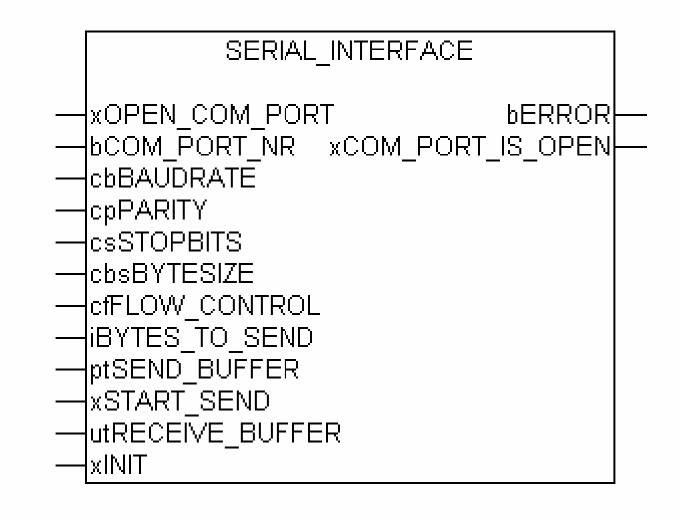
iBYTES_TO_SEND	INT	Number of bytes to be transmitted
ptSEND_BUFFER	POINTER TO BYTE	Pointer to the transmitter buffer

In/Output parameter:	**Data type:**	**Comments:**
xSTART_SEND	BOOL	Activates transmitting of data The function block will reset the parameter after transmission.

utRECEIVE_BUFFER	typRING_BUFFER	TYPE typRING_BUFFER :
		STRUCT
		Index : INT;
		Data : ARRAY[0..255] OF BYTE;
		END_STRUCT
		END_TYPE

xINIT	BOOL	
Output parameter:	**Data type:**	**Comments:**
bERROR	BYTE	0x00: No error. 0x01: This library is not supported by the firmware. 0x02: COM port outside of the valid area. 0x03: This function block instance has not yet been assigned a COM port. 0x04: This function block instance has already been assigned a different COM port. 0x05: COM port is already opened. 0x06: COM port is already closed. 0x07: COM port is not opened. 0x08: A write operation is still active (COM1). 0x09: These transfer parameters are not supported by the COM port. 0x0A: The current settings of the bus module could not be read. 0x0B: This library version does not yet support the temporary setting of communication parameters. 0x0C: Bus module could not be initialized. 0x0D: Error when writing data into the FIFO memory of the bus module. 0x0E: The contents of the FIFO memory was not sent (continuous sending). 0x0F: Internal error

Graphic display:



Function description:

Call the function block within each PLC cycle to ensure communication performance.

If using serial communication module 750-65x/003-000 or 750-652, the communication parameters are configurable by the user. The following parameters of this module are configurable:

-cbCOM_BAUDRATE
-cbsCOM_BYTESIZE
-csCOM_STOPBITS
-cpCOM_PARITY
-cfCOM_FLOW_CONTROL

When using any other serial communication modules (750-65x/000-000), the module will have a factory preset configuration that cannot be defined by the user. When using these modules, the parameters specified above are not to be used, or should only be used with the default values. Incorrect usage of these parameters will result in the following error: “bError := 0x09: Not supported parameterset”.

The function block supports transparent data transmission. The driver does not make a protocol available.

The controller detects and assigns the port numbers of the serial modules independently beginning from the left with COM2. The controller service interface is always COM1.

Sending data:
Assign the data source as well as the number of Bytes to be transmitted to the function block. Set the signal xSTART_SEND . The function block will reset xSTART_SEND after the execution.

Receiving data:
Received data will be displayed in the “utReceiveBuffer.data” array. utReceiveBuffer.Index is a pointer that will be incremented with each received character.

typRING_BUFFER (Data type)

WAGO-I/O-*PRO* 32 Library elements			
Category:			
Name:	typRING_BUFFER		
Type:	Data type **X**	Enumeration	
Library name:	Serial_Interface_01.lib		

Structure:

Elements:	**Data type:**	**Description:**
Index	INT	Indicates number of received characters. Points to the place in **Data** where the next received character will be stored.
Data	ARRAY[0..255] OF BYTE	Received data

Header:

```
TYPE typRING_BUFFER :
  STRUCT
      Index        :     INT;
      Data         :     ARRAY[0..255] OF BYTE;
  END_STRUCT
END_TYPE
```

Data type description: