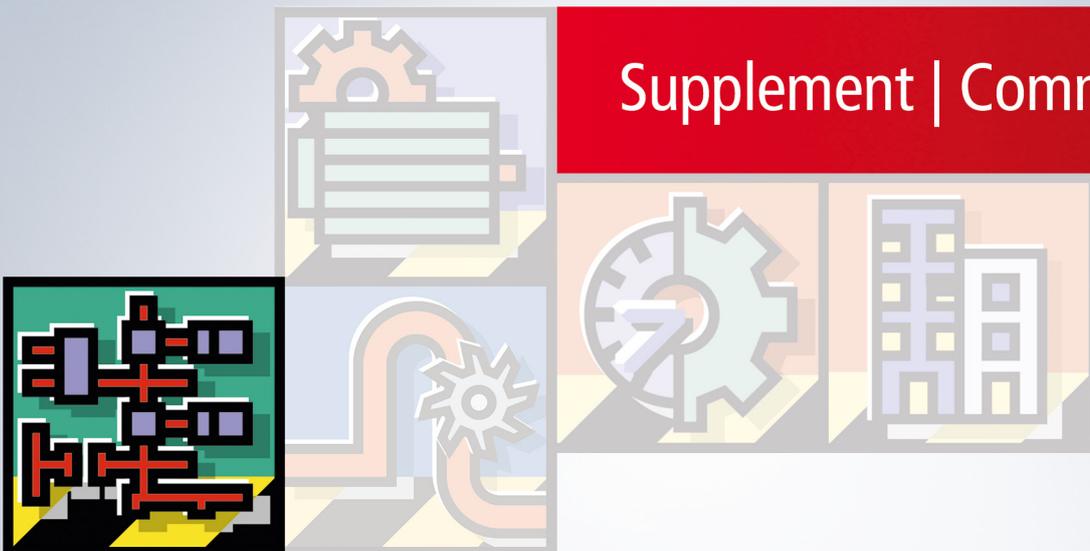


Handbuch | DE

TS6341

TwinCAT 2 PLC Serial Communication 3964R/RK512

Supplement | Communication



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht	8
3	Installation	9
4	ComLib3964R.LIB	10
4.1	P3964R	10
4.2	P3694buffer.....	11
4.3	Fehlercodes von P3964R.....	11
5	ComLibRK512.LIB	13
5.1	RK512	13
5.1.1	DBconfig.....	16
5.2	Fehlercodes von RK512.....	17
6	Beispiele	21
6.1	Bilbiotheken einbinden	21
6.2	Globale Variable.....	22
6.3	Taskkonfiguration	23
6.4	Hintergrundkommunikation	24
6.5	3964R-Kommunikation.....	25
6.6	RK512-Kommunikation	26

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zu widerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!

Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

GEFAHR

Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

WARNUNG

Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

VORSICHT

Schädigung von Personen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

Tipp oder Fingerzeig



Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

2 Übersicht

Die TwinCAT SPS Bibliothek COMlib bietet Funktionsbausteine und Datenstrukturen für die serielle Datenkommunikation. Die COMlib unterstützt die seriellen Beckhoff Busklemmen KL6xxx und die Standard PC-Schnittstellen COMx.

Die Bibliothek COMlib wird in dem Dokument „COMlib SPS-Bibliothek – Serielle Datenkommunikation“ beschrieben. Dieses Dokument ist als Ergänzung dazu zu sehen.

3964R - Kommunikation

Die Kommunikationsbibliothek COMlib3964R realisiert aufbauend auf der COMlib ein fehlersicherndes Protokoll zur Übertragung beliebiger Daten. Zur Fehlersicherung dienen eine Prüfsumme und Wiederholungen der Datentelegramme im Fehlerfall.

Zusätzlich zu den Grundfunktionen Senden und Empfangen transparenter Daten unterstützt die COMlib3964R das Protokoll 3964R. Die Bibliotheken COMlib und COMlib3964R ersetzen damit die bisherige Bibliothek P3964lib. Beide Bibliotheken sind Bestandteil des 3964R-Paketes.

RK512 – Kommunikation

Um Datenbausteine, wie sie z. B. in Siemens-Steuerungen verwendet werden, zu übertragen, wird das Protokoll RK512 benutzt. Die Kommunikationsbibliothek COMlibRK512 nutzt dazu das 3964R-Protokoll und wickelt den notwendigen Telegrammverkehr ab. Der Anwender parametriert dazu nur den Funktionsbaustein RK512.

Das Kommunikationspaket "Serial Communication RK512" beinhaltet alle notwendigen Bibliotheken. Es unterstützt die Übertragung von Datenbausteinen bis zu einer Länge von 128 Datenwörtern.

3 Installation

Installation von COMlib3964R.LIB

Zur Verwendung des 3964R-Protokolls werden folgende Bibliotheken benötigt. Diese Bibliotheken müssen in das TwinCAT Verzeichnis TwinCAT\PLC\LIB kopiert werden.

- COMlib.LIB
- ChrAsc.LIB
- ChrAsc.OBJ
- COMlib3964R.LIB

Das beiliegende Testprogramm (COMlib3964Test.pro, COMlib3964Test.wsm) kopieren Sie in ein beliebiges Projektverzeichnis, z. B. nach TwinCAT\PLC.

- <https://infosys.beckhoff.com/content/1031/tcplclib3964r/Resources/12566747531.zip>

Installation von COMlibRK512.LIB

Zur Verwendung des RK512-Protokolls werden folgende Bibliotheken benötigt. Diese Bibliotheken müssen in das TwinCAT Verzeichnis TwinCAT\PLC\LIB kopiert werden.

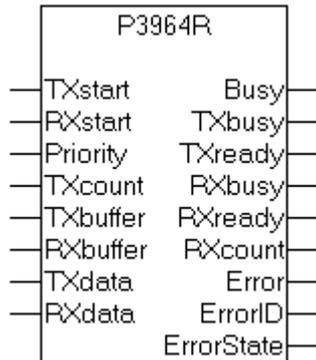
- COMlib.LIB
- ChrAsc.LIB
- ChrAsc.OBJ
- COMlib3964R.LIB
- COMlibRK512.LIB

Das beiliegende Testprogramm (COMlibRK512Test.pro, COMlibRK512Test.wsm) kopieren Sie in ein beliebiges Projektverzeichnis, z. B. nach TwinCAT\PLC.

- <https://infosys.beckhoff.com/content/1031/tcplclib3964r/Resources/12566750219.zip>

4 ComLib3964R.LIB

4.1 P3964R



```

VAR_INPUT
  TXstart  : BOOL; (* Startsignal, Flankengetriggert *)
  RXstart  : BOOL; (* Startsignal, Flankengetriggert *)
  Priority  : BOOL; (* 3964-Priorität *)
  TXcount  : INT;  (* Anzahl Datenbytes in TxData *)
END_VAR
VAR_OUTPUT
  Busy      : BOOL; (* Baustein aktiv *)
  TXbusy    : BOOL; (* Sendezustand *)
  TXready   : BOOL; (* Senden beendet *)
  RXbusy    : BOOL; (* Empfangszustand *)
  RXready   : BOOL; (* Empfang beendet *)
  RXcount   : INT;  (* Anzahl Datenbytes in RxData *)
  Error     : BOOL; (* Fehler *)
  ErrorID   : INT;  (* Fehlernummer *)
  ErrorState : STRING(80); (* interner Fehlerzustand *)
END_VAR
VAR_IN_OUT
  TXbuffer : COMbuffer; (* Zwischenpuffer zur Hardware *)
  RXbuffer : COMbuffer; (* Zwischenpuffer zur Hardware *)
  TXdata   : P3964buffer; (* Anwendungsdaten (zu senden) *)
  RXdata   : P3964buffer; (* Anwendungsdaten (empfangen) *)
END_VAR

```

Hardwareanbindung

Der Funktionsbaustein P3964R wickelt das 3964R-Protokoll ab. Über die verwendete Hardwareschnittstelle entscheidet die Hintergrundkommunikation, die durch einen separaten Baustein (PcComControl, KL6Control, KL6Control5B) abgewickelt wird. Die zugehörigen [Datenpuffer](#) [► 11] Txbuffer und Rxbuffer werden an den Baustein P3964R übergeben.

