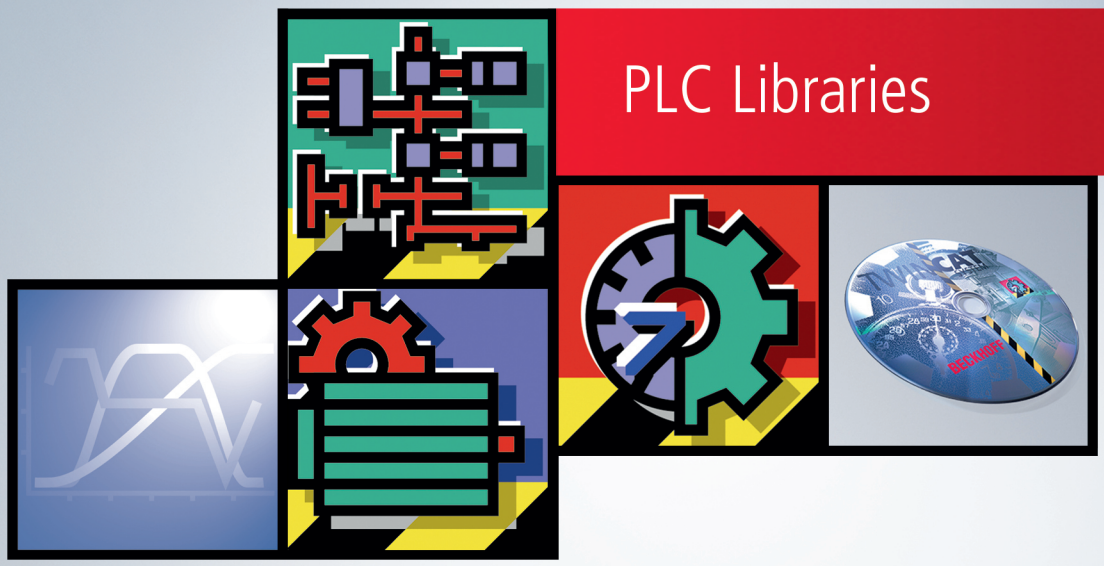
BECKHOFF New Automation Technology

Manual | EN

TX1200

TwinCAT 2 | PLC Library: COMlibRK512BC



2022-05-11 | Version: 1.0

Table of contents

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

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 TwinCAT PLC library COMlib supplies function blocks and data structures for serial data communication. COMlib supports the Beckhoff KL6xxx serial bus terminals.

The COMlib3964RBC communication library extends COMlib to implement an error-protected protocol for the transmission of any type of data. A checksum in combination with repetition of data telegrams in the event of an error are used for error protection.

The RK512 protocol is used to transmit data blocks such as are used in Siemens controllers. The COMlibRK512BC communication library uses the 3964R protocol for this purpose, and handles the telegram traffic that it requires. The user only has to parameterize the RK512 function block.

The "Serial Communication RK512" communication package contains all the necessary libraries. It supports the transmission of data blocks with a length of up to 128 data words.

The documentation for the PLC-Library Serial Communication COMlibBC provides a basis for understanding this communication process.

3 Installation

Installation is performed by copying the following library files into the TwinCAT directory TwinCAT\PLC\LIB.

- COMlibBC5B.LB6
- COMlib3964RBC.LB6
- COMlibRK512BC.LB6

The associated test program should be copied to any project directory of your choice, e.g. to TwinCAT\PLC.

- https://infosys.beckhoff.com/content/1033/tcplclibrk512bc/Resources/zip/12263897099.zip

4 Functional description

4.1 Data Block Configuration

The RK512 function block has two operating modes [▶ 11]. In passive mode it receives data and request telegrams from a communication partner who is addressing data blocks by means of its data block number. Numbered data blocks are not known to IEC-1131, and therefore initially also not to TwinCAT. In TwinCAT, data blocks are variables of various types such as arrays or data structures (STRUCT).

In order to define a numbered data block, the method DBconfig belonging to the RK512 function block as called with all the necessary parameters during the initialization phase. The method is called once for each data block that is to be addressed by communication partners. This configuration is not needed for the active operating mode (send and fetch).

In the active operating mode variable contents can be sent to or fetched from communication partners, independently of this data block configuration. In active mode the data block number refers to the partner device, and only needs to be known there.

Parameters

DbAdr

DbAdr is the memory address of a PLC variable that is to be defined as a data block. The address is determined using the ADR function.

e.g.: DbAdr := ADR(PLCvar);

The PLC variable can be of any type. An ARRAY OF WORD, for example, or a STRUCT data structure would be suitable.

DbSize

DbSize is the size of the PLC variable at *DbAdr* in bytes, determined by the SIZEOF function.

e.g.: DbSize := SIZEOF(PLCvar);

RemoteDbNr

RemoteDbNr contains the number of the data block at the communication partner.

TxBuffer

The send data buffer *TxBuffer* is of type COMbuffer. This parameter is not needed by the configuration, but must however be passed as an IN_OUT parameter.

