

SIEMENS

Ingenuity for life

24/7

NEWS

Industry Online Support

Home

Kommunikations- bibliothek LCom für SIMOTION

SIMOTION

<https://support.industry.siemens.com/cs/ww/de/view/48955385>

Siemens
Industry
Online
Support



Rechtliche Hinweise

Nutzung der Anwendungsbeispiele

In den Anwendungsbeispielen wird die Lösung von Automatisierungsaufgaben im Zusammenspiel mehrerer Komponenten in Form von Text, Grafiken und/oder Software-Bausteinen beispielhaft dargestellt. Die Anwendungsbeispiele sind ein kostenloser Service der Siemens AG und/oder einer Tochtergesellschaft der Siemens AG („Siemens“). Sie sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit und Funktionsfähigkeit hinsichtlich Konfiguration und Ausstattung. Die Anwendungsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern bieten lediglich Hilfestellung bei typischen Aufgabenstellungen. Sie sind selbst für den sachgemäßen und sicheren Betrieb der Produkte innerhalb der geltenden Vorschriften verantwortlich und müssen dazu die Funktion des jeweiligen Anwendungsbeispiels überprüfen und auf Ihre Anlage individuell anpassen.

Sie erhalten von Siemens das nicht ausschließliche, nicht unterlizenzierbare und nicht übertragbare Recht, die Anwendungsbeispiele durch fachlich geschultes Personal zu nutzen. Jede Änderung an den Anwendungsbeispielen erfolgt auf Ihre Verantwortung. Die Weitergabe an Dritte oder Vervielfältigung der Anwendungsbeispiele oder von Auszügen daraus ist nur in Kombination mit Ihren eigenen Produkten gestattet. Die Anwendungsbeispiele unterliegen nicht zwingend den üblichen Tests und Qualitätsprüfungen eines kostenpflichtigen Produkts, können Funktions- und Leistungsmängel enthalten und mit Fehlern behaftet sein. Sie sind verpflichtet, die Nutzung so zu gestalten, dass eventuelle Fehlfunktionen nicht zu Sachschäden oder der Verletzung von Personen führen.

Haftungsausschluss

Siemens schließt seine Haftung, gleich aus welchem Rechtsgrund, insbesondere für die Verwendbarkeit, Verfügbarkeit, Vollständigkeit und Mangelfreiheit der Anwendungsbeispiele, sowie dazugehöriger Hinweise, Projektierungs- und Leistungsdaten und dadurch verursachte Schäden aus. Dies gilt nicht, soweit Siemens zwingend haftet, z.B. nach dem Produkthaftungsgesetz, in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der schuldhaften Verletzung des Lebens, des Körpers oder der Gesundheit, bei Nichteinhaltung einer übernommenen Garantie, wegen des arglistigen Verschweigens eines Mangels oder wegen der schuldhaften Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegen oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist mit den vorstehenden Regelungen nicht verbunden. Von in diesem Zusammenhang bestehenden oder entstehenden Ansprüchen Dritter stellen Sie Siemens frei, soweit Siemens nicht gesetzlich zwingend haftet.

Durch Nutzung der Anwendungsbeispiele erkennen Sie an, dass Siemens über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden kann.

Weitere Hinweise

Siemens behält sich das Recht vor, Änderungen an den Anwendungsbeispielen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in den Anwendungsbeispielen und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Ergänzend gelten die Siemens Nutzungsbedingungen (<https://support.industry.siemens.com>).

Securityhinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.

Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z.B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.

Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter:

<https://www.siemens.com/industrialsecurity>.

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter: <https://www.siemens.com/industrialsecurity>.

Inhaltsverzeichnis

Rechtliche Hinweise	2
1 Bibliotheksübersicht	4
1.1 Verschiedene Anwenderszenarien	5
1.2 Hard- und Softwarevoraussetzungen.....	9
2 FBLCom_Communication.....	10
2.1 Bausteinparameter	11
2.2 Funktionsweise.....	13
2.2.1 Kommunikationsaufbau.....	13
2.2.2 Kommunikationsabbau.....	18
2.2.3 Daten senden	19
2.2.4 Daten empfangen.....	23
2.2.5 Lebenszeichen	29
2.2.6 Uhrzeitsynchronisation	30
2.3 Fehlerrückgabe und Diagnose	32
2.3.1 Statusmeldungen	34
2.4 Übersicht der Strukturen und Konstanten	44
3 Arbeiten mit der Bibliothek.....	51
3.1 Bibliotheksaufbau	51
3.2 Integration der Bibliothek in SCOUT	52
4 Literaturhinweise	53
5 Ansprechpartner	53
6 Historie.....	53

1 Bibliotheksübersicht

Was erhalten Sie?

Das vorliegende Dokument beschreibt die Bausteinbibliothek **LCom**.

Die Bausteinbibliothek **LCom** kann für die Standard TCP-Kommunikation genutzt werden. Aufgrund der großen Verbreitung des TCP-Standards ist somit eine einfache und schnelle Realisierung des Datenaustauschs zwischen verschiedenen Geräten möglich.

Zusätzlich sind weitere Kommunikationsfunktionalitäten über TCP hinaus in der Bausteinbibliothek implementiert. Diese erweiterten Funktionalitäten werden über ein definiertes Protokoll (im Folgenden als LCom-Protokoll bezeichnet) bereitgestellt.

Die Bausteinbibliothek **LCom** bietet dem Anwender folgende Vorteile:

- Der Anwender muss die Systemfunktionen (Bsp.: tcpOpenClient/ tcpOpenServer, tcpSend, tcpReceive, tcpCloseConnection) zum Aufbauen, Senden/Empfangen und Abbauen einer Verbindung nicht kennen und programmieren.
- Bei Fehlern/Störungen schließt der Funktionsbaustein automatisch die Verbindung und versucht diese neu aufzubauen.
- Zyklische Datenübertragung, der Anwender definiert die Zykluszeit.
- Einmalige Datenübertragung.
- Konsistente Datenübertragung.
- Abgleich der Kommunikationsparameter.
- Überwachung der Kommunikationsverbindung durch zyklisches Senden eines Lebenszeichens (schnelle Reaktionszeiten bei Verbindungsausfall). Bei reiner TCP-Kommunikation liegt diese typischerweise im Sekundenbereich.
- Datensätze mit definierter Länge bis zu 64 kB mit LCom-Protokoll V1.
- Datensätze mit definierter Länge bis zu 16 MB mit LCom-Protokoll V2.
- Rückmeldung vom Sender/Empfänger über erfolgreich übertragene Daten auf Applikationsebene.
- Einfache Uhrzeitsynchronisation.
- Der Funktionsbaustein enthält eine Datenstruktur zur Diagnose.
- Der Funktionsbaustein ist multiinstanzfähig.

Gültigkeitsbereich der Bibliothek

- SIMOTION SCOUT ab V4.4
- SCOUT TIA ab V4.4
- SIMOTION D4x5-2
- SIMOTION D410
- SIMOTION C230, C240
- SIMOTION P320, 350

1.1 Verschiedene Anwenderszenarien

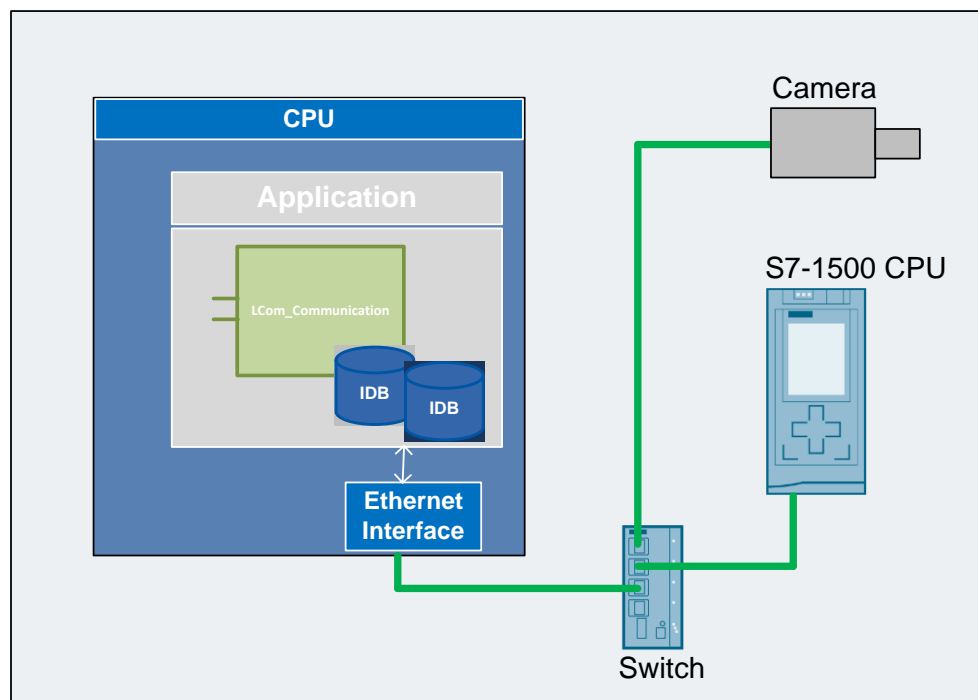
Einsatzmöglichkeiten für die Verwendung der Bibliothek LCom

Mit dem Funktionsbaustein **FBLCom_Communication** der Bibliothek **LCom** wird eine Punkt-zu-Punkt Vollduplexverbindung über Industrial Ethernet, basierend auf dem TCP-Standard, aufgebaut.

Der Funktionsbaustein kann für die Standard TCP-Kommunikation zu anderen Geräten (Bsp.: Kamera, Controller) genutzt werden.

Da der Funktionsumfang des Transportprotokolls TCP für viele Anwendungen im Automatisierungsbereich nicht ausreicht, ist in der Bausteinbibliothek **LCom** ein eigenes Protokoll (LCom-Protokoll) definiert und implementiert. Mit dem LCom-Protokoll werden zusätzliche Kommunikationsfunktionalitäten ermöglicht.