Senden

Sendedaten werden von der SPS in den Sendedatenpuffer Txdata eingetragen. Die eingetragene Anzahl der Datenbytes wird in Txcount übergeben und anschließend Txstart auf TRUE gesetzt. Txstart ist flankengetriggert, das heißt, vor einem neuen Sendeversuch muß der Baustein mit Txstart=FALSE aufgerufen werden. Im Sendezustand ist Txbusy=TRUE. Nachdem die Daten erfolgreich übertragen wurden, wird Txready=TRUE.

Empfangen

Zum Empfang von Daten wird RxStart auf TRUE gesetzt. Nachdem ein vollständiger Datensatz empfangen wurde, geht der Ausgang Rxready auf TRUE und Rxcount Datenbytes liegen im Empfangsdatenpuffer Rxdata bereit. Rxstart ist ebenfalls flankengetriggert. Im Empfangszustand ist Rxbusy=TRUE.

Empfang abbrechen

Der Baustein P3964R kann entweder im Sende- oder im Empfangsmodus betrieben werden. Es ist empfehlenswert, den Baustein in der Zeit, in der nicht gesendet wird auf Empfang zu schalten. Solange der Baustein im Empfangsmodus ist, jedoch noch kein Startzeichen der Gegenstelle empfangen wurde kann durch eine steigende Flanke an Txstart der Empfang unterbrochen und das Senden von Daten eingeleitet werden. Nach der Übertragung geht der Baustein in den Grundzustand (Busy=FALSE) und wird erst durch eine neue Flanke an Rxstart auf Empfang geschaltet.

Priorität

Bei einer 3964R-Datenverbindung darf jede Gegenstelle zu einem beliebigen Zeitpunkt senden. Versuchen beide Seiten gleichzeitig zu senden, kommt es zu einer Kollision. In diesem Fall geht die Seite mit niedriger Priorität in den Empfangszustand und die Seite mit hoher Priorität sendet erneut. Aus diesem Grund kann bei niedriger Priorität der Ausgang RxReady TRUE werden, obwohl Daten gesendet werden sollten. Die Daten in RxData sind dann gegebenenfalls auszuwerten. Die Einstellung der Priorität auf beiden Seiten ist Vereinbarungssache.

Fehler

Im Fehlerfall geht der Baustein in den Grundzustand (Busy=FALSE) und der Ausgang Error wird TRUE. Die ErrorId gibt den Fehlercode [► 11] zurück. ErrorState gibt bei Kenntnis der internen Protokollstruktur Aufschluss über den Zustand, in dem der Fehler aufgetreten ist.

4.2 P3694buffer

Datenpuffer vom Typ P3964buffer werden in Verbindung mit dem Funktionsbaustein P3964R verwendet. Diese Puffer werden vom Anwender beschrieben und gelesen.

```

TYPE P3964buffer
  STRUCT
    D : ARRAY [0..16#0FFF] OF BYTE;
  END_STRUCT
END_TYPE
    
```

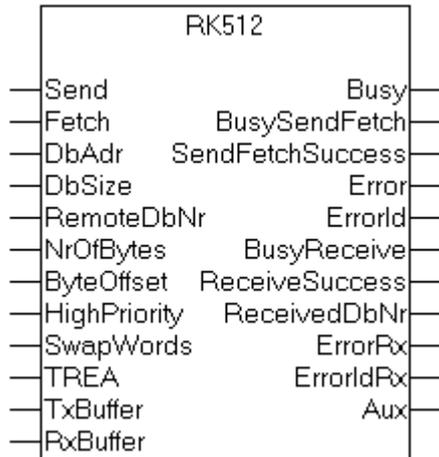
4.3 Fehlercodes von P3964R

Nr.	Fehlerkonstante	Beschreibung
4	P3964ERR_ZVZ	Die Zeichenverzugszeit wurde überschritten. ZVZ = 220 ms. Die Zeichenverzugszeit ist der maximale Zeitabstand zwischen der Übertragung zweier aufeinanderfolgender Zeichen.
5	P3964ERR_QVZ	Die Quittungsverzugszeit wurde überschritten. Die Gegenstelle sendet während der Telegrammabwicklung zweimal ein Quittungszeichen (DLE = 10H). QVZ = 2 s. Die erste Quittung wird am Anfang der Übertragung erwartet. Der Ausgang ErrorState ist im Fehlerfall "Wait_DLE_TXstart". Ursache können z. B. eine fehlerhafte physikalische Verbindung oder eine falsche Schnittstellenparameter sein. Die zweite Quittung wird nach der Übertragung der Nutzdaten erwartet. Im Fehlerfall ist der ErrorState "Wait_DLE_TXend". Ursache können in diesem Fall z. B. Datenverlust, Datenverfälschung oder vertauschte Datenbytes eines Datenwortes (vergleiche Eingang SwapWords) sein.
6	P3964ERR_WVZ	Die Wiederholungsverzugszeit wurde überschritten. WVZ = 4 s. Im Fehlerfall wird ein Telegramm wiederholt. Bleibt im Wiederholungsfall das Telegramm von der Gegenstelle aus, meldet der Funktionsbaustein 3964R diesen Fehler.
7	P3964ERR_WRONGBCC	Prüfsummenfehler beim Datenempfang. Jedes Telegramm wird mit einer Prüfsumme versehen. Beim Datenempfang wird die berechnete Prüfsumme mit der empfangenen Prüfsumme verglichen.

Nr.	Fehlerkonstante	Beschreibung
9	P3964ERR_COMERTX	Schnittstellenfehler beim Senden
10	P3964ERR_COMERRRX	Schnittstellenfehler beim Empfang
11	P3964ERR_NOTXDATA	Parametrierungsfehler. Die Anzahl der zu sendenden Datenbytes <i>TxCount</i> ist Null.
20	P3964ERR_NAK	Ein Telegramm wurde von der Gegenstelle mit einer Negativquittung (NAK) zurückgewiesen. Der Fehler kann u. a. in folgenden Übertragungszuständen (<i>ErrorState</i>) auftreten: <i>Wait_DLE_TXstart</i> : Die lokale SPS leitet ein Telegramm mit einem Startzeichen ein. Die Gegenstelle weist das Telegramm mit NAK ab. <i>Wait_DLE_TXend</i> : Die Gegenstelle weist das Telegramm nach Nutzdatenübertragung mit NAK ab. Ursache kann in diesem Fall z. B. ein Prüfsummenfehler in der Gegenstelle sein.