RxBuffer

The receive data buffer *RxBuffer* is of type COMbuffer. This parameter is not needed by the configuration, but must however be passed as an IN_OUT parameter.

Sample

```
VAR
    (* declare some DB
    (* the type of data doen't matter but the
    (* size shouldn't be larger than 128 bytes
    (**)
    DB1 : ARRAY[1..64] OF WORD; (* exemplary type of db *)
    DB5 : ARRAY[1..64] OF WORD; (* exemplary type of db *)
    DB10 : ARRAY[1..64] OF WORD; (* exemplary type of db *)
```

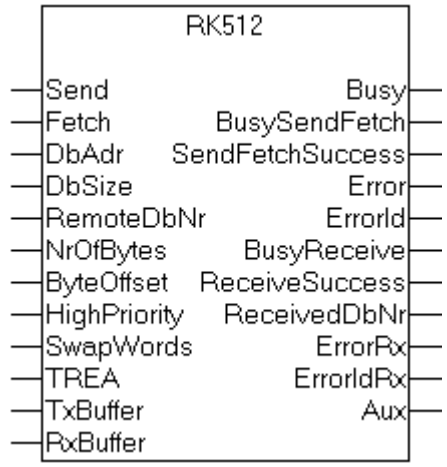
```
    (* input and ouput data for the RK512 function block *)
    RK512com : RK512;

    initialized : BOOL;
END_VAR


IF NOT initialized THEN
    RK512com.DBconfig( RemoteDbNr:=5, DbAdr:=ADR(DB5), DbSize:=SIZEOF(DB5), TxBuffer:=TxBuffer, RxBu
ffer:=RxBuffer );
    RK512com.DBconfig( RemoteDbNr:=10, DbAdr:=ADR(DB10), DbSize:=SIZEOF(DB10), TxBuffer:=TxBuffer, R
xBuffer:=RxBuffer );
    initialized := TRUE;
END_IF
```

The initialization in this sample has the effect that the communication partner may read or write data blocks 5 and 10. Any attempt to access another data block is considered as an error and is refused.

4.2 RK512




```
VAR_INPUT
    Send : BOOL;
    Fetch : BOOL;
    DbAdr : POINTER TO BYTE;
    DbSize : UINT;
    RemoteDbNr :BYTE;
    NrOfBytes : UINT;
    ByteOffset : UINT;
    HighPriority : BOOL;
    SwapWords : BOOL;
    TREA : TIME;
END_VAR
VAR_OUTPUT
    Busy : BOOL;
    BusySendFetch : BOOL;
    SendFetchSuccess : BOOL;
    Error : BOOL;
    ErrorId : INT;
    BusyReceive : BOOL;
    ReceiveSuccess : BOOL;
    ReceivedDbNr : BYTE;
    ErrorRx : BOOL;
    ErrorIdRx : INT;
    Aux : RK512_Auxiliary_t;
END_VAR
VAR_IN_OUT
    TxBuffer : ComBuffer;
    RxBuffer : ComBuffer;
END_VAR
```

Operation modes

The RK512 function block distinguishes two quite different operating modes, passive and active operation.

In the passive mode the function block waits for and answers data telegram from the communication partner. The communication partner can send the data blocks to the local controller, or can fetch data from it. In order for the RK512 function block to be able to accept data blocks or to return data that has been requested, it must previously be configured for passive operation by DBconfig [▶ 10].

In active mode, the RK512 function block either sends data blocks to the communication partner (SEND) or fetches them from it (FETCH).

As long as neither of the function block's *Send* or *Fetch* inputs switch it into active operation it waits for telegrams from communication partners, changing when appropriate into passive mode. The two operating modes can be mixed. In that case the RK512 function block attempts to synchronize the telegram traffic. Since mixed mode operation can introduce delays into the flow of data, it should be avoided if possible.

One instance of the RK512 function block serves just one serial interface. It is not possible for multiple instances to handle data traffic over the same serial interface at the same time.

Input and Output Parameters

Send

With a positive edge at the *Send* input, data from the variable at the *DbAdr* input is sent to the communication partner where it is saved in the data block whose number is given at *RemoteDbNr*. Send and fetch cannot be executed at the same time.

Fetch

With a positive edge at the *Fetch* input, data from the data block *RemoteDbNr* is requested from the communication partner and stored in the variable at input *DbAdr*. Send and fetch cannot be executed at the same time.

DbAdr

DbAdr is the memory address of a PLC variable determined by the ADR function.

e.g.: DbAdr := ADR(PLCvar);

The PLC variable can be of any type. An ARRAY OF WORD, for example, or a STRUCT data structure would be suitable.

Data from this variable is transmitted to the communication partner with a send command, and is fetched and stored in this variable with a fetch command.

DbSize

DbSize is the size of the PLC variable at *DbAdr* in bytes, determined by the SIZEOF function.

e.g.: DbSize := SIZEOF(PLCvar);

RemoteDbNr

RemoteDbNr contains the number of the data block at the communication partner to which the data is to be sent or from which it is to be fetched.

NrOfBytes

NrOfBytes contains the number of data bytes to be transferred. The number can be less than or equal to the variable size *DbSize*.