Abbildung 1-1 Anwendungsszenarien

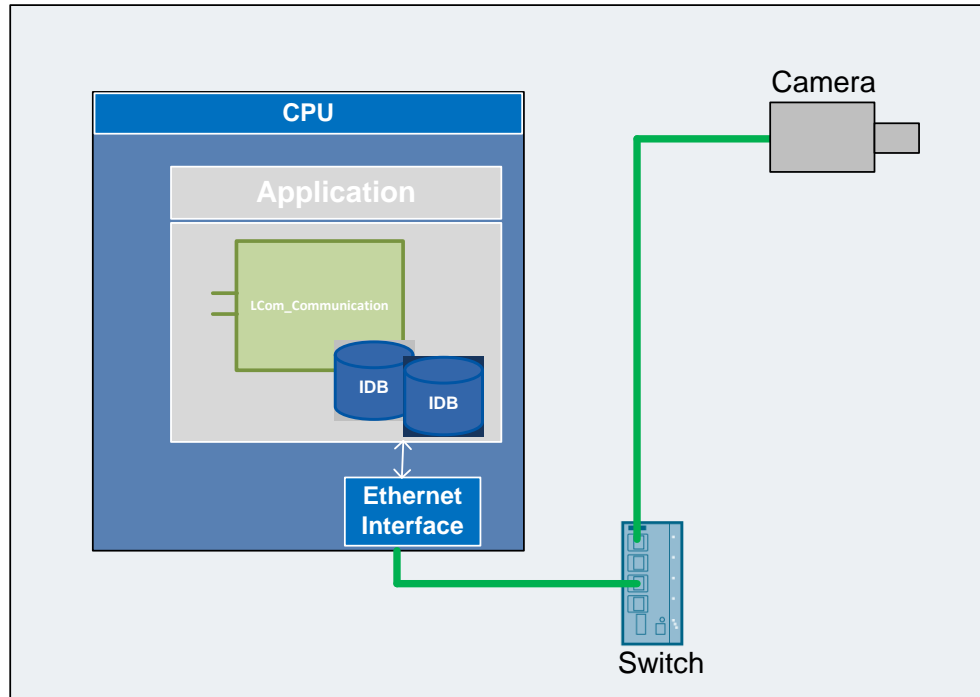


Nachfolgend sind einige Szenarien für einen möglichen Einsatz der Bibliothek **LCom** aufgezeigt:

Szenario 1

Der Funktionsbaustein **FBLCom_Communication** wird für die Standard TCP-Kommunikation genutzt. Das Transportprotokoll TCP garantiert die Übertragung eines endlosen Datenstroms. TCP ist nicht paketorientiert und ermöglicht daher nicht die Übertragung von Datensätzen einer definierten Gesamtlänge.

Abbildung 1-2 LCom_Communication - Standard TCP Kommunikation



Die LCom Bibliothek bietet folgende Vorteile:

- Der Anwender muss die Systemfunktion (Bsp.: tcpOpenClient/ tcpOpenServer, tcpSend, tcpReceive, tcpCloseConnection) zum Aufbauen, Senden/Empfangen und Abbauen einer Verbindung nicht kennen und programmieren.
- Bei Fehlern/Störungen schließt der Funktionsbaustein automatisch die Verbindung und versucht diese neu aufzubauen.
- Zyklische Datenübertragung, der Anwender definiert die Zykluszeit.
- Einmalige Datenübertragung.
- Rückmeldung vom Sender/Empfänger über erfolgreich übertragene Daten auf Applikationsebene.
- Der Funktionsbaustein enthält eine Datenstruktur zur Diagnose.
- Der Funktionsbaustein ist multiinstanzfähig.

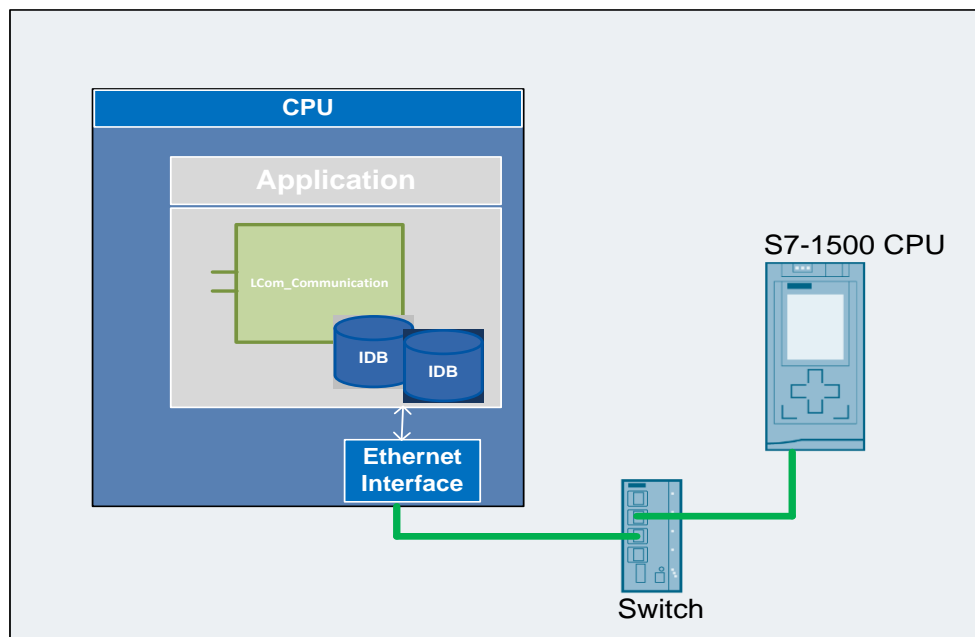
Szenario 2

Bei der Verwendung des LCom-Protokolls können Datensätze einer definierten Gesamtlänge gesendet und vom Partner konsistent empfangen werden. Um einen Verbindungsabbruch schnell zu erkennen, wird ein zyklisches Lebenszeichen gesendet. Die Zykluszeit des periodischen Lebenszeichens wird vom Anwender vorgegeben.

Zur Uhrzeitsynchronisation zweier Steuerungen kann die aktuelle Uhrzeit einer Steuerung an den Partner geschickt und dort als Systemzeit übernommen werden.

Um die zusätzlichen Kommunikationsfunktionalitäten zu nutzen, muss der Kommunikationspartner (Bsp.:S7-1500 CPU) das LCom-Protokoll unterstützen.

Abbildung 1-3 LCom_Communication – LCom-Protokoll



Die LCom Bibliothek bietet folgende Vorteile:

- Der Anwender muss die Systemfunktion (Bsp.: tcpOpenClient/ tcpOpenServer, tcpSend, tcpReceive, tcpCloseConnection) zum Aufbau, Senden/Empfangen und Abbauen einer Verbindung nicht kennen und programmieren.
- Bei Fehlern/Störungen schließt der Funktionsbaustein automatisch die Verbindung und versucht diese neu aufzubauen.
- Zyklische Datenübertragung, der Anwender definiert die Zykluszeit.
- Einmalige Datenübertragung.
- Konsistente Datenübertragung.
- Abgleich der Kommunikationsparameter.
- Überwachung der Kommunikationsverbindung durch zyklisches Senden eines Lebenszeichens (schnelle Reaktionszeiten bei Verbindungsausfall). Bei reiner TCP-Kommunikation liegt diese typischerweise im Sekundenbereich.
- Datensätze mit definierter Länge bis zu 64 kB mit LCom-Protokoll V1.
- Datensätze mit definierter Länge bis zu 16 MB mit LCom-Protokoll V2.

- Rückmeldung vom Sender/Empfänger über erfolgreich übertragene Daten auf Applikationsebene.
- Einfache Uhrzeitsynchronisation.
- Der Funktionsbaustein enthält eine Datenstruktur zur Diagnose.
- Der Funktionsbaustein ist multiinstanzfähig.

1.2 Hard- und Softwarevoraussetzungen

Voraussetzungen für diese Bibliothek

Um die Funktionalität der hier beschriebenen Bibliothek nutzen zu können, sind nachfolgend genannte Hard- und Softwarevoraussetzungen einzuhalten:

Hardware

Tabelle 1-1 Hardware Komponenten

Nr.	Komponente	Firmware-Version
1.	SIMOTION D4x5-2	Firmware-Version ab V4.4
2.	SIMOTION D410	
3.	SIMOTION C230, C240	
4.	SIMOTION P320, P350	

Software

Tabelle 1-2 Software Komponenten

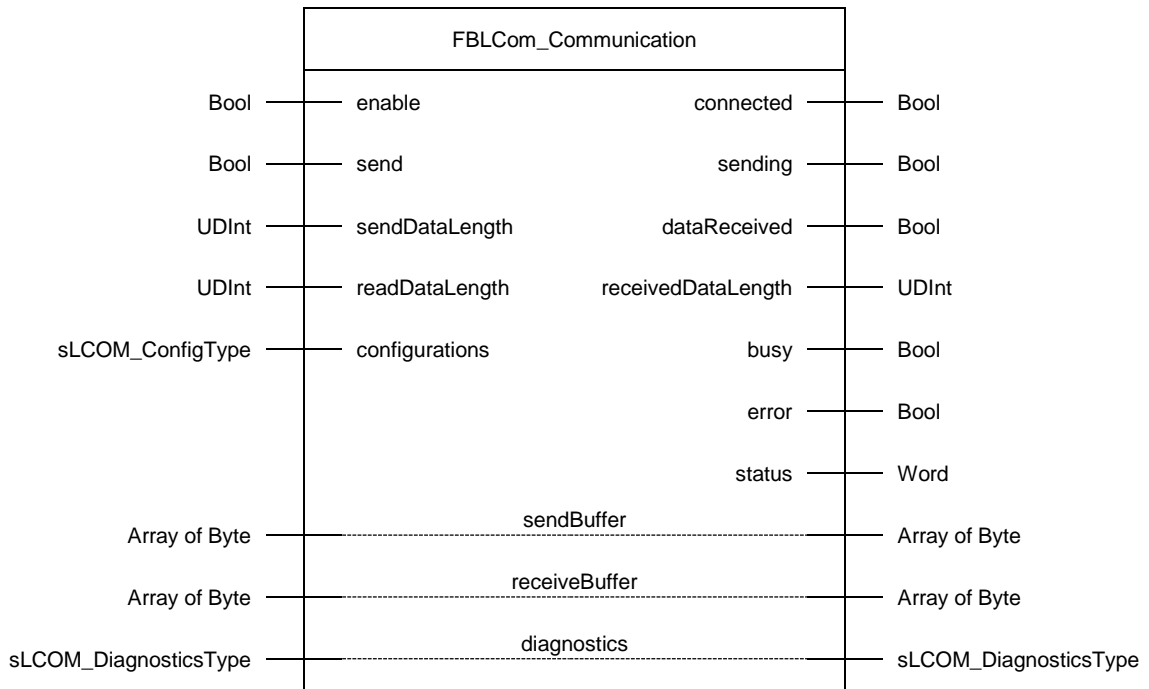
Nr.	Komponente	Artikelnummer	Hinweis
5.	SIMOTION SCOUT	-	-
6.	SCOUT TIA	-	-

2 FBLCom_Communication

Was steht hier?

In diesem Kapitel werden die Bausteinparameter und die Funktionsweise des Funktionsbausteins **FBLCom_Communication** beschrieben.

Abbildung 2-1 Blockbild FBLCom_Communication



2.1 Bausteinparameter

Die nachfolgenden Tabellen listen alle Bausteinparameter des Funktionsbausteins **FBCom_Communication** auf.