5 ComLibRK512.LIB

5.1 RK512



```

VAR_INPUT
  Send      : BOOL;          (* SEND command *)
  Fetch     : BOOL;          (* FETCH command *)
  DbAdr     : POINTER TO BYTE; (* ADR(DB), SEND=source, FETCH=target *)
  DbSize    : UINT;          (* SIZEOF(DB), size in byte for Send or Fetch *)
  RemoteDbNr : BYTE;          (* SEND=target, FETCH=source *)
  NrOfBytes : UINT;          (* number of bytes to send or fetch *)
  ByteOffset : UINT;          (* byte offset in remote DB *)
  HighPriority : BOOL;        (* 3964R priority *)
  SwapWords  : BOOL;          (* swaps every two bytes in the data stream before/
after transmission *)
  TREA       : TIME;          (* Timeout for answer *)
END_VAR
VAR_OUTPUT
  Busy      : BOOL;          (* FB is busy *)
  BusySendFetch : BOOL;      (* FB is with SendFetchCommand busy *)
  SendFetchSuccess : BOOL;    (* Send or Fetch successfully finished *)
  Error     : BOOL;          (* RK512 error *)
  ErrorId   : INT;           (* RK512 error id *)
  BusyReceive : BOOL;        (* FB is busy with remote request *)
  ReceiveSuccess : BOOL;      (* DB has been received from the remote PLC *)
  ReceivedDbNr : BYTE;        (* Number of the DB that has been received *)
  ErrorRx    : BOOL;          (* RK512 error. For receive mode only *)
  ErrorIdRx  : INT;           (* RK512 error id. For receive mode only *)
  Aux       : RK512_Auxiliary_t; (* Additional errors for debugging *)
END_VAR
VAR_IN_OUT
  TxBuffer : ComBuffer;
  RxBuffer : ComBuffer;
END_VAR

```

Betriebsarten

Der Funktionsbaustein RK512 unterscheidet grundsätzlich zwei Betriebsarten, den passiven Betrieb und den aktiven Betrieb.

In der passiven Betriebsart wartet der Baustein auf Datentelegramme vom Kommunikationspartner und antwortet darauf. Der Kommunikationspartner kann Datenbausteine in die lokale Steuerung senden oder Daten von ihr holen. Damit der Funktionsbaustein RK512 Datenbausteine entgegennehmen oder die angeforderten Daten zurücksenden kann, muss er vorher für die passive Betriebsart mit [DBconfig](#) [► 16] konfiguriert werden.

In der aktiven Betriebsart sendet der Funktionsbaustein RK512 Datenbausteine an den Kommunikationspartner (SEND) oder holt Datenbausteine von ihm ab (FETCH).

Solange der Funktionsbaustein nicht über seine Eingänge *Send* oder *Fetch* in den aktiven Betrieb wechselt, wartet er auf Telegramme vom Kommunikationspartner und wechselt gegebenenfalls automatisch in den passiven Betrieb. Beide Betriebsarten können gemischt werden. In diesem Fall versucht der Funktionsbaustein RK512 den Telegrammverkehr zu synchronisieren. Da ein Mischbetrieb zu Verzögerungen im Datenverkehr führen kann, sollte er wenn möglich vermieden werden.

Eine Instanz des Funktionsbausteins RK512 bedient genau eine serielle Schnittstelle. Es können nicht mehrere Instanzen zur selben Zeit den Datenverkehr über dieselbe serielle Schnittstelle abwickeln.

Eingangs- und Ausgangsparameter

Send

Mit einer positiven Flanke am Eingang *Send* werden Daten aus der Variablen am Eingang *DbAdr* an den Kommunikationspartner übertragen und dort im Datenbaustein mit der Nummer *RemoteDbNr* abgelegt. *Send* und *Fetch* können nicht gleichzeitig ausgeführt werden.

Fetch

Mit einer positiven Flanke am Eingang *Fetch* werden Daten aus dem Datenbaustein *RemoteDbNr* vom Kommunikationspartner angefordert und in der Variablen am Eingang *DbAdr* gespeichert. *Send* und *Fetch* können nicht gleichzeitig ausgeführt werden.

DbAdr

DbAdr ist die Speicheradresse einer SPS-Variablen, die mit der ADR-Funktion ermittelt wird.

z. B.: `DbAdr := ADR(PLCvar);`

Die SPS-Variable kann beliebigen Typs sein. Beispielsweise ist ein ARRAY OF WORD oder eine Datenstruktur STRUCT geeignet.

Mit einem *Send*-Kommando werden die Daten aus dieser Variablen an den Kommunikationspartner übertragen, mit einem *Fetch*-Kommando werden Daten geholt und in der Variablen gespeichert.

DbSize

DbSize ist die Größe der SPS-Variablen an *DbAdr* in Byte, die mit der SIZEOF-Funktion ermittelt wird.

z. B. `DbSize := SIZEOF(PLCvar);`

RemoteDbNr

RemoteDbNr enthält die Nummer des Datenbausteins beim Kommunikationspartner zu dem Daten gesendet werden bzw. aus dem Daten geholt werden.

NrOfBytes

NrOfBytes enthält die Anzahl der zu übertragenden Datenbytes. Die Anzahl kann kleiner oder gleich der Variablengröße *DbSize* sein.

ByteOffset

Der *ByteOffset* gibt an, ab welchem Datenbyte die Daten im Datenbaustein des Kommunikationspartners gespeichert, bzw. geholt werden.

HighPriority

Die Priorität bezieht sich auf das 3964R-Protokoll. Beide Kommunikationspartner sollten unterschiedliche Priorität haben. In dem Fall, dass beide Partner zur gleichen Zeit senden, kommt es zu einer Kollision. Diese Kollision wird dadurch aufgelöst, dass der Partner mit niedriger Priorität in den Empfangsmodus schaltet und der Partner mit hoher Priorität erneut sendet.

SwapWords

Datenbausteine sind normalerweise Wortweise organisiert. Je nachdem, wie die Datenwörter im Speicher abgelegt werden, ist es notwendig die Datenbytes der übertragenen Wörter zu tauschen. Ist *SwapWords* TRUE, werden in allen Datentelegrammen die Datenbytes eines Wortes getauscht.

TREA

Jedes Telegramm eines Kommunikationspartners wird mit einem Antworttelegramm quittiert. Diese Quittierung wird mit dem Timeout *TREA* zeitlich überwacht. *TREA* liegt z. B. bei 15 Sekunden.

Busy

Busy wird TRUE, sobald der Baustein in den aktiven oder passiven Betrieb wechselt, also seinen Ruhezustand verlässt. Solange *Busy* TRUE ist, kann er keine neues Kommandos annehmen.

BusySendFetch

BusySendFetch wird TRUE, sobald der Baustein durch eine positive Flanke am *Send-* oder *Fetch-*Eingang in den aktiven Betrieb wechselt. Nachdem *BusySendFetch* FALSE wird, ist die Übertragung abgeschlossen und entweder der Ausgang *SendFetchSuccess* oder *Error* ist gesetzt.

SendFetchSuccess

SendFetchSuccess signalisiert, dass eine mit *Send* oder *Fetch* eingeleitete Datenübertragung erfolgreich abgeschlossen wurde.

Error

Error wird TRUE, wenn bei einer durch *Send* oder *Fetch* eingeleiteten Datenübertragung ein Fehler auftritt.