ByteOffset

The *ByteOffset* indicated the data byte within the communication partner's data block from where the data is to be fetched or where it is to be stored.

HighPriority

The priority is relevant to the 3964R protocol. The two communication partners should have different priorities. If it happens that the two partners transmit at the same time, there is a collision. This collision is resolved by the partner with the lower priority switching to receive mode, allowing the higher priority partner to transmit again.

SwapWords

Data blocks are usually organized by words. Depending on how the data words are stored in the memory, it may be necessary to swap data bytes within the transmitted words. If *SwapWords* is TRUE, the data bytes of each word in all the data telegrams are swapped.

TREA

Every telegram from a communication partner is acknowledged by an answer telegram. The time required for this acknowledgement is monitored with respect to the *TREA* timeout period. *TREA* might be about 15 seconds.

Busy

Busy becomes TRUE as soon as the function block switches into either active or passive mode, i.e. as soon as it leaves the idle state. It cannot accept a new command as long as *Busy* remains TRUE.

BusySendFetch

BusySendFetch becomes TRUE as soon as a positive edge at the *Send* or *Fetch* input switches the function block into active mode. Once *BusySendFetch* has become FALSE the transmission has been completed, and either the *SendFetchSuccess* or the *Error* output is set.

SendFetchSuccess

SendFetchSuccess indicates that a data transmission triggered by Send or Fetch has been successfully completed.

Error

Error becomes TRUE if an error occurs during a data transmission triggered by *Send* or *Fetch*.

ErrorId

ErrorId provides an error number [▶ 14] if an error occurs.

BusyReceive

BusyReceive indicates that the RK512 function block is in receive mode, i.e. in the passive operation mode. The function block changes automatically into passive mode from the idle state (*Busy* is FALSE) as soon as a telegram is received from the communication partner. With falling edge at *BusyReceive* either *ReceiveSuccess* or *ErrorRx* is set.

ReceiveSuccess

ReceiveSuccess is set to TRUE after a data block has been successfully received from the communication partner. This signal is only relevant to passive operation. In other words, *ReceiveSuccess* is not set to TRUE if a data block has been actively fetched from the communication partner with *Fetch*.

ReceivedDbNr

As soon as ReceiveSuccess becomes TRUE, *ReceivedDbNr* indicates the number of the data block that has been received.

ErrorRx

The *ErrorRx* signal indicates that an error has occurred during data reception in passive mode

ErrorIdRx

If an error occurs in passive mode, ErrorIdRx indicates the error number [▶ 14].

Aux

Aux is a data structure containing additional error messages for diagnostic purposes.

TxBuffer

The send data buffer *TxBuffer* is of type COMbuffer. This buffer is used by the RK512 function block, and is not changed by the user.

RxBuffer

The receive data buffer *RxBuffer* is of type COMbuffer. This buffer is used by the RK512 function block, and is not changed by the user.

4.3 Error Codes

No.	No. (hex)	Error constant	Description
4	16#04	RK512ERR_P3964ERR_ZVZ	The character delay time was exceeded. ZVZ = 220 ms. The character delay time is the maximum time interval between the transmission of two succeeding characters.
5	16#05	RK512ERR_P3964ERR_QVZ	The acknowledgement delay time was exceeded. The other end sent an acknowledgement character (DLE = 10H) twice as the telegram was being handled. QVZ = 2 s. The first acknowledgement was expected at the start of the transmission. The output ErrorState when an error initially occurs is "Wait_DLE_TXstart". The cause might lie with a faulty physical connection or an incorrect interface parameter. The second acknowledgement is expected after the user data has been transmitted. If an error occurs, the ErrorState is "Wait_DLE_TXend". In this case the cause might, for instance, be data loss, data corruption or data bytes of a data word that have become swapped (see the SwapWords input).
6	16#06	RK512ERR_P3964ERR_WVZ	The repeat delay time was exceeded. WVZ = 4 s. A telegram is repeated if an error occurs. If the telegram is not repeated by the other device, the 3964R function block reports this error.
7	16#07	RK512ERR_P3964ERR_WRONGBCC	Checksum error during data reception. Each telegram is provided with a checksum. When the data is received, the calculated checksum is compared with the checksum that has been received.
9	16#09	RK512ERR_P3964ERR_COMERRTX	Interface error when transmitting
10	16#0A	RK512ERR_P3964ERR_COMERRRX	Interface error when receiving