Eingangsparameter

Tabelle 2-1 Eingangsparameter FBCom_Communication

Parameter	Datentyp	Kommentar
enable	Bool	TRUE: Startet den FB mit einer steigenden Flanke. Es wird automatisch die Verbindung zum Partner aufgebaut. FALSE (default): Mit einer fallenden Flanke, wird die Verbindung abgebaut.
send	Bool	TRUE: Senden der Daten, die am Eingangsparameter <i>sendBuffer</i> verschaltet sind. FALSE (default): Daten die am Eingangsparameter <i>sendBuffer</i> verschaltet sind, werden nicht versendet.
sendDataLength	UDInt	Zu sendende Datenlänge in Bytes. (default: 4294967295)
readDataLength	UDInt	Ohne LCom-Protokoll, (<i>configurations.sConnection.u8ComService = LCOM_TCP_CONNECTION</i>): 0: Daten aus dem TCP Puffer werden nicht gelesen. FB erhält somit keine Daten vom Partner. 1 .. 4294967294: Anzahl der Bytes, die an der Schnittstelle empfangen werden müssen, bevor die Ausgangsvariable <i>dataReceived = TRUE</i> ist. 4294967295, 16#FFFFFFFF (default): Alle Daten, die an der Schnittstelle zur Verfügung stehen, werden gelesen (Ad-Hoc-Modus).
		Mit LCom-Protokoll, (<i>configurations.sConnection.u8ComService = LCOM_LCOM_CONNECTION</i>): 0: Daten aus dem TCP Puffer werden nicht gelesen. Der FB erhält somit keine Daten vom Partner. 1 .. 4294967295: Keine Relevanz für das Empfangsverhalten.
configurations	sLCOM_ConfigType	FB Konfiguration

Ausgangsparameter

Tabelle 2-2 Ausgangsparameter FBCom_Communication

Parameter	Datentyp	Beschreibung
connected	Bool	Ohne LCom-Protokoll, (<i>configurations.sConnection.u8ComService = LCOM_TCP_CONNECTION</i>): TRUE: Die TCP-Verbindung zum Partner ist aufgebaut. FALSE: Die TCP-Verbindung zum Partner ist nicht aufgebaut.
		Mit LCom-Protokoll, (<i>configurations.sConnection.u8ComService = LCOM_LCOM_CONNECTION</i>): TRUE: Die TCP-Verbindung zum Partner ist aufgebaut und die Konfigurationsdaten wurden erfolgreich ausgetauscht. FALSE: Die Konfigurationsdaten wurden nicht erfolgreich ausgetauscht bzw. die Verbindung besteht nicht.
sending	Bool	TRUE: Senden der Daten, die am Eingangsparameter <i>sendBuffer</i> verschaltet sind. Sendedaten nicht verändern. FALSE: Daten die am Eingangsparameter <i>sendBuffer</i> verschaltet sind, werden nicht versendet.
dataReceived	Bool	TRUE: Neue Daten wurden empfangen. Der Wert steht für einen Zyklus an. FALSE: Es stehen für den Anwender keine Daten bereit.
receivedDataLength	UDInt	Liefert die empfangene Datenlänge in Bytes.
busy	Bool	TRUE: Die Bausteinverarbeitung erfolgt automatisch. Es ist kein Eingriff vom Anwender erforderlich. FALSE: Es erfolgt keine Bausteinverarbeitung. Es ist ein Eingriff vom Anwender erforderlich. Übergabeparameter sind ggf. fehlerhaft. Genauere Information liefert der Ausgabeparameter <i>status</i> bzw. der Diagnosepuffer.
error	Bool	TRUE: Es ist ein Fehler aufgetreten. Genauere Information liefert der Ausgabeparameter <i>status</i> bzw. der Diagnosepuffer. FALSE: Es ist kein Fehler aufgetreten.
status	Word	FB Status, siehe Kapitel 2.3 Fehlerrückgabe und Diagnose

Durchgangsparameter

Tabelle 2-3 Durchgangsparameter FBCom_Communication

Parameter	Datentyp	Beschreibung
sendBuffer	Array of Byte	Sendedaten
receiveBuffer	Array of Byte	Empfangsdaten
diagnostics	sLCOM_DiagnosticsType	Diagnoseinformationen

2.2 Funktionsweise

2.2.1 Kommunikationsaufbau

Bevor der FB die Verbindung zum Partner aufbauen kann, müssen vorher in der Anwenderapplikation die Verbindungsparameter parametrieren werden. Alle relevanten Informationen sind im Eingangsparameter *configurations* enthalten. Für die bessere Übersichtlichkeit der Parametrierung wurde dafür eine Struktur *sConnectionType* definiert.

Tabelle 2-4 Verbindungsparameter

Parameter	Datentyp	Beschreibung
<i>sConnectionType</i>	Struct	
<i>u8ComService</i>	USInt	Konfiguration des Kommunikationsprotokolls. 1, LCOM_TCP_CONNECTION: Das Standard TCP Protokoll wird verwendet. 2, LCOM_LCOM_CONNECTION(default): Das LCom Protokoll wird verwendet. Mit dem LCom Protokoll werden zusätzliche Kommunikationsfunktionalitäten ermöglicht
<i>bolsClient</i>	Bool	TRUE: Aktiver Verbindungsaufbau als TCP-Client. FALSE (default): Passiver Verbindungsaufbau als TCP-Server.
<i>u16LocalPort</i>	UInt	Lokaler Port (default: 3456). Siehe Systemfunktion <i>tcpOpenClient/tcpOpenServer</i> .
<i>u16PartnerPort</i>	UInt	Nur bei aktivem Verbindungsaufbau (<i>configurations.sConnection.bolsClient = TRUE</i>): Port der Partnerseite (default: 3456) Siehe Systemfunktion <i>tcpOpenClient</i> .
<i>au8PartnerIP</i>	Array of USInt	Bei aktivem Verbindungsaufbau (<i>configurations.sConnection.bolsClient = TRUE</i>): IP Adresse vom Partner. Bei passivem Verbindungsaufbau (<i>configurations.sConnection.bolsClient = FALSE</i>) und zusätzlich konfigurierter Verbindungspartner (<i>configurations.boAcceptUnknownPartner = FALSE</i>): Es wird nur diese Partner IP Adresse bei dem Verbindungsaufbau akzeptiert.

Parameter	Datentyp	Beschreibung
boAcceptUnknownPartner	Bool	Nur bei passivem Verbindungsaufbau (<i>configurations.sConnection.bolsClient = FALSE</i>): TRUE (default): Vorher nicht konfigurierter Verbindungspartner wird akzeptiert. FALSE: Es wird nur der konfigurierte Verbindungspartner akzeptiert.
tLifeSignCycleTime	Time	Nur mit LCom-Protokoll, (<i>configurations.sConnection.u8ComService = LCOM_LCOM_CONNECTION</i>): Lebenszeichenzyklus T#1ms .. T#24d20h30m20s630ms (default: 1s).

Konfigurationsparameter in der Applikation

```
//connection configuration
//1: native TCP, 2: with LCOM protocol
gspComLComParameter.sConnection.u8ComService := LCOM_LCOM_CONNECTION;
//FALSE: passive (server), TRUE: active connection establishment (client)
gspComLComParameter.sConnection.boIsClient := FALSE;
//local port number
gspComLComParameter.sConnection.ul6LocalPort := 3456;
//remote port number, use only in case of client or specified partner
gspComLComParameter.sConnection.ul6PartnerPort := 3456;
//IP address xxx.xxx.xxx.xxx = [1].[2].[3].[4]
gspComLComParameter.sConnection.au8PartnerIP[0] := 192;
gspComLComParameter.sConnection.au8PartnerIP[1] := 168;
gspComLComParameter.sConnection.au8PartnerIP[2] := 150;
gspComLComParameter.sConnection.au8PartnerIP[3] := 18;
//TRUE: accept requests from all partners, FALSE: only specified partner is allowed
gspComLComParameter.sConnection.boAcceptUnknownPartner := TRUE;
//life sign cycle time (T#1ms...T#+24d20h31m23s647ms)
gspComLComParameter.sConnection.tLifeSignCycleTime := T#1000ms;
```

Nach einer steigenden Flanke des booleschen Eingangs *enable* werden die Konfigurationsparameter übernommen und versucht eine TCP-Verbindung aufzubauen. Wird das LCom-Protokoll verwendet, wird auch die LCom-Protokollversion überprüft. Mit der Versionserkennung wird automatisch kontrolliert, mit welcher Protokollversion der FB mit dem Partner kommunizieren kann (Protokollversion V1 oder Protokollversion V2).

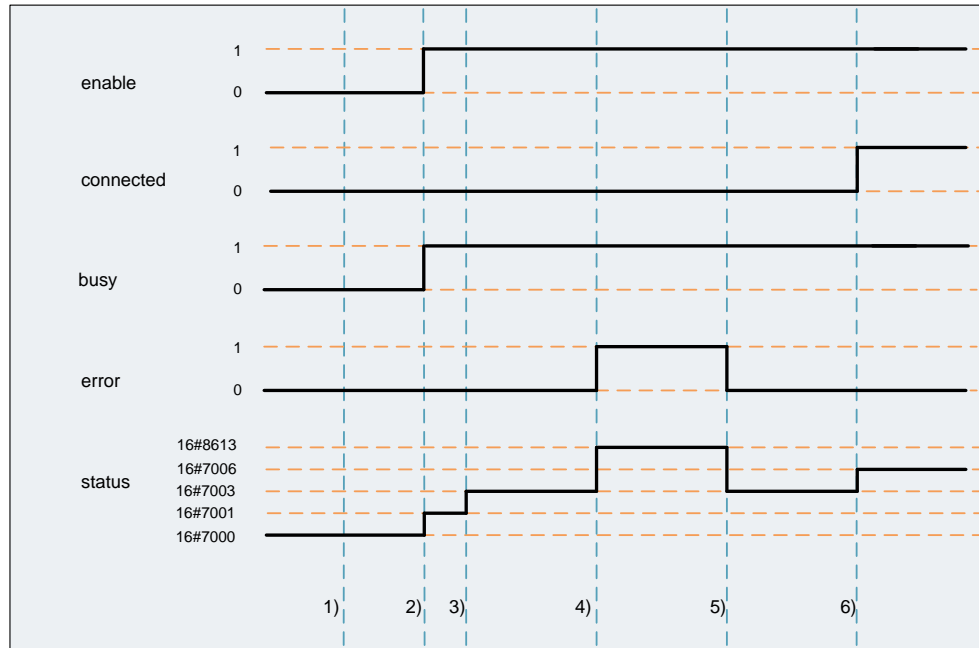
Steht dem Partner die LCom-Protokollversion V2 nicht zur Verfügung, wird der FB automatisch die TCP Verbindung schließen und neu aufbauen und dann die LCom-Protokollversion V1 verwenden.