ErrorId

ErrorId zeigt im Fehlerfall eine [Fehlernummer \[► 17\]](#) an.

BusyReceive

BusyReceive zeigt an, dass sich der Funktionsbaustein RK512 im Empfangsmodus, d. h. in der passiven Betriebsart befindet. Der Baustein wechselt automatisch in die passive Betriebsart, sobald im Ruhezustand (*Busy* gleich FALSE) ein Telegramm vom Kommunikationspartner empfangen wird. Mit fallender Flanke an *BusyReceive* wird entweder *ReceiveSuccess* oder *ErrorRx* gesetzt.

ReceiveSuccess

Nachdem ein Datenbaustein vom Kommunikationspartner erfolgreich empfangen wurde wird *ReceiveSuccess* TRUE. Dieses Signal betrifft nur den passiven Betrieb, *ReceiveSuccess* wird also nicht TRUE, wenn mit *Fetch* aktiv ein Datenbaustein vom Kommunikationspartner geholt wird.

ReceivedDbNr

Sobald *ReceiveSuccess* TRUE wird zeigt *ReceivedDbNr* die Nummer des empfangenen Datenbausteins an.

ErrorRx

Das Signal *ErrorRx* zeigt an, dass beim Datenempfang in der passiven Betriebsart ein Fehler aufgetreten ist

ErrorIdRx

ErrorIdRx zeigt im Fehlerfall im der passiven Betriebsart die [Fehlernummer \[► 17\]](#) an.

Aux

Aux ist eine Datenstruktur, die zusätzliche Fehlermeldungen zur Diagnose enthält.

TxBuffer

Der Sendedatenpuffer *TxBuffer* ist vom Typ COMbuffer. Dieser Puffer wird vom Baustein RK512 verwendet und wird vom Benutzer nicht geändert.

RxBuffer

Der Empfangsdatenpuffer *RxBuffer* ist vom Typ COMbuffer. Dieser Puffer wird vom Baustein RK512 verwendet und wird vom Benutzer nicht geändert.

5.1.1 DBconfig

Der Funktionsbaustein RK512 arbeitet in zwei Betriebsarten [► 13]. In der passiven Betriebsart empfängt er Daten und Anforderungstelegramme von einem Kommunikationspartner, der Datenbausteine über ihre Datenbausteinnummer adressiert. Nummerierte Datenbausteine sind in der IEC-1131 und damit auch in TwinCAT zunächst unbekannt. Datenbausteine sind in TwinCAT Variablen verschiedenen Typs, z. B. Arrays oder Datenstrukturen (STRUCT).

Zur Definition eines nummerierten Datenbausteins wird in einer Initialisierungsphase die Methode DBconfig des Funktionsbausteins RK512 mit allen notwendigen Parametern aufgerufen. Der Aufruf der Methode wird einmalig für jeden Datenbaustein durchgeführt, der vom Kommunikationspartner adressiert werden soll. Für die aktive Betriebsart (Send und Fetch) ist diese Konfiguration nicht notwendig.

In der aktiven Betriebsart können unabhängig von dieser Datenbausteinkonfiguration Variableninhalte zum Kommunikationspartner gesendet bzw. von diesem geholt werden. Die Datenbausteinnummer bezieht sich in der aktiven Betriebsart auf die Gegenstelle und muss nur dort bekannt sein.

Parameter

DbAdr

DbAdr ist die Speicheradresse einer SPS-Variablen, die als Datenbaustein definiert werden soll. Die Adresse wird mit der ADR-Funktion ermittelt.

z. B.: `DbAdr := ADR(PLCvar);`

Die SPS-Variable kann beliebigen Typs sein. Beispielsweise ist ein ARRAY OF WORD oder eine Datenstruktur STRUCT geeignet.

DbSize

DbSize ist die Größe der SPS-Variablen an *DbAdr* in Byte, die mit der SIZEOF-Funktion ermittelt wird.

z. B. `DbSize := SIZEOF(PLCvar);`

RemoteDbNr

RemoteDbNr enthält die Nummer des Datenbausteins beim Kommunikationspartner.

TxBuffer

Der Sendedatenpuffer *TxBuffer* ist vom Typ COMbuffer. Dieser Parameter wird für die Konfiguration nicht benötigt, muss aber als IN_OUT-Parameter übergeben werden.

RxBuffer

Der Empfangsdatenpuffer *RxBuffer* ist vom Typ COMbuffer. Dieser Parameter wird für die Konfiguration nicht benötigt, muss aber als IN_OUT-Parameter übergeben werden.

Beispiel

```

VAR
(* declare some DB
(* the type of data doesn'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 output 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,
RxBuffer:=RxBuffer );
    RK512com.DBconfig( RemoteDbNr:=10, DbAdr:=ADR(DB10), DbSize:=SIZEOF(DB10), TxBuffer:=TxBuffer,
RxBuffer:=RxBuffer );
    initialized := TRUE;
END_IF
    
```

Durch die Initialisierung im Beispiel wird erreicht, dass der Kommunikationspartner die Datenbausteine 5 und 10 beschreiben und lesen darf. Jeder Zugriff auf einen anderen Datenbaustein wird mit einem Fehler abgewiesen.

5.2 Fehlercodes von RK512

Nr.	Nr. (hex)	Fehlerkonstante	Beschreibung
4	16#04	RK512ERR_P3964ERR_ZVZ	Die Zeichenverzugszeit wurde überschritten. ZVZ = 220 ms. Die Zeichenverzugszeit ist der maximale Zeitabstand zwischen der Übertragung zweier aufeinanderfolgender Zeichen.
5	16#05	RK512ERR_P3964ERR_QVZ	Die Quittungsverzugszeit wurde überschritten. Die Gegenstelle sendet während der Telegrammabwicklung zweimal ein Quittungszeichen (DLE = 10H). QVZ = 2 s. Die erste Quittung wird am Anfang der Übertragung erwartet. Der Ausgang <i>ErrorState</i> ist im Fehlerfall "Wait_DLE_TXstart". Ursache können z. B. eine fehlerhafte physikalische Verbindung oder falsche Schnittstellenparameter sein. Die zweite Quittung wird nach der Übertragung der Nutzdaten erwartet. Im Fehlerfall ist der <i>ErrorState</i> "Wait_DLE_TXend". Ursache können in diesem Fall z. B. Datenverlust,