No.	No. (hex)	Error constant	Description
11	16#0B	RK512ERR_P3964ERR_NOTXDATA	Parameterization error. The number of bytes that are to be sent, TxCount, is zero.
120	16#78	RK512ERR_P3964ERR_NAK	A telegram was refused by the other device with a negative acknowledgement (NAK). This error can occur in the following transmission states (ErrorState), amongst others: Wait_DLE_TXstart: the local PLC begins a telegram with a start character. The other device refuses the telegram with NAK. Wait_DLE_TXend: the other device refuses the telegram with NAK after the user data has been transmitted. In this case the cause can, for example, be a checksum error at the other device.
12	16#0C	RK512ERR_INVALIDDATATYPE	Invalid data type A telegram has been received whose data type identifier is not supported. Only data blocks with the identifier 'D' can be handled.
16	16#10	RK512ERR_ERRORPROTOCOLHEADER	Faulty protocol header The header of a data telegram is not in accordance with the RK512 specification
20	16#14	RK512ERR_DBNOTAVAILIBLE	Data block not available. A data block that is not available has been transmitted or requested. Data blocks that are accessed by the partner device must first be registered for this purpose with the DBconfig method.
22	16#16	RK512ERR_INVALIDCOMMAND	Invalid command A telegram with an invalid command identifier (SEND / FETCH) has been received.
52	16#34	RK512ERR_INVALIDSIZE	Invalid length quoted Either the DbSize or the NrOfBytes parameter is invalid, or a data telegram that is too long has been received.
53	16#35	RK512ERR_INVALIDDBADR	Invalid data block address The data block address DbAdr is invalid.
54	16#36	RK512ERR_SYNCERROR	Synchronization error A synchronization error can occur if both communication partners start to transmit at the same time.
257	16#101	RK512ERR_TIMEOUT	Timeout on the RK512 telegram level An expected reaction telegram has not been received within the waiting time TREA.
258	16#102	RK512ERR_ERRORREMOTEREATEL	Error in the reaction telegram An error was reported in the communication partner's reaction telegram. The error number can be read from the additional error information in AUX.
259	16#103	RK512ERR_INVALIDSIZEREMOTEREATEL	The communication partner's reaction telegram has an incorrect length
260	16#104	RK512ERR_TIMEOUTREAPEATSENDFETCH	Timeout when transmitting In spite of a number of repetitions, a send or fetch telegram could not be transmitted.
261	16#105	RK512ERR_3964ERROR	Error on the 3964R protocol level Predictable errors on this level are not reported as a general RK512ERR_3964ERROR error, but with a detailed error number.

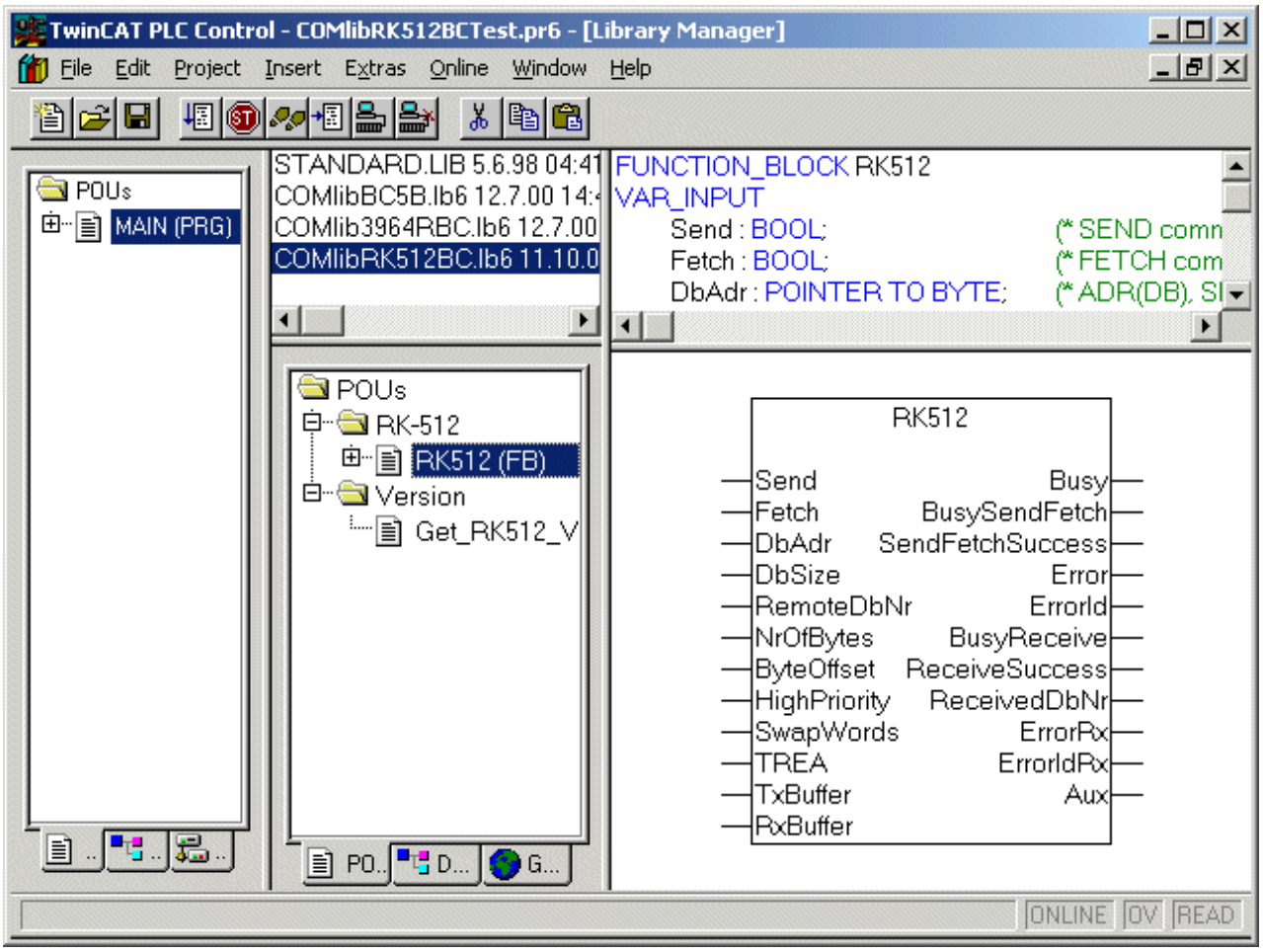
No.	No. (hex)	Error constant	Description
262	16#106	RK512ERR_3964NOTBUSYNOTREADY	The transmission of a telegram has been finished without having been successfully completed.