Funktionsverläufe

Die nachfolgenden Abbildungen sollen den Kommunikationsaufbau verdeutlichen:

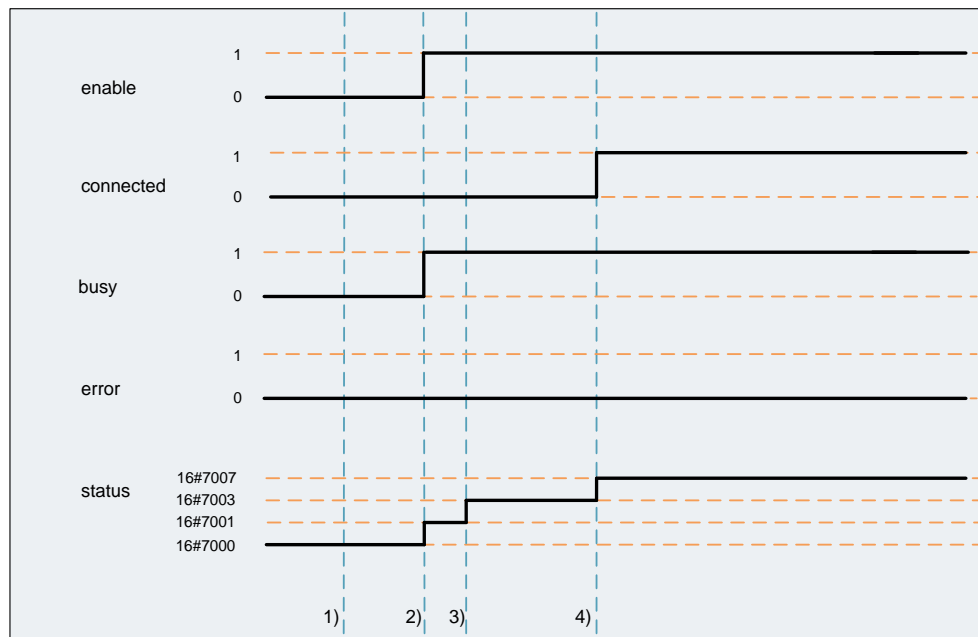
- Verbindungsaufbau mit LCom Protokoll V1
- Verbindungsaufbau mit LCom Protokoll V2
- Verbindungsaufbau ohne LCom Protokoll

Abbildung 2-2 Aufbau mit LCom-Protokoll V1



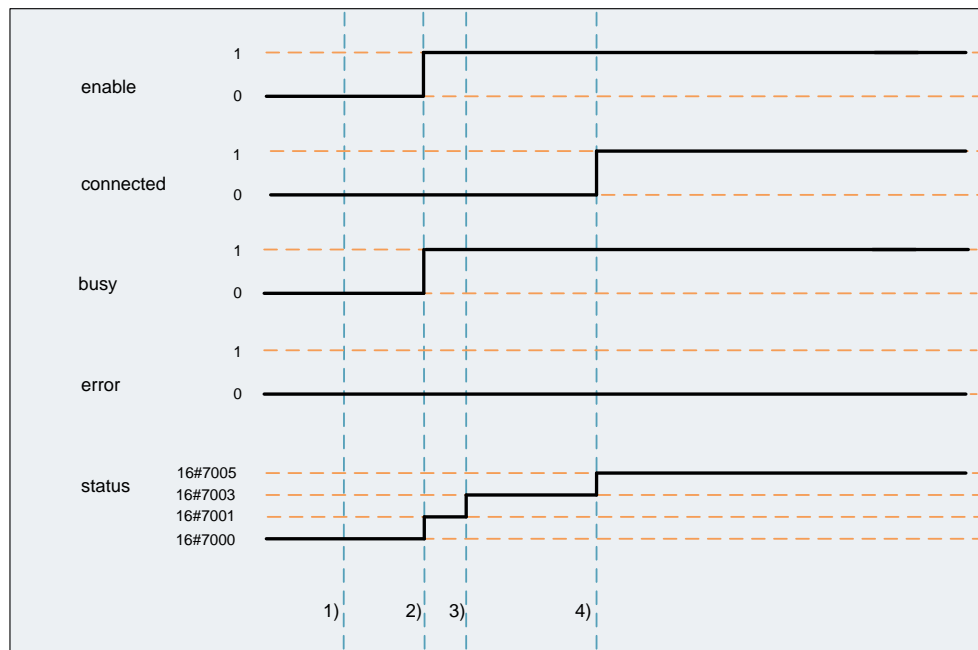
1. Der FB wurde noch nicht gestartet.
2. Nach der L/H Flanke an `enable` beginnt die Bausteinbearbeitung. Die Ausgangsvariable `busy` wird TRUE.
3. Der FB versucht als TCP-Client eine Verbindung zum Partner aufzubauen und die Konfigurationsdaten nach LCom Protokoll V2 auszutauschen (`status = LCOM_STATUS_CLIENT_TRY_CONNECT`).
4. Der FB hat erkannt, dass der Partner nicht das LCom-Protokoll V2 unterstützt und setzt die Ausgangsvariable `error = TRUE` und schließt selbständig die Verbindung (`status = LCOM_ERR_WRONG_LCOM_HEADER`).
5. Der FB versucht als TCP-Client erneut eine Verbindung zum Partner aufzubauen und die Konfigurationsdaten nach LCom-Protokoll V1 auszutauschen (`status = LCOM_STATUS_CLIENT_TRY_CONNECT`).
6. Die Ausgangsvariable `connected` wird TRUE, wenn die Konfigurationsdaten erfolgreich mit dem Partner ausgetauscht wurden.

Abbildung 2-3 Aufbau mit LCom-Protokoll V2



1. Der FB wurde noch nicht gestartet.
2. Nach der L/H Flanke an *enable* beginnt die Bausteinbearbeitung. Die Ausgangsvariable *busy* wird TRUE.
3. Der FB versucht als TCP-Client eine Verbindung zum Partner aufzubauen und die Konfigurationsdaten nach LCom Protokoll V2 auszutauschen (*status* = LCOM_STATUS_CLIENT_TRY_CONNECT).
4. Die Ausgangsvariable *connected* wird TRUE, wenn die Konfigurationsdaten erfolgreich mit dem Partner ausgetauscht wurden.

Abbildung 2-4 Aufbau ohne LCom-Protokoll



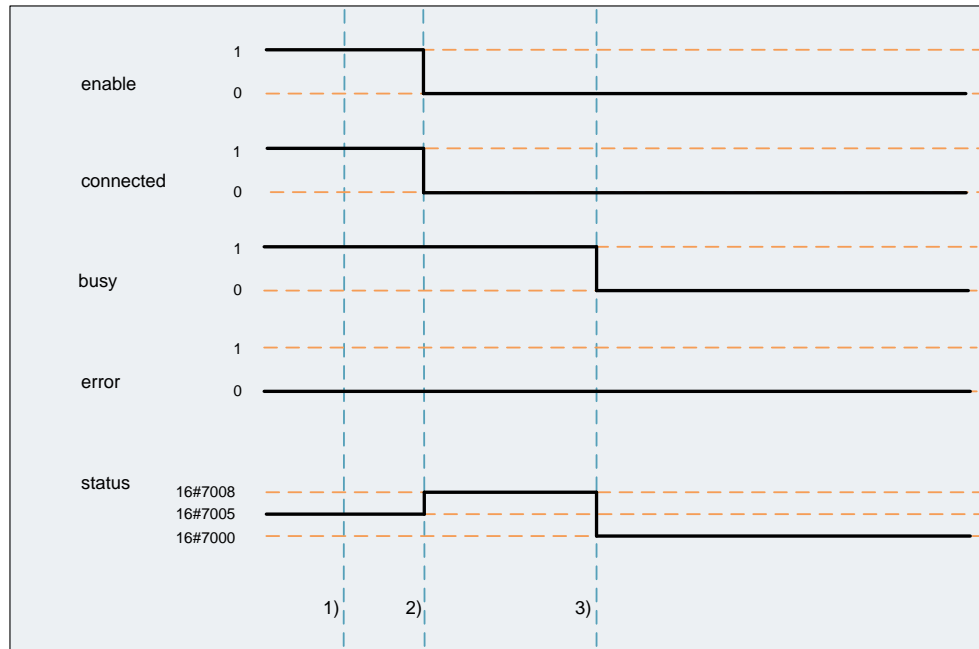
1. Der FB wurde noch nicht gestartet.
2. Nach der L/H Flanke an *enable* beginnt die Bausteinbearbeitung. Die Ausgangsvariable *busy* wird TRUE.
3. Der FB versucht als TCP-Client eine Verbindung zum Partner aufzubauen (*status* = LCOM_STATUS_CLIENT_TRY_CONNECT).
4. Die Ausgangsvariable *connected* wird TRUE, wenn die TCP-Verbindung erfolgreich zum Partner aufgebaut wurde.

2.2.2 Kommunikationsabbau

Nach einer fallenden Flanke des booleschen Eingangs *enable* wird die TCP-Verbindung abgebaut. Das Abbauen der Verbindung wird über den *status* (LCOM_STATUS_DISCONNECTING) dem Anwender mitgeteilt. Wenn die Verbindung geschlossen ist, ist die Bausteinbearbeitung beendet und die Ausgangsvariable *busy* wird FALSE.

Funktionsverlauf

Abbildung 2-5 Verbindungsabbau ohne LCom-Protokoll



1. Die TCP Verbindung ist aufgebaut.
2. Nach der H/L Flanke an *enable* baut der FB automatisch die Verbindung zum Partner ab (*status* = LCOM_STATUS_LCOM_DISCONNECTING).
3. Die Verbindung zum Partner wurde erfolgreich abgebaut. Die Ausgangsvariable *busy* wird FALSE.

2.2.3 Daten senden

Der Funktionsbaustein **FBCom_Communication** kann Daten zyklisch oder einmalig zum Partner senden. Der Sendezyklus und die Überwachungszeit (Zeit, in der ein gesendetes Paket vom Empfänger quittiert werden muss), wurden zur besseren Übersichtlichkeit in einer Struktur `sSenderType` definiert.

Tabelle 2-5 Sendeparameter

Parameter	Datentyp	Beschreibung
<code>sSenderType</code>	Struct	
<code>tCycleTime</code>	Time	Sendezyklus T#0ms .. T#24d20h30m20s630ms (default:T#1s)
<code>tAckTimeout</code>	Time	Nur mit LCom-Protokoll, (<i>configurations.sConnection.u8ComService = LCOM_LCOM_CONNECTION</i>): Die Variable <code>tAckTimeout</code> bestimmt die Überwachungszeit nach der eine Quittierung des gesendeten Pakets spätestens erwartet wird. Bei Überschreiten dieser Zeit wird die Verbindung geschlossen und neu aufgebaut. T#1ms .. T#24d20h30m20s630ms (default:T#1s)

Sendeparameter in der Applikation

```
//sender configuration
//time interval for cyclic communication
(T#1ms...T#+24d20h31m23s647ms)
gspComLComParameter.sSender.tCycleTime := T#3s;
//time interval in case of missing confirmation
(T#1ms...T#+24d20h31m23s647ms)
gspComLComParameter.sSender.tAckTimeout := T#2s;
```

Hinweis

Wenn das LCom-Protokoll verwendet wird und der Partnerseite steht nur die Protokollversion V1 zur Verfügung, dann werden die lokalen Sendeparameter mit den Empfangsparametern der Partnerseite abgeglichen (Empfängereinstellung Protokollversion V1: *sCfgReceiver.u16CycleTime* und *sCfgReceiver.u16AckTimeout*).

