

Manual | EN

TX1200

TwinCAT 2 | PLC Library: TcModbusTCPBC



Table of contents

1 Foreword	5
1.1 Notes on the documentation	5
1.2 Safety instructions	6
1.3 Notes on information security.....	7
2 Overview	8
3 Function blocks	9
3.1 FB_ModbusTcpOpen	9
3.2 FB_ModbusTcpClose.....	10
3.3 FB_ModbusTcpRequest	12
4 Appendix	14
4.1 BC9xxx return codes	14

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.



EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.



Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our <https://www.beckhoff.com/secguide>.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at <https://www.beckhoff.com/secinfo>.

2 Overview

The library contains function blocks that permit the exchange of data between the **BC9xxx** Bus Controller and a remote partner on the MODBUS/TCP port.

Comments

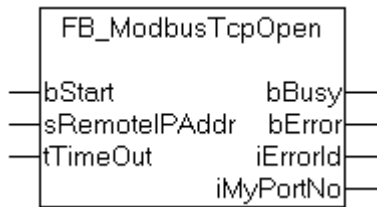
- The number of IP connections that may be open simultaneously is restricted to four, in order to maintain the use of resources at a minimum. If it is necessary to establish a further connection, a connection that is not required must first be closed.
- After about 10 seconds, a connection that has not been used during that period is automatically closed, and must be reopened if needed.

Function blocks

Name	Description
FB_ModbusTcpOpen [► 9]	Open a MODBUS/TCP connection to a remote partner.
FB_ModbusTcpClose [► 10]	Close a MODBUS/TCP connection to a remote partner.
FB_ModbusTcpRequest [► 12]	Send a MODBUS request to a remote partner.

3 Function blocks

3.1 FB_ModbusTcpOpen



The function block opens a TCP/IP connection to a remote partner on the MODBUS/TCP port. To keep resource consumption to a minimum, the maximum number of open connections has been limited to 4. If it is necessary to establish a further connection, a connection that is not currently in use can first be closed using the [FB_ModbusTcpClose \[► 10\]](#) function block. If successful, the FB_ModbusTcpOpen function block supplies a port number in the *iMyPortNo* output variable. This port number is required in order to send a MODBUS request using either of the [FB_ModbusTcpRequest \[► 12\]](#) or [FB_ModbusTcpClose \[► 10\]](#) function blocks, e.g. to close the connection. When opening the connection, the *tTimeout* input parameter is also used to specify the time-out period for the MODBUS request and for the disconnection. After about 10 seconds, a connection that has been opened but has not been used during that period is automatically closed, and must be reopened if needed.

VAR_INPUT

```
VAR_INPUT
    bStart      : BOOL;
    sRemoteIPAddr: STRING(15);
    tTimeOut    : TIME;
END_VAR
```

bStart: the function block is activated by a rising edge at this input.

sRemotelIPAddr: this variable is a string containing the network address (IP address) of the remote partner.

tTimeOut: indicates the time, which must not be exceeded during command execution.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    iErrorId   : WORD;
    iMyPortNo  : WORD;
END_VAR
```

bBusy: if the function block is activated, this output is set. It remains set until a feedback is received.

bError: if an error should occur during the execution of the command, then this output is set, after the bBusy output has been reset.

iErrorId: supplies the command-specific error code (table) or the [device-specific error code \[► 14\]](#) if the bError output is set.

iMyPortNo : when successful, the port number associated with the connection is returned at this output. The port number is a dynamic number, generated afresh each time the connection is established.

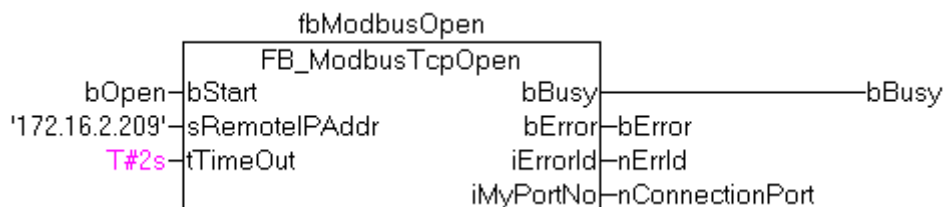
Error Codes	Description
0x8500	Resource error
0x8501	No socket available
0x8502	Open connection error

Sample of calling the function block in FBD:

```

PROGRAM MAIN
VAR
  fbModbusOpen   : FB_ModbusTcpOpen;
  bOpen          : BOOL;
  bBusy         : BOOL;
  bError        : BOOL;
  nErrId       : WORD;
  nConnectionPort: WORD;
END_VAR