5 Linking into a PLC Program

5.1 Link Libraries

Create a new PLC project with TwinCAT PLC Control in order to perform the library linking.

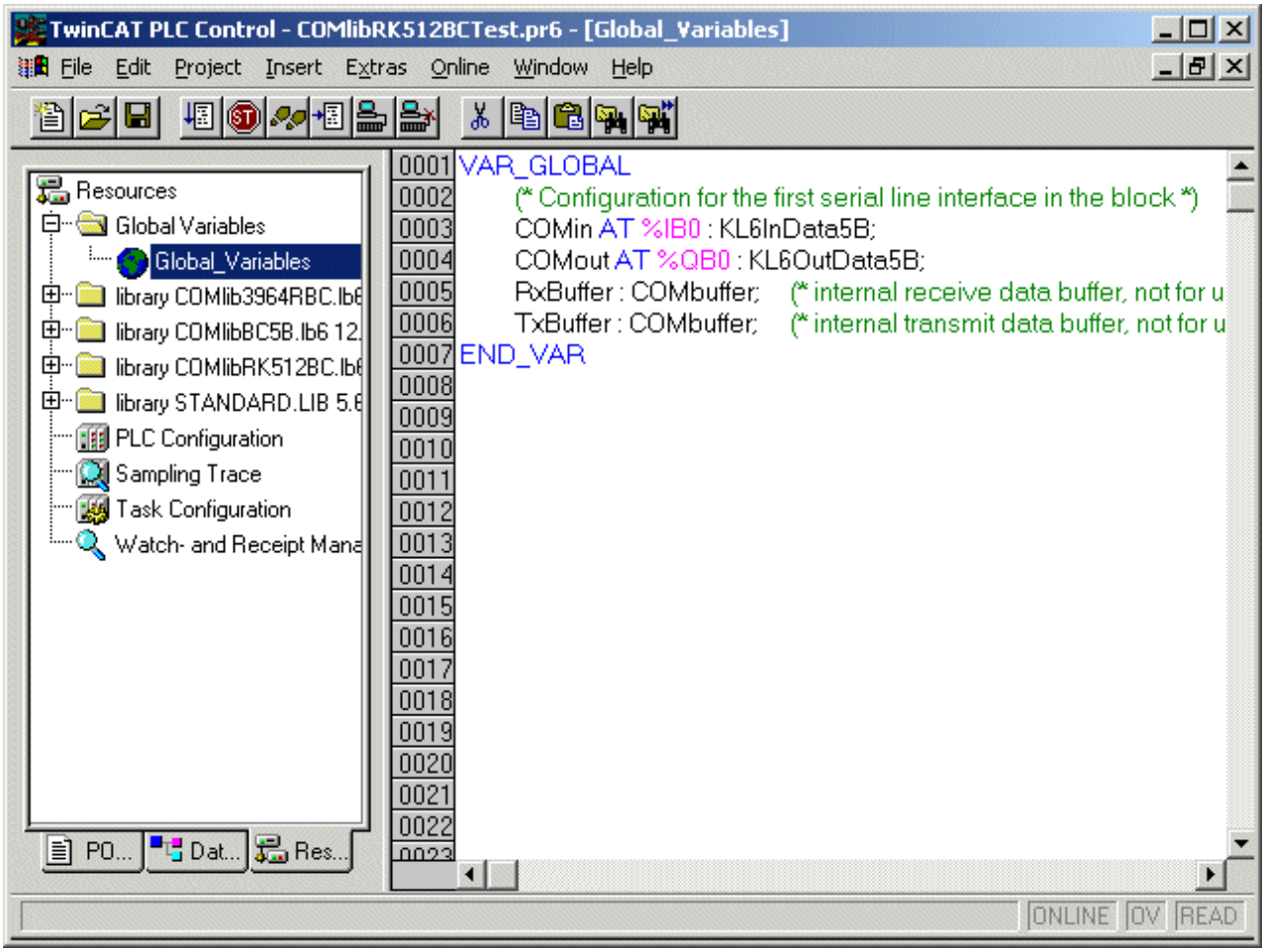
Go to Library Management and add the libraries ComLibBC5B.LIB, ComLib3964RBC.LIB and ComLibRK512BC.LIB.