Der Abgleich erfolgt nach folgenden Regeln:

- Die Zykluszeit (*cycleTime*) und die Überwachungszeit (*ackTimeout*) werden jeweils auf den größeren der beiden Werte gesetzt

Es ist zu empfehlen die Empfangsparameter der Partnerseite auf 0 zu setzen.

Die Sendeparameter können auch im laufenden Betrieb geändert werden. Somit ist es möglich, die Zykluszeiten zu ändern, ohne eine erneute L/H Flanke am *enable* Eingang.

Die aktuell wirksamen Einstellungen werden in der Diagnosestruktur angezeigt.

Der Funktionsbaustein **FBCom_Communication** erwartet am Eingang *sendBuffer* ein ARRAY of BYTE. Durch Marshalling können Sie ihre Daten in den Speicherbereich vom *sendBuffer* kopieren. Dabei werden die Daten mit der SIMOTION Systemfunktion ANYTYPE_TO_BIGBYTEARRAY von der Hilfsstruktur in das Byte-Array geschrieben. Eine detaillierte Beschreibung zum Marshalling finden Sie in der Hilfe von SIMOTION SCOUT.

Das Senden der Daten vom *sendBuffer* wird mit dem booleschen Eingang *send* gestartet. Mit dem Eingangsparameter *sendDataLength* wird dem FB mitgeteilt, wie viele Daten des *sendBuffer* zum Versand bereitstehen.

Um einen inkonsistenten Zugriff auf die Sendedaten (Schreiben auf Daten, die gerade gesendet werden) vermeiden zu können, wird über die boolesche Ausgangsvariable *sending* angezeigt, dass ein Sendevorgang aktiv ist. Wenn *sending* TRUE ist, darf die Applikation die Daten im *sendBuffer* nicht ändern.

ACHTUNG

Ist die konfigurierte *sendDataLength* zu groß wird diese vom FB begrenzt. Die maximal mögliche Datenlänge wird von der Größe des eigenen *sendBuffer* begrenzt. Wird eine Begrenzung vom Baustein vorgenommen, wird am *status* folgender Wert ausgegeben 16#7600 (LCOM_WARN_SEND_DATA_LENGTH). Die Information wird auch im Diagnosepuffer eingetragen.

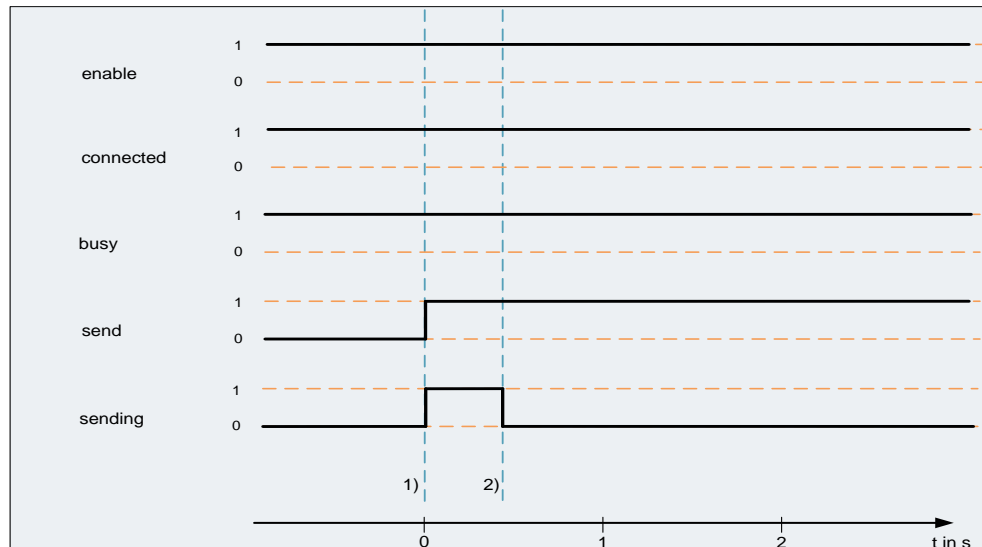
Bei Verwendung des LCom-Protokolls wird auch die Größe des Empfangspuffers auf der Partnerseite berücksichtigt. Ist der Empfangspuffer der Partnerseite (*receiveBuffer*) kleiner als der eigene *sendBuffer*, wird nur die Byteanzahl verschickt, die in den Empfangspuffer der Partnerseite passen.

Wird eine Begrenzung vom Baustein vorgenommen, wird am *status* folgender Wert ausgegeben 16#7614 (LCOM_WARN_RECEIVE_BUFFER_LEN). Die Information wird auch im Diagnosepuffer eingetragen.

Funktionsverläufe

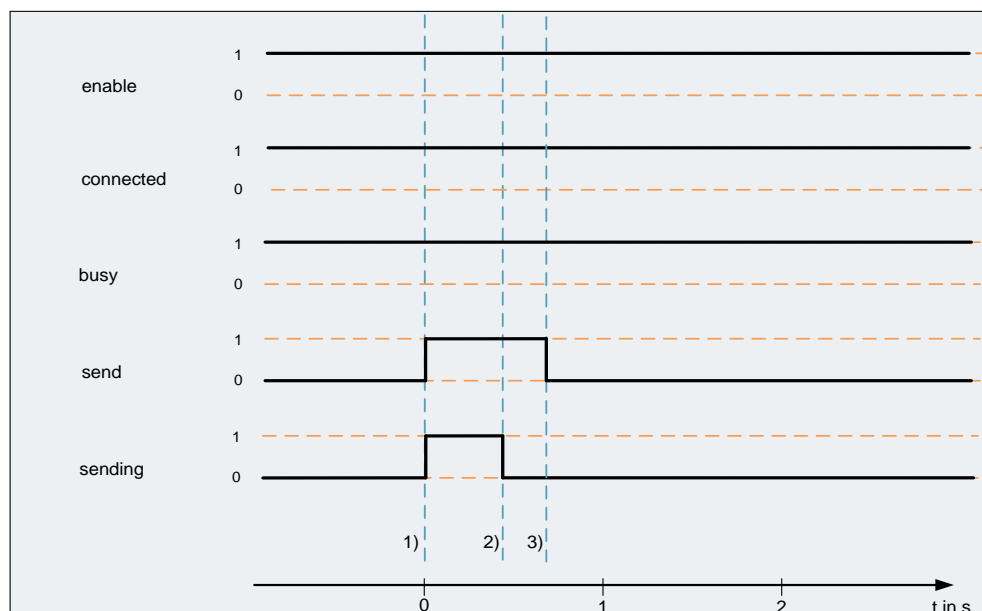
Einmalige Datenübertragung

Abbildung 2-6 Einmalige Übertragung, configurations.sSender.tCycleTime = T#0ms



1. Die Sendezykluszeit (*configurations.sSender.tCycleTime*) ist T#0ms. Der Sendevorgang wird mit *send* = TRUE gestartet. Dass der Sendevorgang aktiv ist, sieht der Anwender über die Ausgangsvariable *sending* = TRUE.
2. Der Sendevorgang ist beendet (*sending* = FALSE).

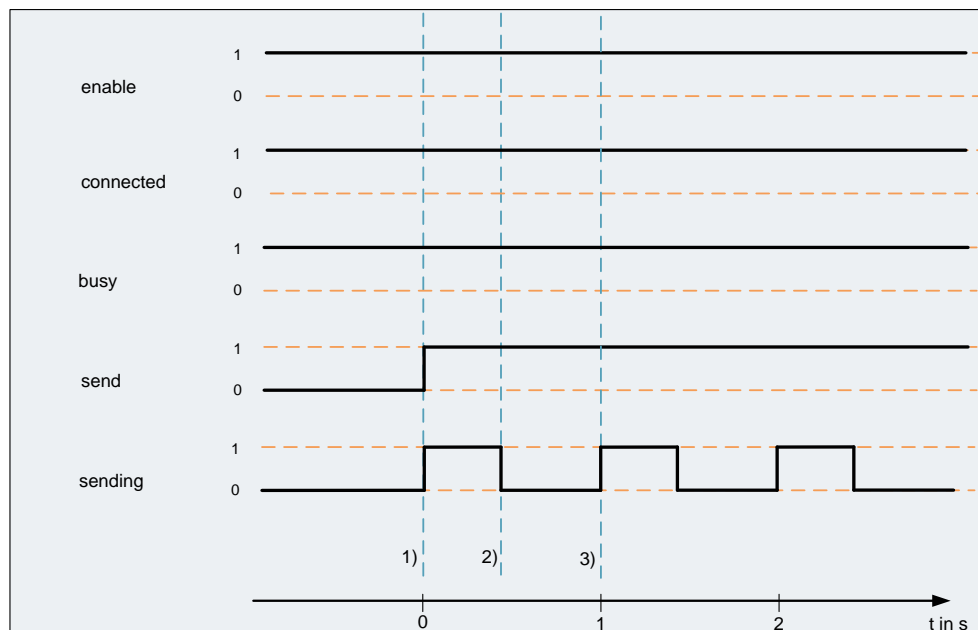
Abbildung 2-7 Einmalige Übertragung, configurations.sSender.tCycleTime = T#1s



1. Die Sendezykluszeit (*configurations.sSender.tCycleTime*) ist T#1s. Der Sendevorgang wird mit *send* = TRUE gestartet. Dass der Sendevorgang aktiv ist, sieht der Anwender über die Ausgangsvariable *sending* = TRUE.
2. Der Sendevorgang ist beendet (*sending* = FALSE).
3. *Send* wird vor dem Beginn der nächsten Sendezykluszeit wieder FALSE.

Zyklische Datenübertragung

Abbildung 2-8 Zyklische Datenübertragung, `configurations.sSender.tCycleTime = T#1s`



1. Die Sendezykluszeit (`configurations.sSender.tCycleTime`) ist $T\#1s$. Der Sendevorgang wird mit `send = TRUE` gestartet. Dass der Sendevorgang aktiv ist, sieht der Anwender über die Ausgangsvariable `sending = TRUE`.
2. Der Sendevorgang ist beendet (`sending = FALSE`).
3. Nach Ablauf der Sendezykluszeit ($T\#1s$) werden die Daten erneut versendet.

2.2.4 Daten empfangen

Mit der Eingangsvariable *readDataLength* kann man die Empfangseigenschaften und das daraus resultierende Rückmeldeverhalten der Ausgangsvariable *dataReceived* (empfangene Nutzdaten sind gültig) mehrfach beeinflussen.

Grundsätzlich muss man dabei unterscheiden, ob man nur das Standard TCP (Szenario 1) oder das LCom-Protokoll (Szenario 2) nutzt.