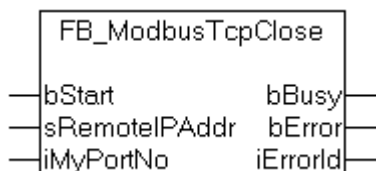
```



In this sample, a rising edge at the *bOpen* input causes a TCP/IP connection to the remote partner with network address '172.16.2.209' to be established.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v2.7.0 and below	BC9xxx (165) firmware version >= 0xB6	TcModbusTCPBC.Lb6

3.2 FB_ModbusTcpClose

The `FB_ModbusTcpClose` function block closes a connection to a remote partner that was opened with the `FB_ModbusTcpOpen` [► 9] function block. The number of IP connections that may be open is restricted to four, in order to maintain the use of resources at a minimum. If it is necessary to establish a further connection, a connection that is not currently in use must first be closed. A connection is automatically closed after about 10 seconds if it is not used during that period. A confirmation from the remote partner is not expected when a connection is closed. This means that under certain circumstances (e.g., if network communication has been interrupted) it can happen that a connection still exists. The remote partner should then close any such open connections. A Bus Controller operating as a remote partner closes such open connections automatically.

VAR_INPUT

```

VAR_INPUT
  bStart      : BOOL;
  sRemoteIPAddr: STRING(15);
  iMyPortNo  : WORD;
END_VAR

```

bStart: the function block is activated by a rising edge at this input.

sRemoteIPAddr: this variable is a string containing the network address (IP address) of the remote partner whose connection is to be closed.

iMyPortNo: the port number associated with the connection is passed here as a parameter. The port number is generated afresh (dynamically) each time the connection is established with the [FB_ModbusTcpOpen \[► 9\]](#) function block.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorId   : WORD;
END_VAR
```

bBusy: if the function block is activated, this output is set. It remains set until a feedback is received.

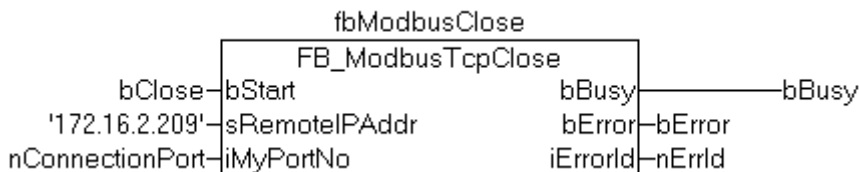
bError: if an error should occur during the execution of the command, then this output is set, after the bBusy output has been reset.

iErrorId: supplies the command-specific error code (table) or the device-specific error code [► 14] if the bError output is set.

Error Codes	Description
0x8400	Connection is already closed.
0x8401	Connection close timeout.
0x8402	Connection close resource error.

Sample of calling the function block in FBD:

```
PROGRAM MAIN
VAR
  fbModbusClose : FB_ModbusTcpClose;
  bOpen         : BOOL;
  bBusy         : BOOL;
  bError        : BOOL;
  nErrId        : WORD;
  nConnectionPort: WORD;
END_VAR
```

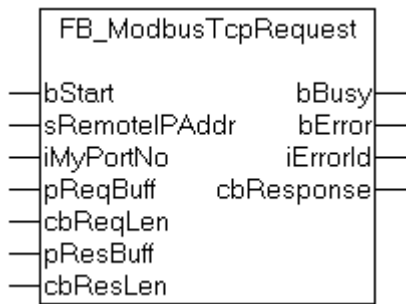


In this sample, a rising edge at the *bClose* input causes a TCP/IP connection to the remote partner with network address '172.16.2.209' to be closed.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v2.7.0 and below	BC9xxx (165) firmware version >= 0xB6	TcModbusTCPBC.Lb6

3.3 FB_ModbusTcpRequest



The `FB_ModbusTcpRequest` function block can send a MODBUS request to a remote partner. The connection required for this must first have been opened with the `FB_ModbusTcpOpen` [► 9] function block. The dynamic port number returned as the connection is established is passed as an input parameter to the `iMyPortNo` variable. Connections that are not used are automatically closed after about 10 seconds, or may be closed by the remote partner. If there is no connection, error code 0x8305 is returned, for example, to the `iErrorId` output. The `FB_ModbusTcpRequest` function block transmits the entire MODBUS frame, beginning with the UNIT ID. The TCP section is inserted by the BC9xxx Bus Controller at the start.