Nr.	Nr. (hex)	Fehlerkonstante	Beschreibung
			Datenverfälschung oder vertauschte Datenbytes eines Datenwortes (vergleiche Eingang <i>SwapWords</i>) sein.
6	16#06	RK512ERR_P3964ERR_WVZ	Die Wiederholungsverzugszeit wurde überschritten. WVZ = 4 s. Im Fehlerfall wird ein Telegramm wiederholt. Bleibt im Wiederholungsfall das Telegramm von der Gegenstelle aus, meldet der Funktionsbaustein 3964R diesen Fehler.
7	16#07	RK512ERR_P3964ERR_WRONGBCC	Prüfsummenfehler beim Datenempfang. Jedes Telegramm wird mit einer Prüfsumme versehen. Beim Datenempfang wird die berechnete Prüfsumme mit der empfangenen Prüfsumme verglichen.
9	16#09	RK512ERR_P3964ERR_COMERRTX	Schnittstellenfehler beim Senden
10	16#0A	RK512ERR_P3964ERR_COMERRRX	Schnittstellenfehler beim Empfang
11	16#0B	RK512ERR_P3964ERR_NOTXDATA	Parametrierungsfehler. Die Anzahl der zu sendenden Datenbytes <i>TxCount</i> ist Null.
120	16#78	RK512ERR_P3964ERR_NAK	Ein Telegramm wurde von der Gegenstelle mit einer Negativquittung (NAK) zurückgewiesen. Der Fehler kann u. a. in folgenden Übertragungszuständen (<i>ErrorState</i>) auftreten: <i>Wait_DLE_TXstart</i> : Die lokale SPS leitet ein Telegramm mit einem Startzeichen ein. Die Gegenstelle weist das Telegramm mit NAK ab. <i>Wait_DLE_TXend</i> : Die Gegenstelle weist das Telegramm nach Nutzdatenübertragung mit NAK ab. Ursache kann in diesem Fall z. B. ein Prüfsummenfehler in der Gegenstelle sein.
12	16#0C	RK512ERR_INVALIDDATATYPE	Ungültiger Datentyp Es wurde ein Telegramm mit einer nicht unterstützten Datentypkennung empfangen. Es können nur Datenbausteine mit der Kennung 'D' verarbeitet werden.

Nr.	Nr. (hex)	Fehlerkonstante	Beschreibung
16	16#10	RK512ERR_ERRORPROTOCOLHEADER	Fehlerhafter Protokoll-Header Der Kopf eines Datentelegramms entspricht nicht der RK512-Spezifikation
20	16#14	RK512ERR_DBNOTAVAILABLE	Datenbaustein nicht verfügbar. Es wurde ein Datenbaustein gesendet oder angefordert, der nicht verfügbar ist. Datenbausteine, auf die die Gegenstelle zugreift, müssen zunächst mit der DBconfig-Methode angemeldet werden.
22	16#16	RK512ERR_INVALIDCOMMAND	Ungültiges Kommando Es wurde ein Telegramm mit ungültiger Kommandokennung (SEND / FETCH) empfangen.
52	16#34	RK512ERR_INVALIDSIZE	Ungültige Längenangabe Einer der Parameter <i>DbSize</i> oder <i>NrOfBytes</i> ist ungültig oder es wurde ein zu langes Datentelegramm empfangen.
53	16#35	RK512ERR_INVALIDDBADR	Ungültige Datenbausteinadresse Die Datenbausteinadresse <i>DbAdr</i> ist ungültig.
54	16#36	RK512ERR_SYNCERROR	Synchronisationsfehler Ein Synchronisationsfehler kann auftreten, wenn beide Kommunikationspartner zur gleichen Zeit beginnen zu senden.
257	16#101	RK512ERR_TIMEOUT	Timeout auf RK512 Telegrammebene Ein erwartetes Reaktionstelegramm konnte nicht innerhalb der Wartezeit <i>TREA</i> empfangen werden.
258	16#102	RK512ERR_ERRORREMOTEREATEL	Fehler im Reaktionstelegramm Im Reaktionstelegramm des Kommunikationspartners wurde eine Fehlermeldung übermittelt. Diese Fehlernummer kann in der zusätzlichen Fehlerinformation <i>AUX</i> ausgelesen werden.
259	16#103	RK512ERR_INVALIDSIZEREMOTEREATEL	Das Reaktionstelegramm des Kommunikationspartners hat eine falsche Länge
260	16#104	RK512ERR_TIMEOUTREPEATSENFETCH	Timeout beim Senden Trotz mehrfacher Wiederholung konnte ein Send- oder Fetch-Telegramm nicht gesendet werden
261	16#105	RK512ERR_3964ERROR	Fehler auf der der 3964R-Protokollebene. Vorhersehbare Fehler auf dieser Ebene werden nicht mit dem allgemeinen Fehler

Nr.	Nr. (hex)	Fehlerkonstante	Beschreibung
			RK512ERR_3964ERROR, sondern mit einer detaillierten Fehlernummer ausgegeben.
262	16#106	RK512ERR_3964NOTBUSYNOTREADY	Die Übertragung eines Telegramms wurde beendet, aber nicht erfolgreich abgeschlossen.

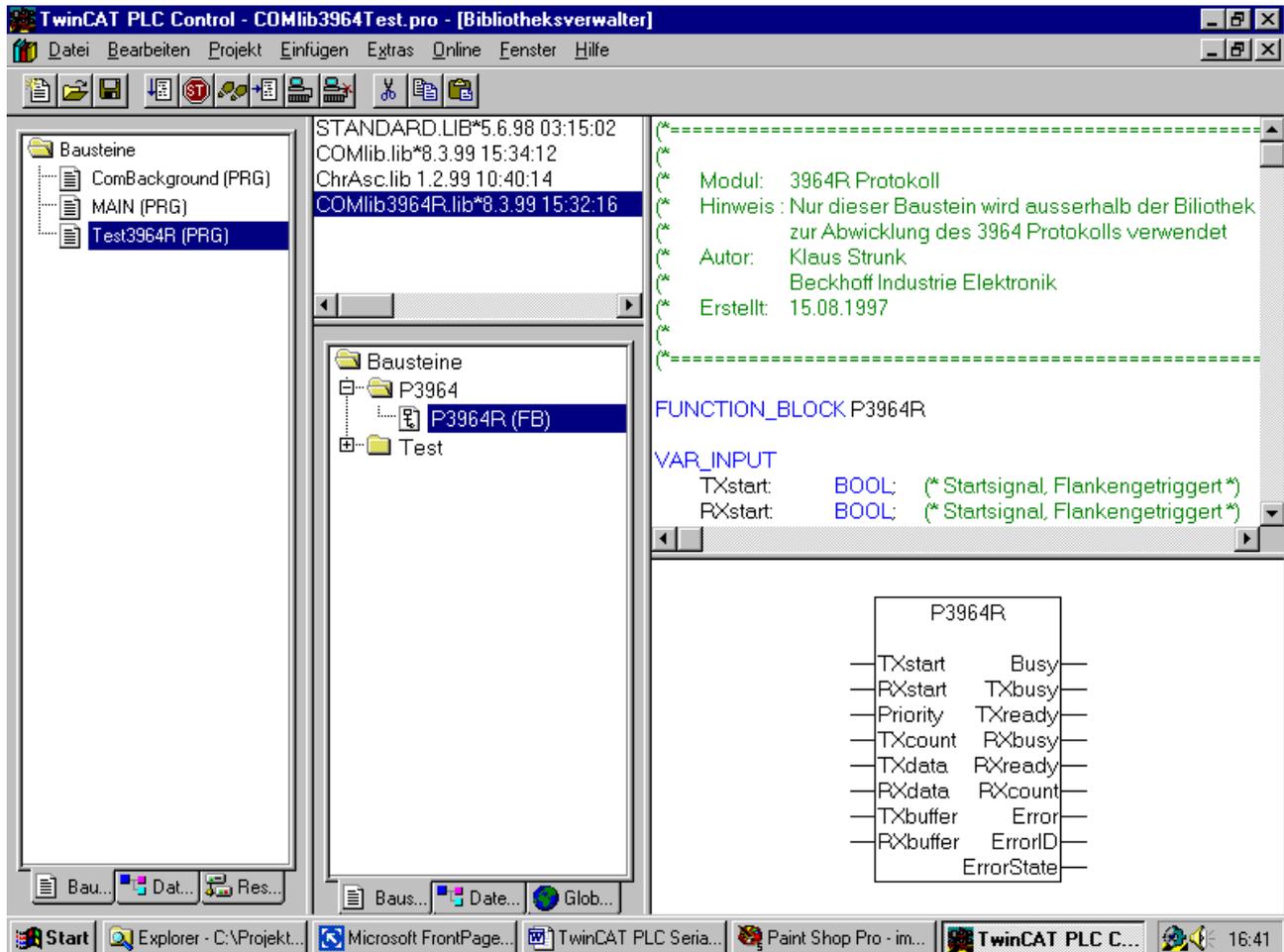
6 Beispiele

6.1 Bibliotheken einbinden

ComLib3964R.LIB einbinden

Legen Sie ein neues SPS-Projekt mit dem TwinCAT PLC Control an, um die Einbindung der Bibliothek nachzuvollziehen.