Für beide Szenarien gilt aber gleichermaßen:

- dass bei einer *readDataLength* gleich 0 keine Nutz- oder Konfigurationsdaten vom Partner ausgewertet werden.
- dass der Ausgangsparameter *dataReceived* für einen Zyklus gesetzt wird, um die Ankunft neuer Daten zu signalisieren.

Die Empfangsdaten sind im Datenbereich enthalten, der am *receiveBuffer* verschaltet ist (ARRAY of BYTE).

Durch Marshalling können Sie ihre Daten aus dem Speicherbereich vom *receiveBuffer* kopieren. Dabei werden die Daten mit der SIMOTION Systemfunktion *BIGBYTEARRAY_TO_ANYTYPE* von der Hilfsstruktur in Variablen vom Typ ANY geschrieben. Eine detaillierte Beschreibung zum Marshalling finden Sie in der Hilfe von SIMOTION SCOUT.

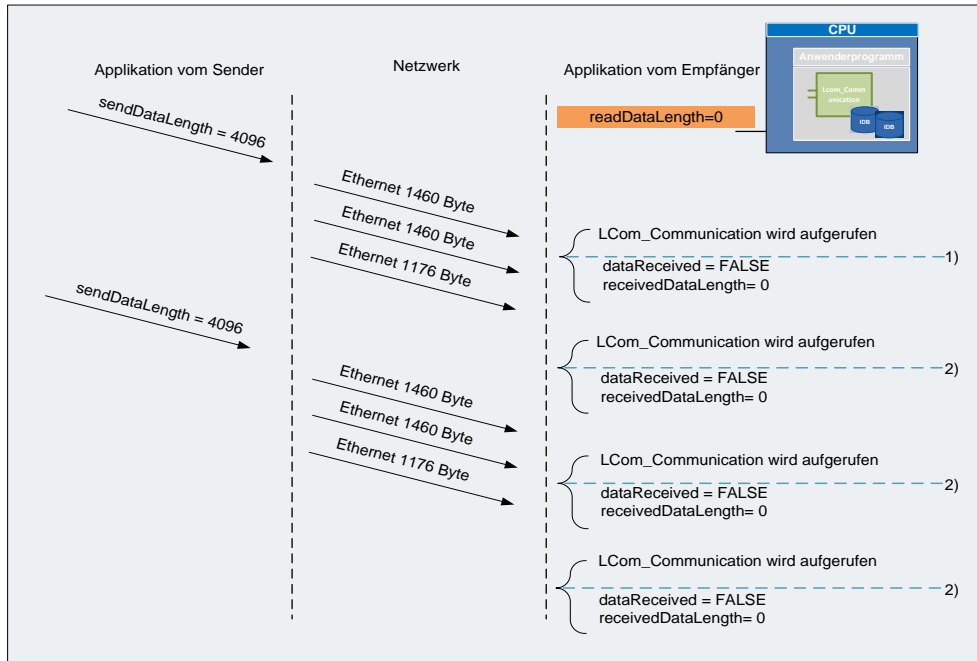
Im Folgenden sind die unterschiedlichen Empfangseigenschaften mit und ohne LCom-Protokoll bei *readDataLength* im Wertebereich zwischen 0 und 4294967295 erläutert.

Nur TCP-Protokoll,
 (configurations.sConnection.u8ComService = LCOM_TCP_CONNECTION):

readDataLength = 0: Daten aus dem TCP-Empfangspuffer werden nicht gelesen.
 Der FB wertet die empfangenen Daten vom Partner nicht aus.

Funktionsverlauf

Abbildung 2-9 Daten empfangen, nur TCP-Protokoll, *readDataLength* = 0

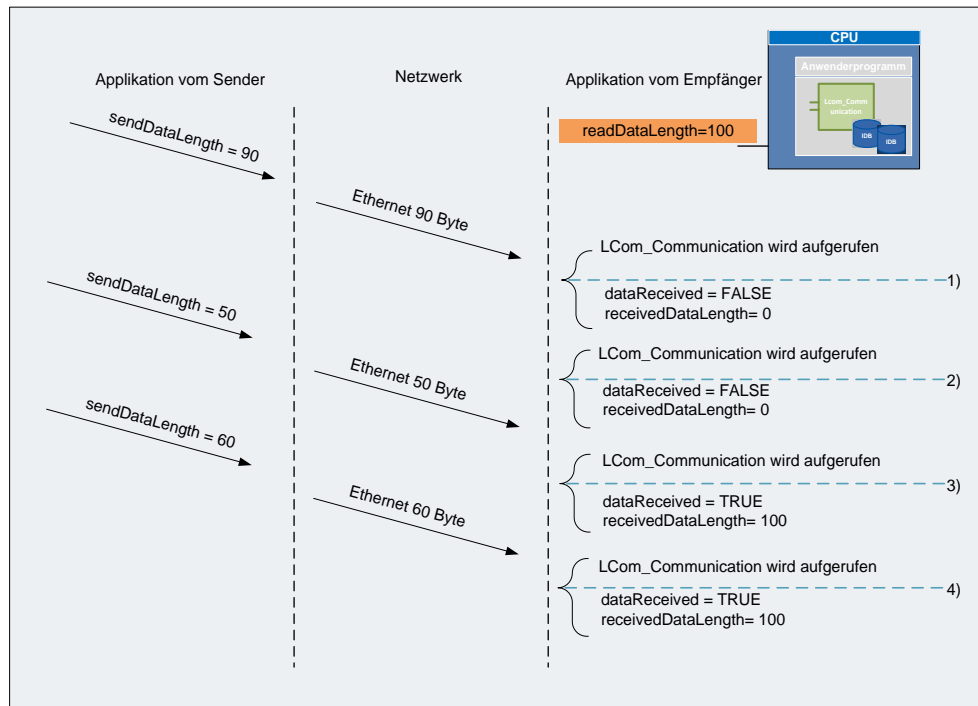


1. Es wurden bereits 1460 Bytes empfangen, da *readDataLength* aber 0 ist, werden die Daten vom FB nicht ausgewertet.
2. Weitere Aufrufe des FB bringen keine Veränderung. Die empfangenen Daten an der Schnittstelle, werden nicht ausgewertet und bleiben im TCP Puffer im System gespeichert.

readDataLength = 1 .. 4294967294: Anzahl der Bytes, die an der Schnittstelle empfangen werden müssen, bevor die Ausgangsvariable *dataReceived* gleich TRUE ist.

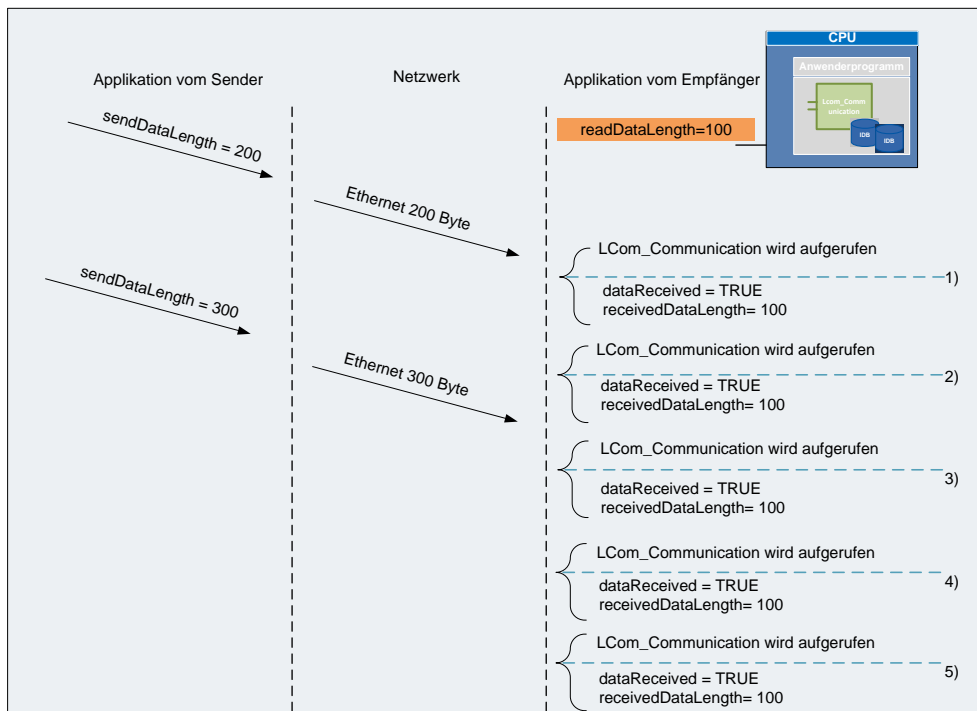
Funktionsverläufe

Abbildung 2-10 Daten empfangen, nur TCP-Protokoll, *sendDataLength* < *readDataLength*



1. Es wurden 90 Bytes empfangen. 90 Bytes sind jedoch kleiner als die geforderten 100 Bytes (*readDataLength*). Deshalb ist *dataReceived* gleich FALSE und *receivedDataLength* gleich 0.
2. An der Schnittstelle wurden keine neuen Daten empfangen (Zwischenaufruf).
3. Es wurden 50 Bytes empfangen. Damit sind insgesamt 140 Bytes empfangen. Die geforderten 100 Bytes werden an den *receiveBuffer* übertragen (*dataReceived* = TRUE, *receivedDataLength* = 100) und der Auftrag ist abgeschlossen. Die restlichen 40 Bytes bilden wieder den ersten Teil der nächsten geforderten 100 Bytes.
4. Es wurden 60 Bytes empfangen. Damit sind insgesamt 100 Bytes empfangen. Die geforderten 100 Bytes werden an den *receiveBuffer* übertragen (*dataReceived* = TRUE, *receivedDataLength* = 100) und der Auftrag ist abgeschlossen.