VAR_INPUT

```
VAR_INPUT
  bStart      : BOOL;
  sRemoteIPAddr: STRING(15);
  iMyPortNo   : WORD;
  pReqBuff    : DWORD;
  cbReqLen    : WORD;
  pResBuff    : DWORD;
  cbResLen    : WORD;
END_VAR
```

bStart: the function block is activated by a rising edge at this input.

sRemoteIPAddr: this variable is a string containing the network address (IP address) of the remote partner.

iMyPortNo: the port number associated with the connection is passed here as a parameter. The port number is generated afresh (dynamically) each time the connection is established with the `FB_ModbusTcpOpen` [► 9] function block.

pReqBuff : contains the address of the buffer with the MODBUS request data. It is the programmer who is responsible for dimensioning the buffer appropriately so that `cbReqLen` bytes can be removed from it. The buffer can be a single variable, an array or a structure, whose address can be found with the ADR operator.

cbReqLen : byte length of the MODBUS request data in the `pReqBuff`-buffer.

pResBuff : contains the address of the buffer with the MODBUS response data. The programmer is himself responsible for dimensioning the buffer to a size that can accept `cbResLen` bytes. The buffer can be a single variable, an array or a structure, whose address can be found with the ADR operator.

cbResLen : byte length of the MODBUS response buffer. It is the programmer who is responsible for dimensioning the buffer to such a size that all the response data can be accepted.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorId   : WORD;
  cbResponse : WORD;
END_VAR
```

bBusy: if the function block is activated, this output is set. It remains set until a feedback is received.

bError: if an error should occur during the execution of the command, then this output is set, after the `bBusy` output has been reset.

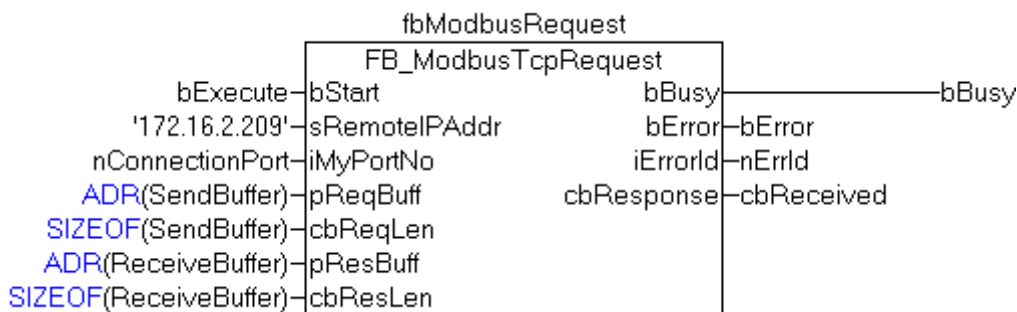
iErrorId: supplies the command-specific error code (table) or the device-specific error code [► 14] if the bError output is set.

cbResponse : the byte length of the response data actually returned in the response buffer.

Error Codes	Description
0x8300	Tx error. Timeout error.
0x8301	Tx error. Remote shutdown.
0x8302	Tx error. Internal error.
0x8303	Tx error. Resource error.
0x8305	Tx error. No remote connection.
0x8100	Rx error. Wrong transaction ID.
0x8101	Rx error. Wrong protocol ID.
0x8102	Rx error. Wrong length.
0x8103	Rx error. Exceeding length.
0x8104	Rx error. Wrong unit ID.

Sample of calling the function block in FBD:

```
PROGRAM MAIN
VAR
  fbModbusRequest : FB_ModbusTcpRequest;
  bExecute        : BOOL;
  nConnectionPort : nConnectionPort;
  SendBuffer      : ARRAY[0..5] OF BYTE := 1, 6, 64, 64, 0, 0;
  ReceiveBuffer   : ARRAY[0..255] OF BYTE;
  bBusy           : BOOL;
  bError          : BOOL;
  nErrId          : WORD;
  cbReceived      : WORD;
END_VAR
```



In this sample, a rising edge at the *bExecute* input causes a request to be sent to the remote partner with network address '172.16.2.209'.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v2.7.0 and below	BC9xxx (165) firmware version >= 0xB6	TcModbusTCPBC.Lb6

4 Appendix

4.1 BC9xxx return codes

Value (hex)	Description
0x8800	No valid network configuration.
0x8E00	Connection closed by target.
0x8F00	Resource error.

Value (hex)	Description Receive Error
0x8100	Receive wrong transaction ID
0x8101	Receive wrong protocol ID
0x8102	Receive wrong length
0x8103	Receive exceeding length
0x8104	Receive wrong unit ID

Value (hex)	Description Transmit Error
0x8300	Transmit timeout
0x8301	Transmit Shutdown
0x8302	Transmit internal error
0x8303	Transmit resource error
0x8305	Transmit remote not connected

Value (hex)	Description Close Error
0x8400	Receive wrong transaction ID
0x8401	Receive wrong protocol ID
0x8402	Receive wrong length

Value (hex)	Description Open Error
0x8500	Open resource error
0x8501	Open resource error_1
0x8502	Open connection error
0x8503	Open connection already open

More Information:
www.beckhoff.com/tx1200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