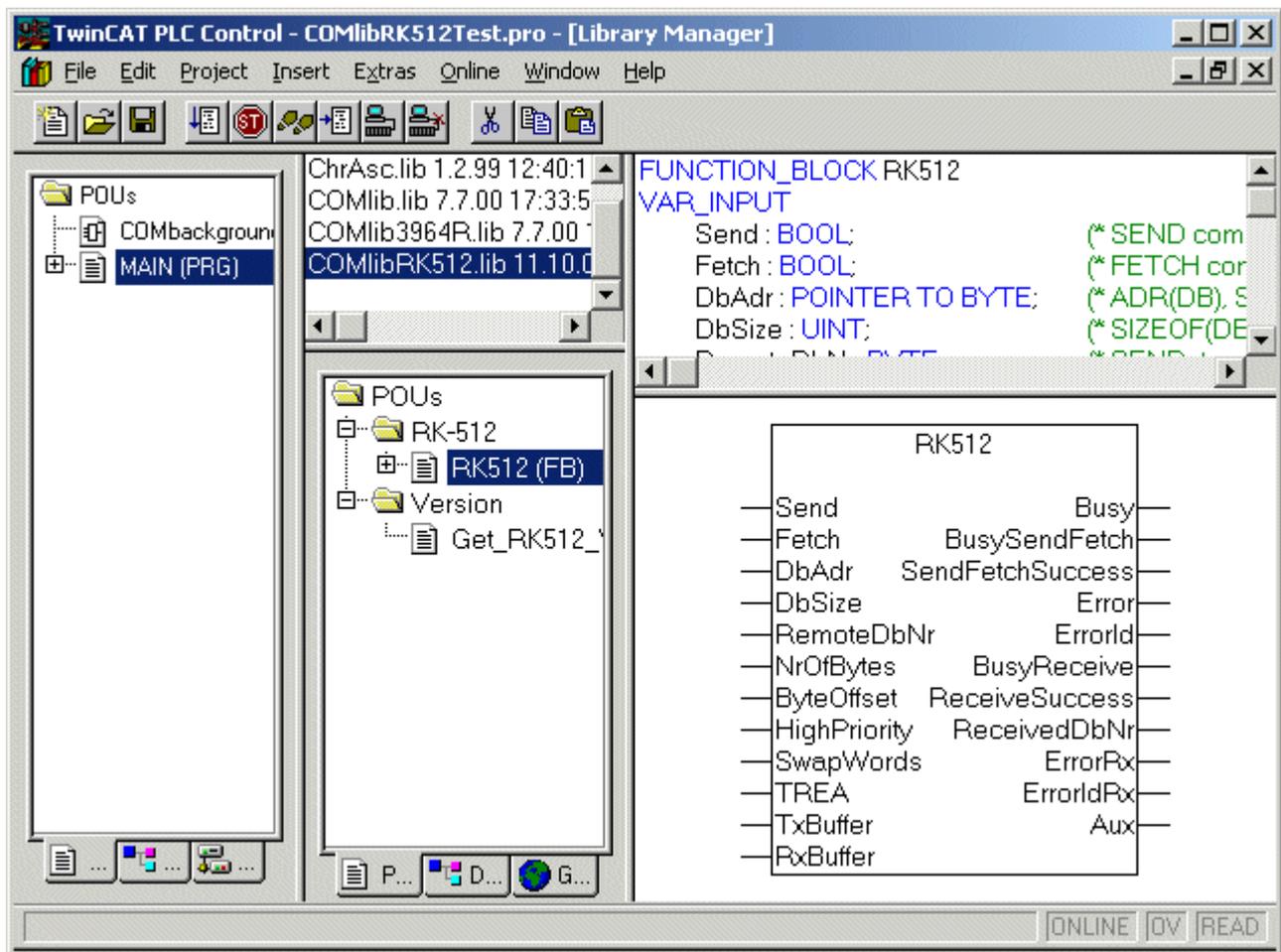
Wechseln Sie in die Bibliotheksverwaltung und fügen Sie die Bibliotheken ChrAsc.LIB und ComLib.LIB, ComLib3964R.LIB ein.



ComLibRK512.LIB einbinden

Legen Sie ein neues SPS-Projekt mit dem TwinCAT PLC Control an, um das Einbinden der Bibliothek nachzuvollziehen.

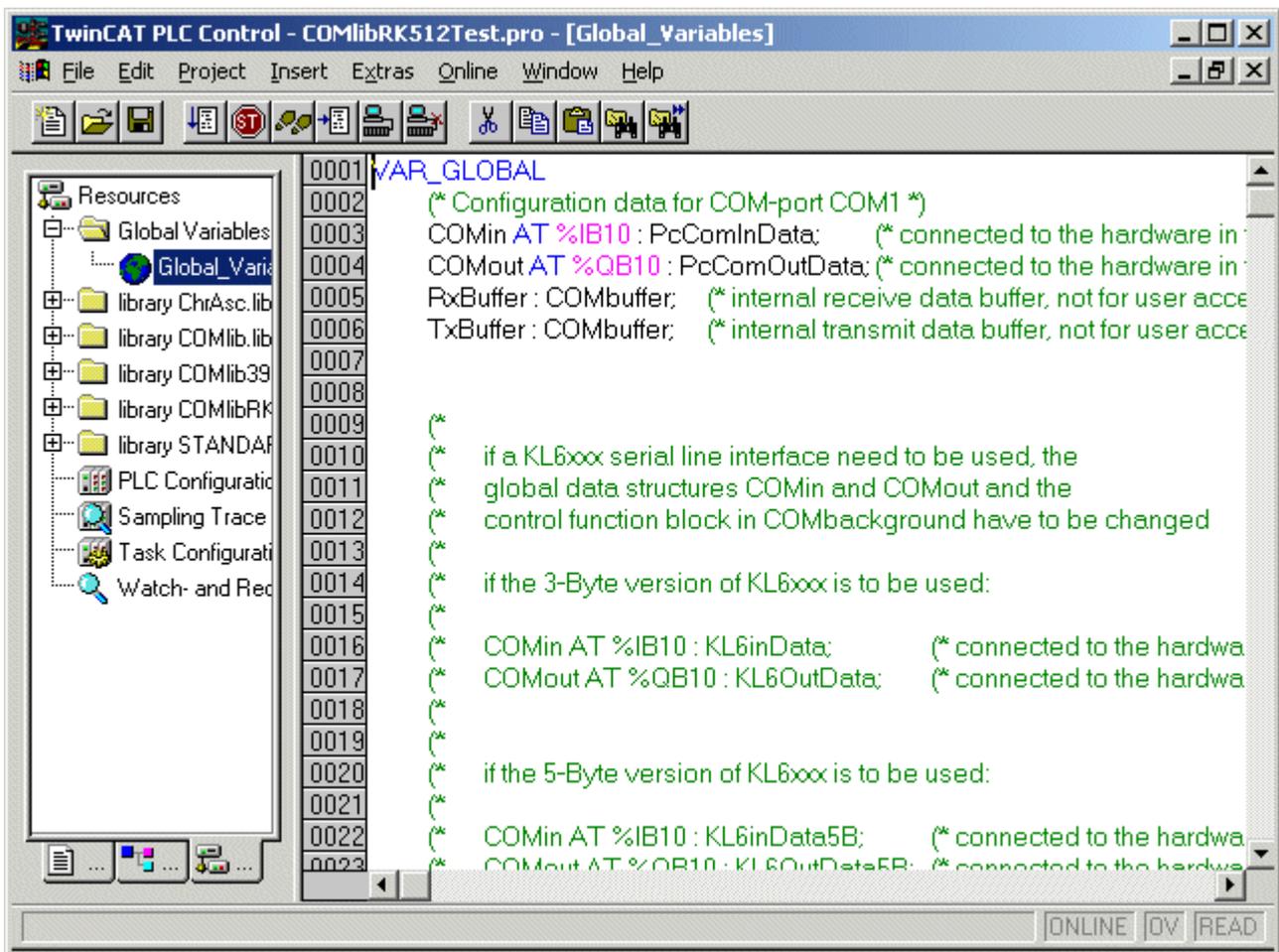
Wechseln Sie in die Bibliotheksverwaltung und fügen Sie die Bibliotheken ChrAsc.LIB, ComLib.LIB, ComLib3964R.LIB und ComLibRK512.LIB ein.