Abbildung 2-11 Daten empfangen, nur TCP-Protokoll, sendDataLength > readDataLength

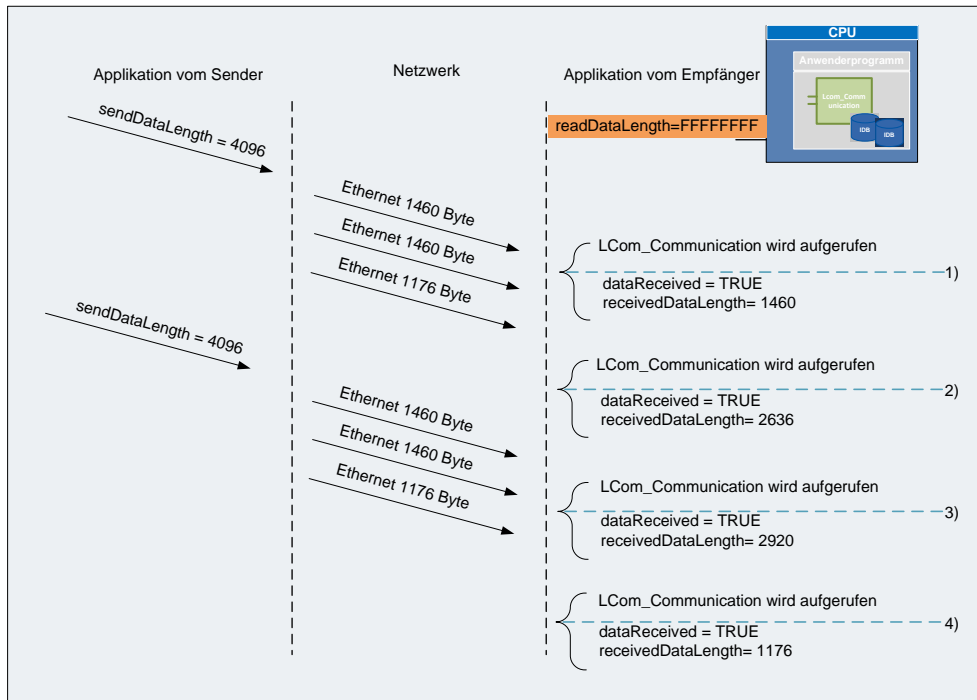


1. Es wurden 200 Bytes empfangen. Die ersten 100 Bytes werden an den *receiveBuffer* übertragen (*dataReceived = TRUE*, *receivedDataLength = 100*). Die restlichen 100 Bytes bilden wieder den ersten Teil der nächsten geforderten 100 Bytes.
2. An der Schnittstelle wurden keine neuen Daten empfangen. Es sind aber noch 100 Bytes vom vorherigen Auftrag vorhanden. Die zweiten 100 Bytes werden an den *receiveBuffer* übertragen (*dataReceived = TRUE*, *receivedDataLength = 100*) und der Auftrag ist abgeschlossen.
3. Es wurden 300 Bytes empfangen. Die ersten 100 Bytes werden an den *receiveBuffer* übertragen (*dataReceived = TRUE*, *receivedDataLength = 100*). Die restlichen 200 Bytes bilden wieder den ersten Teil der nächsten geforderten 100 Bytes.
4. An der Schnittstelle wurden keine neuen Daten empfangen. Es sind aber noch 200 Bytes vom vorherigen Auftrag vorhanden. Die zweiten 100 Bytes werden an den *receiveBuffer* übertragen (*dataReceived = TRUE*, *receivedDataLength = 100*).
5. An der Schnittstelle wurden keine neuen Daten empfangen. Es sind aber noch 100 Bytes vom vorherigen Auftrag vorhanden. Die dritten 100 Bytes werden an den *receiveBuffer* übertragen (*dataReceived = TRUE*, *receivedDataLength = 100*) und der Auftrag ist abgeschlossen.

readDataLength = 4294967295, 16#FFFFFFFF: Alle Daten, die an der Schnittstelle zur Verfügung stehen, werden sofort bei Aufruf gelesen (Ad-Hoc-Modus).

Funktionsverlauf

Abbildung 2-12 Daten empfangen, nur TCP-Protokoll, *readDataLength* = 16#FFFFFFFF



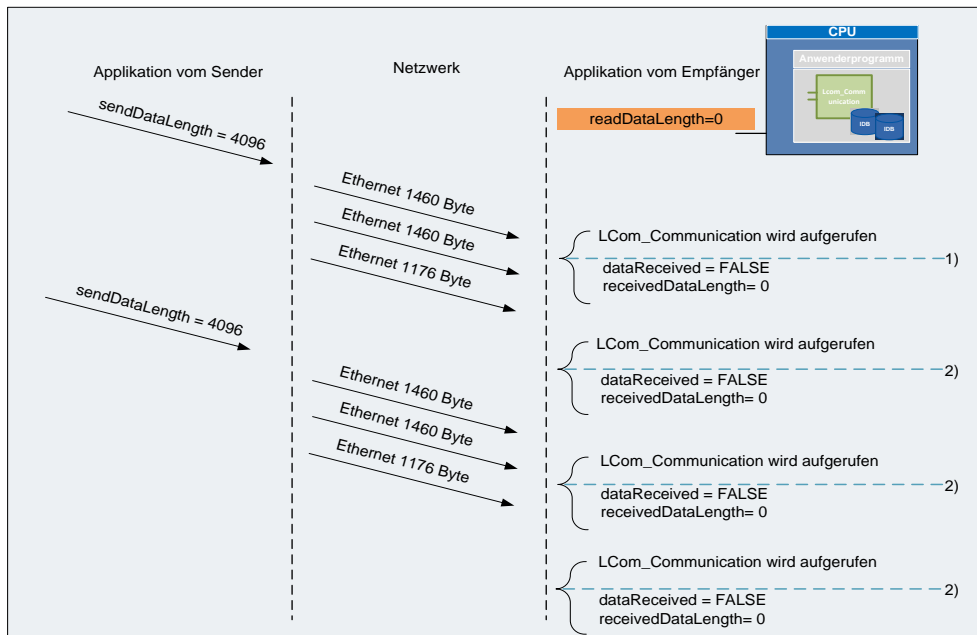
1. Es wurden 1460 Bytes empfangen. Diese werden sofort an den *receiveBuffer* übertragen. (*dataReceived* = TRUE, *receivedDataLength* = 1460).
2. Es wurden 2636 Bytes empfangen. Diese werden sofort an den *receiveBuffer* übertragen. (*dataReceived* = TRUE, *receivedDataLength* = 2636).
3. Es wurden 2920 Bytes empfangen. Diese werden sofort an den *receiveBuffer* übertragen. (*dataReceived* = TRUE, *receivedDataLength* = 2920).
4. Es wurden 1176 Bytes empfangen. Diese werden sofort an den *receiveBuffer* übertragen. (*dataReceived* = TRUE, *receivedDataLength* = 1176).

Mit LCom-Protokoll,
 (configurations.sConnection.u8ComService = LCOM_LCOM_CONNECTION):

readDataLength = 0: Daten aus dem TCP-Empfangspuffer werden nicht gelesen.
 Der FB wertet die empfangenen Daten vom Partner nicht aus.

Funktionsverlauf

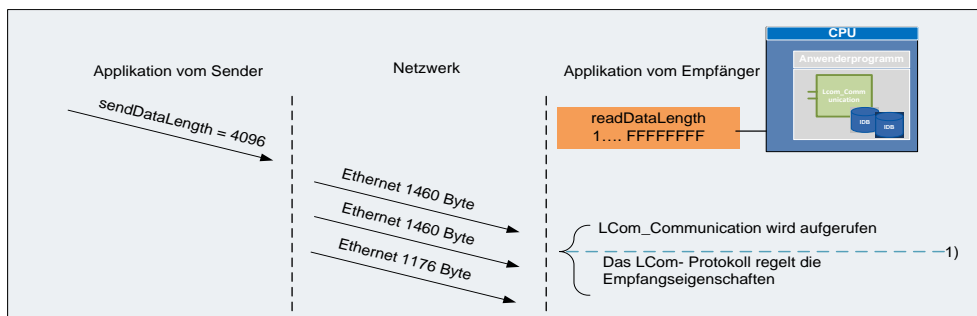
Abbildung 2-13 Daten empfangen, LCom-Protokoll, *readDataLength* = 0



1. Es wurden bereits 1460 Bytes empfangen, da *readDataLength* aber 0 ist, werden die Daten vom FB nicht ausgewertet.
2. Weitere Aufrufe des FB bringen keine Veränderung. Die empfangenen Daten an der Schnittstelle werden nicht ausgewertet und bleiben im TCP Puffer im System gespeichert.

readDataLength = 1 .. 4294967295: Der angegebene Wert hat keine Relevanz für das Empfangsverhalten.

Abbildung 2-14 Daten empfangen, LCom-Protokoll, *readDataLength* <> 0



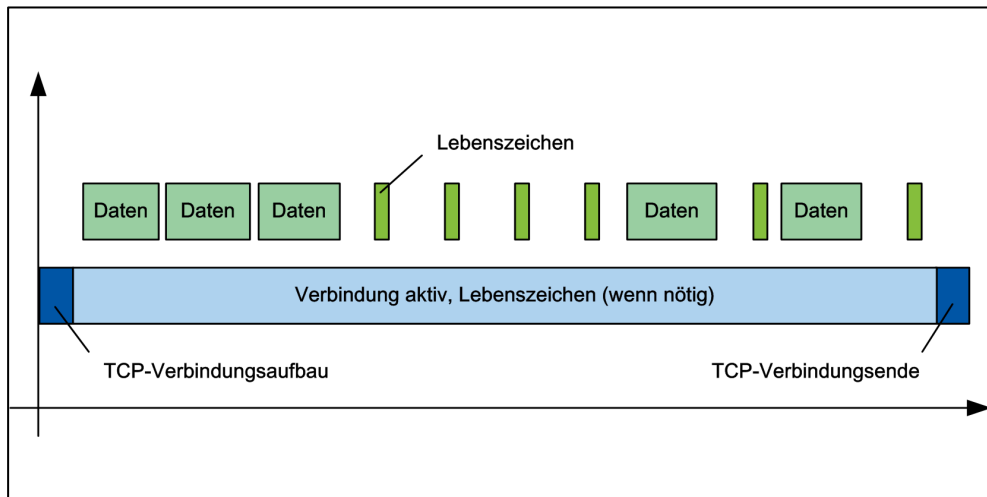
1. Das LCom-Protokoll regelt die die Empfangseigenschaften.

2.2.5 Lebenszeichen

Bei Standard TCP-Kommunikation kann das Erkennen eines Ausfalls des Kommunikationspartners mehrere Sekunden dauern. Durch die Überwachung des Lebenszeichens wird diese Zeitdauer deutlich verkürzt. Das Lebenszeichen wird nur bei Verwendung des LCom-Protokolls gesendet.

Werden Daten übertragen, kann von einer fehlerfreien Verbindung ausgegangen werden. Bei einer Sendepause steht diese Information nicht zur Verfügung. In diesen Phasen ohne Nutzdatenaustausch wird automatisch ein Lebenszeichen generiert. Die Zykluszeit des periodischen Lebenszeichens kann konfiguriert werden (*configurations.sSender.tCycleTime*). Werden nach Ablauf des vierfachen Lebenszeichenzyklus kein neues Lebenszeichen bzw. keine neuen Nutzdaten empfangen, wird von einem Abbruch der Verbindung ausgegangen. Die Verbindung wird automatisch geschlossen und neu aufgebaut.

Abbildung 2-15 Lebenszeichen



Hinweis

Das Vielfache des Lebenszeichenzyklus, nach welchem ein Verbindungsabbruch gemeldet wird, ist mit dem Wert 4 festgelegt.

2.2.6 Uhrzeitsynchronisation

Die Uhrzeitsynchronisation kann nur bei der Kommunikation mit dem LCom-Protokoll verwendet werden. Zur Uhrzeitsynchronisation zweier Steuerungen wird die aktuelle Uhrzeit einer Steuerung an den Partner geschickt und dort als Systemzeit übernommen. Die Parameter für die Uhrzeitsynchronisation werden bei steigender Flanke am *enable* Eingang übernommen. Ist die Uhrzeitsynchronisation aktiv (*configurations.sTimeSync.u8sendMode>0*), wird eine Uhrzeitsynchronisation durchgeführt nachdem die Verbindung aufgebaut ist. Danach erfolgt die Synchronisation abhängig vom eingestellten Sendemodus.