5.2 Global Variable

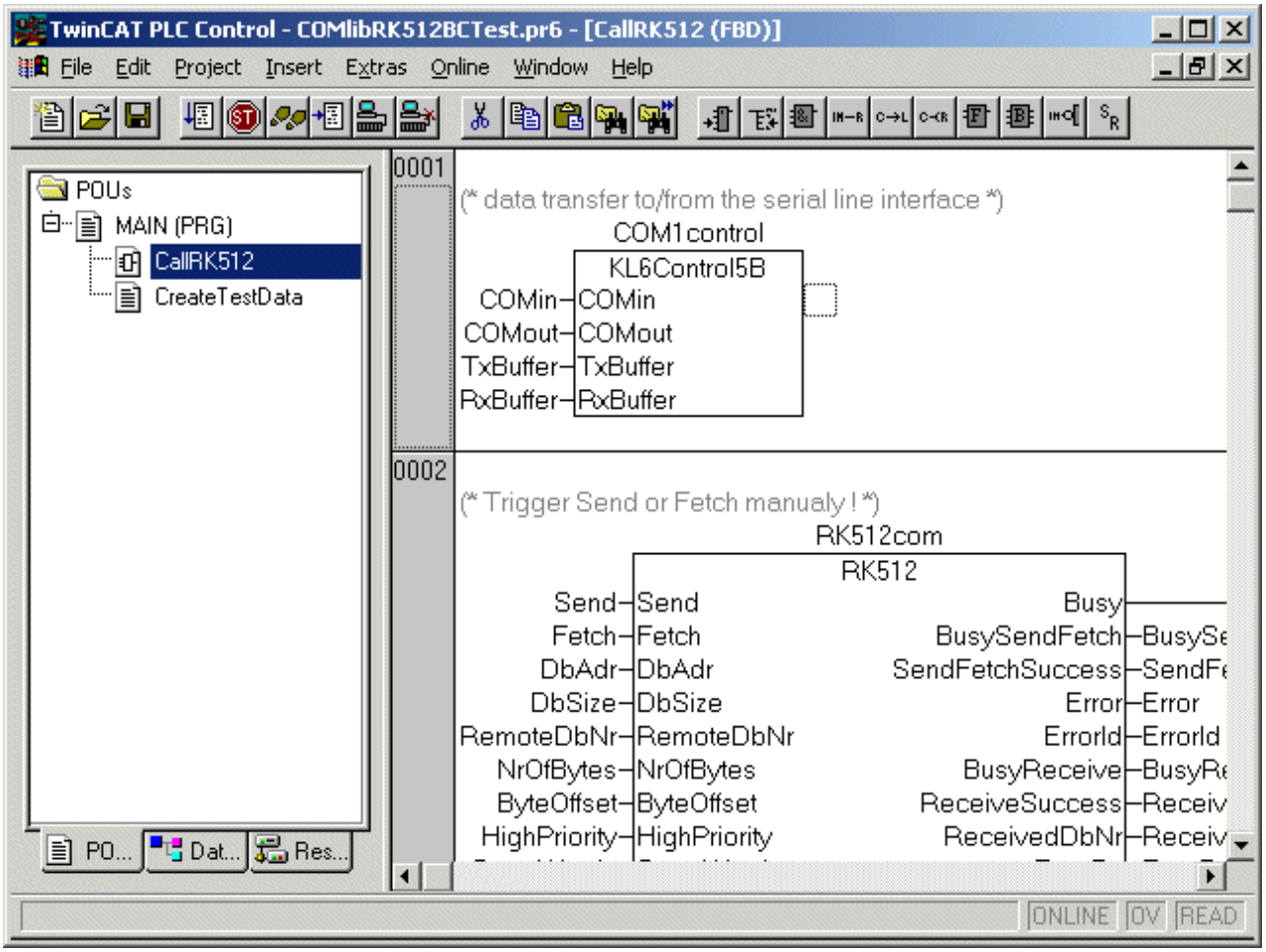
Four global data structures are needed to access a serial interface. Two provide the connection to the hardware in the send and receive directions. Two data buffers are also necessary for intermediate storage.

The data structures of type PcComInData or PcComOutData are linked with the hardware in the TwinCAT System Manager. Please read the corresponding chapter in the COMlibBC documentation in this regard.