6.2 Globale Variable

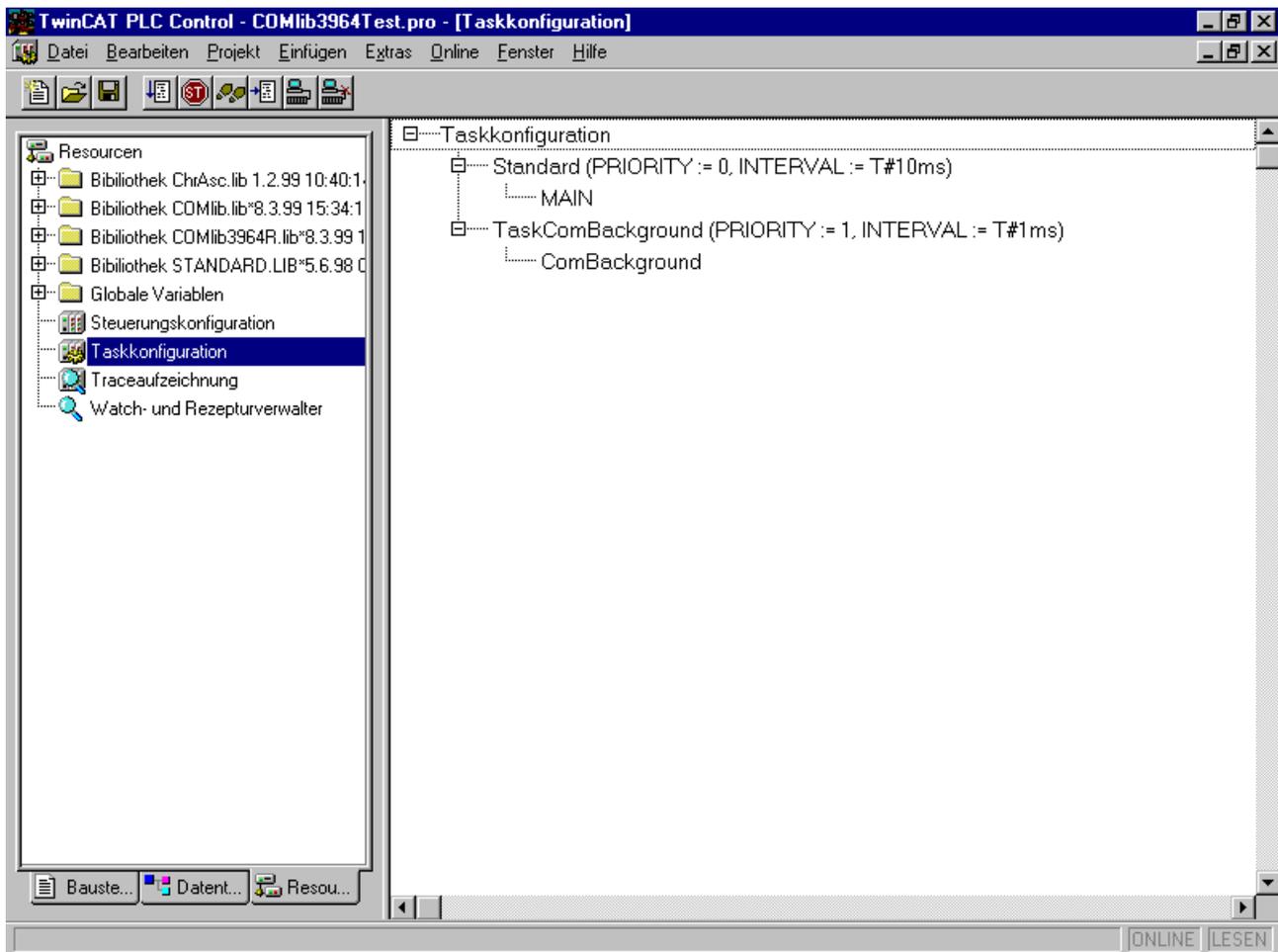
Um eine serielle Schnittstelle anzusprechen, sind vier globale Datenstrukturen notwendig. Zwei stellen die Verbindung zur Hardware in Sende- und Empfangsrichtung her. Zusätzlich werden zwei Datenpuffer als Zwischenspeicher benötigt.

Die Datenstrukturen vom Typ PcComInData bzw. PcComOutData werden im TwinCAT System Manager mit der Hardware verbunden. Lesen Sie dazu das entsprechende Kapitel in der Dokumentation zur [COMlib](#).