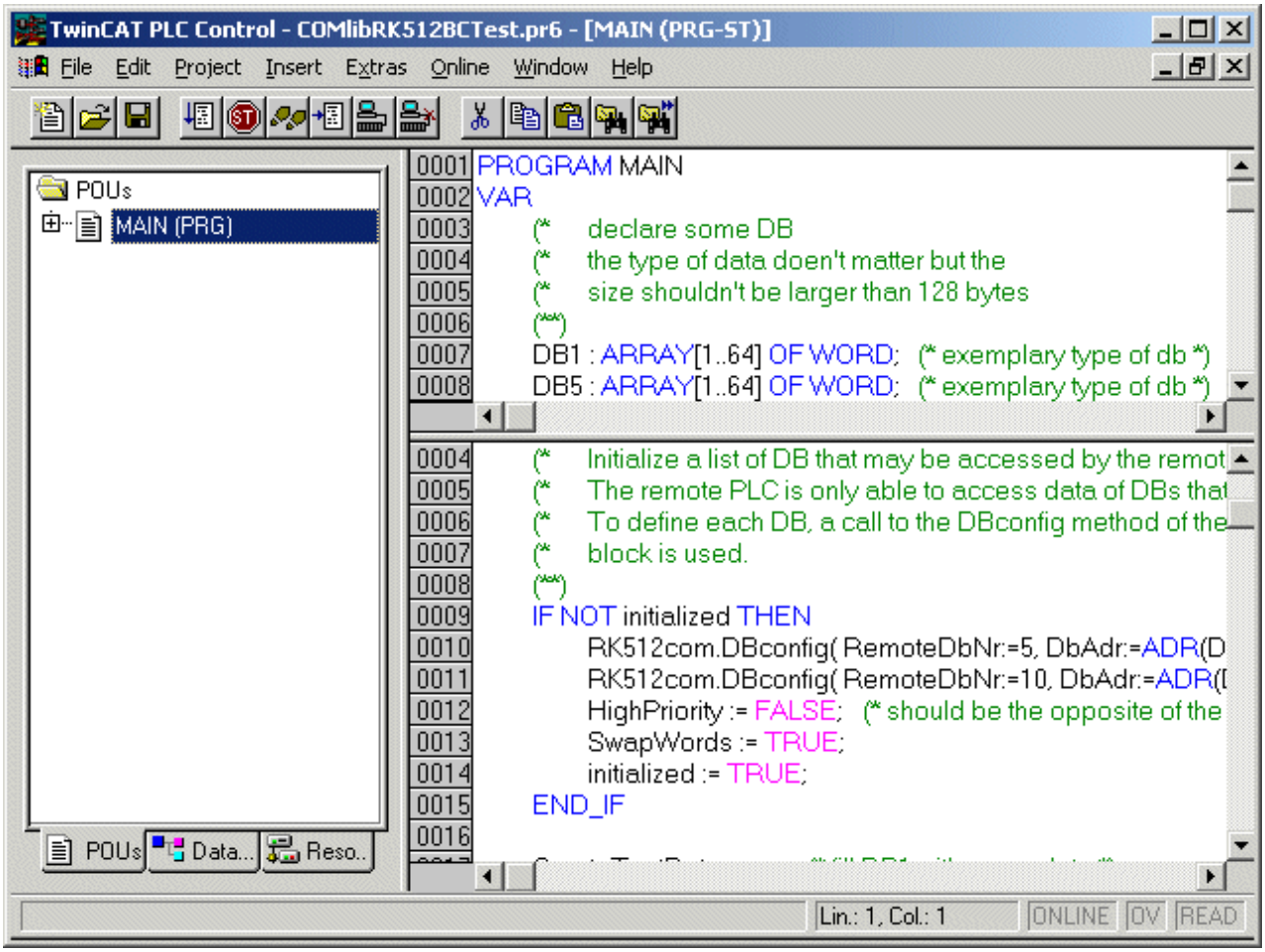
5.3 Background Communication

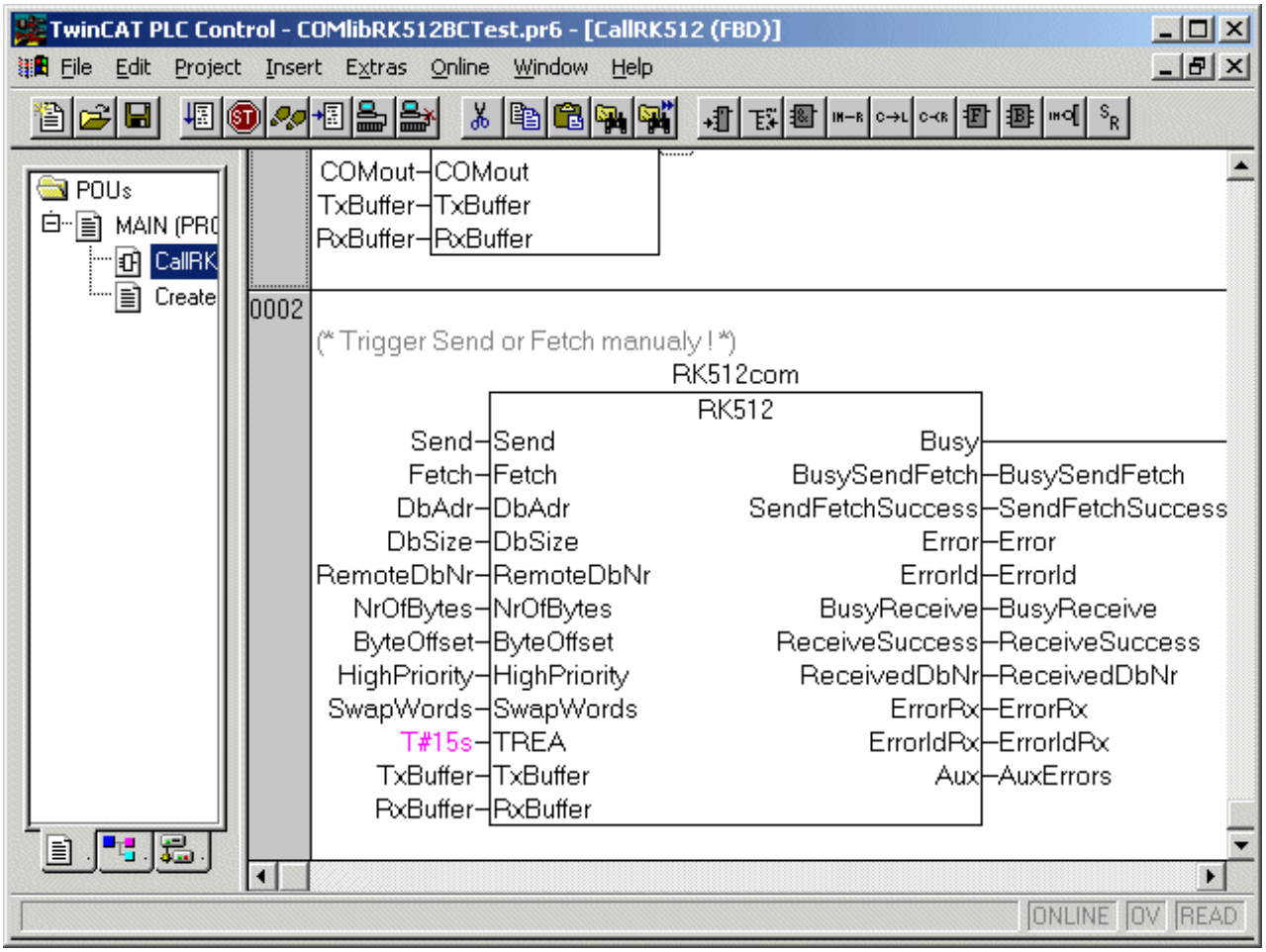
Communication between the serial hardware and the data buffer, which is of type ComBuffer, is handled by a separate function block. The general principles of this subject are described in the documentation for the COMlibBC communication library.



5.4 RK512 Protocol

The sample program defines three WORD arrays, in order to use them as data blocks. During initialization the data blocks 5 and 10 are registered for access by the communication partners by calling the DBconfig method. The function block RK512 behaves passively, and is ready to accept and to answer data telegrams from the partner device. The test program allows active sending or fetching of data to be initiated by writing to the *Send* or *Fetch* variables.


TwinCAT PLC Control - COMlibRK512BCTest.pr6 - [MAIN (PRG-ST)]
File Edit Project Insert Extras Online Window Help
POUs
MAIN (PRG)
POUs Data... Reso..
0001 PROGRAM MAIN
0002 VAR
0003 (* declare some DB
0004 (* the type of data doen't matter but the
0005 (* size shouldn't be larger than 128 bytes
0006 (**)
0007 DB1 : ARRAY[1..64] OF WORD; (* exemplary type of db *)
0008 DB5 : ARRAY[1..64] OF WORD; (* exemplary type of db *)
0004 (* Initialize a list of DB that may be accessed by the remot
0005 (* The remote PLC is only able to access data of DBs that
0006 (* To define each DB, a call to the DBconfig method of the
0007 (* block is used.
0008 (**)
0009 IF NOT initialized THEN
0010 RK512com.DBconfig(RemoteDbNr:=5, DbAdr:=ADR(D
0011 RK512com.DBconfig(RemoteDbNr:=10, DbAdr:=ADR(
0012 HighPriority := FALSE; (* should be the opposite of the
0013 SwapWords := TRUE;
0014 initialized := TRUE;
0015 END_IF
0016
Lin.: 1, Col.: 1
ONLINE
OV
READ


TwinCAT PLC Control - COMlibRK512BCTest.pr6 - [CallRK512 (FBD)]
File Edit Project Insert Extras Online Window Help
POUs
MAIN (PRC
CallRK
Create
COMout COMout
TxBuffer TxBuffer
RxBuffer RxBuffer
0002
(* Trigger Send or Fetch manualy ! *)
RK512com
RK512
Send Send
Fetch Fetch
DbAdr DbAdr
DbSize DbSize
RemoteDbNr RemoteDbNr
NrOfBytes NrOfBytes
ByteOffset ByteOffset
HighPriority HighPriority
SwapWords SwapWords
T#15s TREA
TxBuffer TxBuffer
RxBuffer RxBuffer
Busy
BusySendFetch BusySendFetch
SendFetchSuccess SendFetchSuccess
Error Error
ErrorId ErrorId
BusyReceive BusyReceive
ReceiveSuccess ReceiveSuccess
ReceivedDbNr ReceivedDbNr
ErrorRx ErrorRx
ErrorIdRx ErrorIdRx
Aux AuxErrors
ONLINE OV READ

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