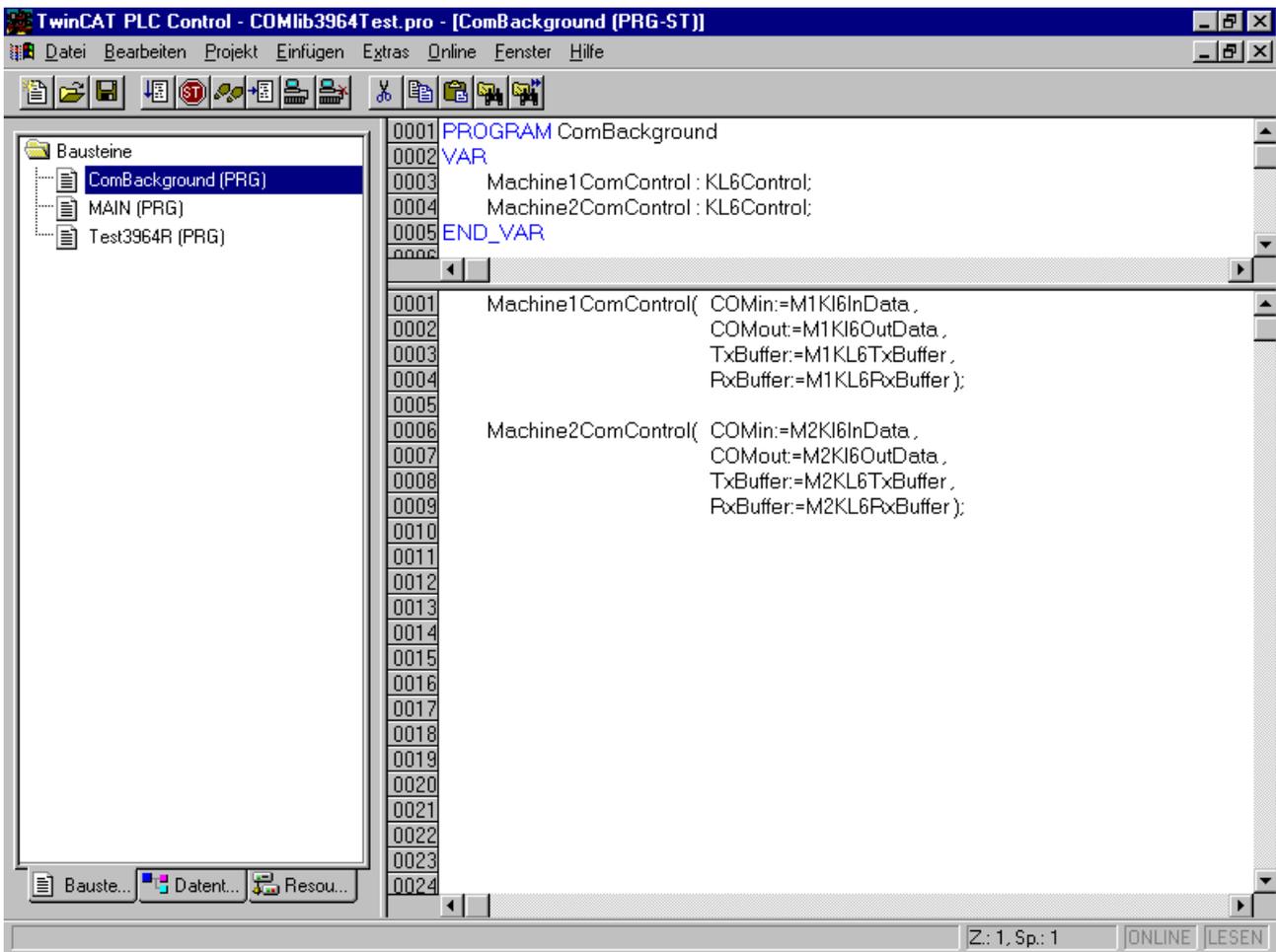
6.3 Taskkonfiguration

Für die Taskkonfiguration muß die Geschwindigkeit der seriellen Schnittstellen beachtet werden. Damit zum Beispiel bei 9600 bps an der seriellen Busklemme alle Daten tatsächlich mit dieser Geschwindigkeit verarbeitet werden können, muß der zugehörige Kommunikationsbaustein mindestens einmal pro Millisekunde aktiv werden. Dementsprechend schnell ist die Task, die den Baustein bedient, einzustellen. Im einfachsten Fall läuft das gesamte PLC-Programm in dieser schnellen Task. Wird die Task langsamer eingestellt funktioniert die Kommunikation, solange die Schnittstelle mit Hardware Handshake arbeitet, in diesem Fall jedoch nicht mit voller Geschwindigkeit. Ohne Handshake können dann Empfangsdaten verloren gehen.



6.4 Hintergrundkommunikation

Die Kommunikation zwischen der seriellen Hardware und den Datenpuffer vom Typ ComBuffer wird in einer separaten schnellen Task abgewickelt.



6.5 3964R-Kommunikation

Das Beispielprogramm sendet zur Demonstration alle 10 Sekunden Daten an eine Maschine 1 und empfängt permanent Daten von einer Maschine 2

```

0001 PROGRAM Test3964R
0002 VAR
0003     Machine1 : P3964R;

0013     (* send data *)
0014     Machine1( TXstart=M1 start,
0015              Rxstart=FALSE,
0016              Priority=TRUE,
0017              TXcount=10,
0018              TXdata=M1TxData,
0019              Rxdata=M1RxData,
0020              TXbuffer=M1KL6TxBuffer,
0021              Rxbuffer=M1KL6RxBuffer);
0022
0023     (* receive data *)
0024     Machine2( TXstart=FALSE,
0025              Rxstart=NOT Machine2.Busy,
0026              Priority=FALSE,
0027              TXcount=0,
0028              TXdata=M2TxData,
0029              Rxdata=M2RxData,
0030              TXbuffer=M2KL6TxBuffer,
0031              Rxbuffer=M2KL6RxBuffer);
0032
0033     Received := FALSE;
0034     IF Machine2.Rxready THEN
0035         Received := TRUE;
0036         (* data in RxData *)
0037         (* number of bytes is Machine2.RxCount *)
0038         M2Count := Machine2.RxCount;

```

6.6 RK512-Kommunikation

Das Beispielprogramm definiert drei WORD-Arrays, um sie als Datenbausteine zu verwenden. Die Datenbausteine 5 und 10 werden während der Initialisierung mit einem Aufruf der Methode DBconfig für den Zugriff durch den Kommunikationspartner bekannt gemacht. Der Funktionsbausteins RK512 verhält passiv und ist bereit Datentelegramme von der Gegenstelle anzunehmen und zu beantworten. Im Testprogramm kann das aktive Senden oder Holen von Daten durch beschreiben der Variablen *Send* bzw. *Fetch* gestartet werden.

TwinCAT PLC Control - COMlibRK512Test.pro - [MAIN (PRG-ST)]

File Edit Project Insert Extras Online Window Help

POUs
 COMbackground
 MAIN (PRG)
 CallRK512
 CreateTestC

```

0001 PROGRAM MAIN
0002 VAR
0003     (* declare some DB
0004     (* the type of data doesn'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 *)
0009
0010     (* Initialize a list of DB that may be accessed by the remote PLC.
0011     (* The remote PLC is only able to access data of DBs that have been
0012     (* To define each DB, a call to the DBconfig method of the RK512 fu
0013     (* block is used.
0014     (*
0015     IF NOT initialized THEN
0016         RK512com.DBconfig( RemoteDbNr:=5, DbAdr:=ADR(DB5), DbSi
0017         RK512com.DBconfig( RemoteDbNr:=10, DbAdr:=ADR(DB10), Db
0018         HighPriority := FALSE; (* should be the opposite of the remote Pl
0019         SwapWords := TRUE;
0020         initialized := TRUE;
0021     END_IF
    
```

Lin.: 1, Col.: 1 ONLINE OV READ

TwinCAT PLC Control - COMlibRK512Test.pro - [CallRK512 (FBD)]

File Edit Project Insert Extras Online Window Help

POUs
 COMbackground
 MAIN (PRG)
 CallRK512
 CreateTestC

```

0001 (* Trigger Send or Fetch manually ! *)
    
```

ONLINE OV READ

Mehr Informationen:
www.beckhoff.de/ts6341

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.de
www.beckhoff.de