Bei der Uhrzeitsynchronisation wird unterschieden in:

- Zyklische Uhrzeitsynchronisation Vorgabe einer Zykluszeit für erneute Synchronisation
- Tägliche Uhrzeitsynchronisation mittels Vorgabe einer Uhrzeit, zu der die Synchronisation täglich durchgeführt wird

Für die bessere Übersichtlichkeit der Parametrierung wurde dafür eine Struktur *sTimeSyncType* definiert.

Tabelle 2-6 Parameter für die Uhrzeitsynchronisation

Parameter	Datentyp	Beschreibung
<i>sTimeSyncType</i>	Struct	
<i>boUsePartnerTimestamps</i>	Bool	TRUE: Die empfangene Zeit wird als Systemzeit übernommen. FALSE (default): Empfangene Uhrzeittelegramme werden ignoriert.
<i>u8SendMode</i>	USInt	Sendemodus Zeitsynchronisation 0, LCOM_NO_TIME_SYNC (default): keine Zeitsynchronisation 1, LCOM_CYCLE_TIME_SYNC: zyklische Zeitsynchronisation. 2, LCOM_TIME_OF_DAY_TIME_SYNC: Uhrzeit, wann die Zeitsynchronisation erfolgen soll.
<i>tCycleTime</i>	Time	Nur bei sendMode 1, LCOM_CYCLE_TIME_SYNC: Sendezykluszeit der Zeitstempel (default: T#1h). >T#0ms
<i>todSendAtTimeOfDay</i>	Time_Of_Day	Nur bei sendMode 2, LCOM_TIME_OF_DAY_TIME_SYNC: Uhrzeit, bei der ein Zeitsynchronisationstelegramm verschickt wird. (default:TOD#05:00:0.0).

Parameter für die Uhrzeitsynchronisation in der Applikation

```
//time synchronization configuration
//synchronize local time with received time stamps
gspComLComParameter.sTimeSync.boUsePartnerTimestamps := FALSE;
//0: inactive, 1: cyclic, 2: at time of day
gspComLComParameter.sTimeSync.u8SendMode := LCOM_NO_TIME_SYNC;
//time interval for cyclic mode (T#1ms...T#+24d20h31m23s647ms)
gspComLComParameter.sTimeSync.tCycleTime := T#60m;
//time synchronization at specified time of day
gspComLComParameter.sTimeSync.todSendAtTimeOfDay := TOD#00:00:0.000;
```

Hinweis

Es darf nur einer der beiden Verbindungspartner die Zeitsynchronisation durchführen. Ist bei beiden Partnern die Variable *boUsePartnerTimestamps* gesetzt, oder beide wollen den Partner synchronisieren, dann wird die Synchronisation nicht ausgeführt und eine Warnung ausgegeben (LCOM_WARN_NO_TIME_SYNC).

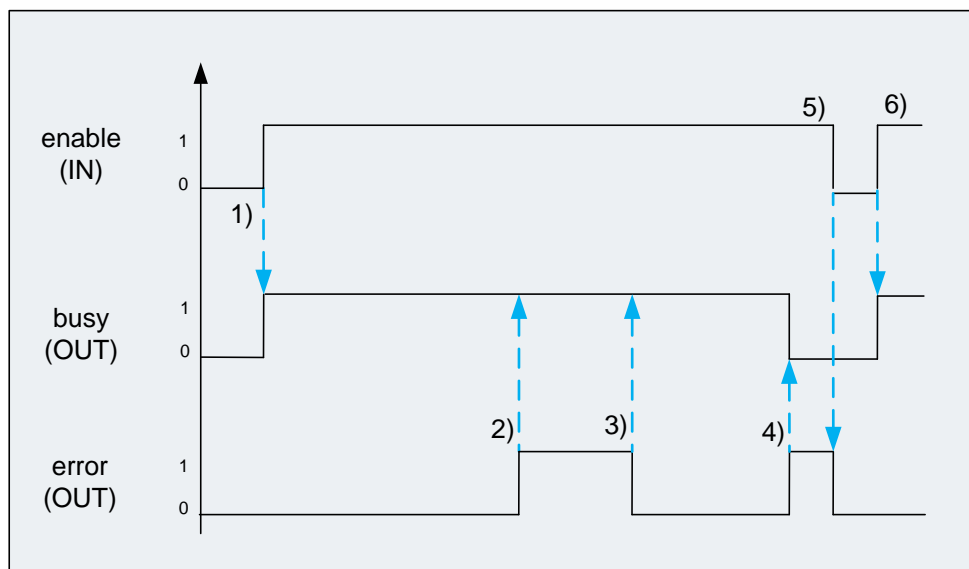
2.3 Fehlerrückgabe und Diagnose

Status- und Fehlermeldungen werden über die Ausgangsvariable *status* angezeigt. Da der Status sich häufig ändern kann und dem Anwender auch eine Baueinheitshistorie zu Verfügung stehen soll, gibt es eine Diagnosestruktur, in die auch Status- bzw. Fehlermeldungen eingetragen werden.

Ein Fehler wird durch Setzen der booleschen Variable *error* angezeigt. Dabei muss man unterscheiden, ob die Fehlerursache durch den Benutzer oder vom Baueinheit behoben werden kann. Handelt es sich um einen Fehler, der vom Baueinheit behoben werden kann, bleibt *busy* gesetzt. Tritt ein Fehler auf, der durch den Benutzer behoben werden muss, wird *busy* zurückgesetzt (Baueinheitbearbeitung deaktiviert).

In der nachfolgenden Abbildung sind die zwei Szenarien als Signalablaufdiagramm dargestellt.

Abbildung 2-16 Signalablaufdiagramm error/busy



1. *Busy* wird mit einer steigenden Flanke von *enable* gesetzt.
2. Ein Fehler tritt auf und *error* wird gesetzt. Da es sich um einen Fehler handelt, der vom Baueinheit selbst behoben werden kann, bleibt *busy* gesetzt.
3. Nach Behebung der Fehlerursache (z.B. Neuaufbau der Verbindung) wird *error* wieder zurückgesetzt.
4. Ein Fehler, der nur durch den Benutzer behoben werden kann, tritt ein. Hierbei wird *error* gesetzt und *busy* zurückgesetzt.
5. Nur durch eine fallende Flanke an *enable* kann der anstehende Fehler, der durch den Benutzer behoben werden muss, quittiert werden.
6. Der Baueinheit wird wieder mit einer steigenden Flanke von *enable* gestartet.

Im Ausgangsparameter *diagnostics* befinden sich verschiedene Teilstrukturen, die durch die Struktur *sLCOM_DiagnosticsType* definiert sind. Im Folgenden soll auf den Diagnosepuffer eingegangen werden. Die vollständige Beschreibung der Struktur *sLCOM_DiagnosticsType* entnehmen Sie bitte aus Kapitel 2.4 Übersicht der Strukturen und Konstanten. Im Diagnosepuffer werden verschiedene Status- bzw. Fehlermeldungen ein-getragen. Das Array ist auf 64 Einträge festgelegt. Dieses Array arbeitet als Ringpuffer. Die Variable *u8BufferIndex* zeigt auf den letzten (aktuellen) Eintrag.

Jeder Eintrag besteht aus folgenden Elementen: Status- bzw. Fehlernummer, Datum und Uhrzeit des Auftretens sowie dem aktuellen Zustand des Fehlers. Der Zustand beim Eintrag in den Puffer ist immer aktiv. Beim Zurücksetzen des Fehlers wird der Status im Puffer auf inaktiv gesetzt. In der Struktur stehen sowohl der Rückgabewert der Systemfunktion als auch vier Zusatzwerte zur Verfügung, die abhängig vom aufgetretenen Fehler, detaillierte Informationen enthalten können.

Tabelle 2-7 Struktur des Diagnosepuffers

Parameter	Datentyp	Beschreibung
u8BufferIndex	USint	Index auf letzten Eintrag
asBuffer	Array [0..63] of Struct	Diagnosepuffer
b16Status	Word	Status vom FB
dtTimestamp	DT	Zeitstempel bei Auftritt der Meldung
bolsActive	Bool	TRUE: Zustand ist noch vorhanden. FALSE: Zustand ist nicht mehr vorhanden.
i32SubFunctionErrorID	DInt	ggf. Rückgabewert einer Systemfunktion
r32AdditionalValue1	Real	Zusatzwert 1
r32AdditionalValue2	Real	Zusatzwert 2
r32AdditionalValue3	Real	Zusatzwert 3
tAdditionalValue4	Time	Zusatzwert 4

Abbildung 2-20 Diagnosepuffer in der Applikation

	Name	Data type
	All	All
18	diagnostics	'sLCOM_DiagnosticsType'
19	sLocalConfig	'sLocalConfigType'
20	sPartnerConfig	'sPartnerConfigType'
21	sStatistics	'sStatisticsType'
22	u8BufferIndex	USINT
23	asBuffer	'asBufferType'
24	asBuffer[0]	'sBufferType'
25	b16Status	WORD
26	dtTimestamp	DT
27	bolsActive	BOOL
28	i32SubFunctionErrorID	DINT
29	r32AdditionalValue1	REAL
30	r32AdditionalValue2	REAL
31	r32AdditionalValue3	REAL
32	tAdditionalValue4	TIME

2.3.1 Statusmeldungen

Für die Statusmeldungen wurden folgende Nummernbänder festgelegt.

Tabelle 2-8 Status Nummernband

Fehlergrund	Nummernband status
Kein Auftrag in Bearbeitung (auch Initialwert)	16#7000
Erster Aufruf nach Eingang eines neuen Auftrags	16#7001
Folgeaufruf während aktiver Bearbeitung ohne weitere Detaillierung	16#7002
Folgeaufruf während aktiver Bearbeitung mit weiterer Detaillierung.	16#7003 .. 16#7FFF
Fehler bei der Parametrierung	16#8200 .. 16#83FF
Fehler bei der Abarbeitung intern (z.B. bei Aufruf einer Systemfunktion)	16#8600 .. 16#87FF

Status – und Warnmeldungen liegen in dem Nummernband 16#7000 bis 16#7FFF. Kommt es zu Fehlern bei der Parametrierung (16#8200 .. 16#83FF), ist ein Eingriff durch den Anwender erforderlich. Der Anwender muss den fehlerhaften Wert, durch einen zulässigen Wert ersetzen und den Baustein mit einer steigenden Flanke am *enable* Eingang neu starten.

Bei einem internen Fehler (16#8600 .. 16#87FF) wird der Funktionsbaustein automatisch die Verbindung schließen und versuchen diese neu aufzubauen. Eine erneute steigende Flanke am *enable* Eingang ist nicht notwendig.

Um die Anwenderkonstanten einfach zuordnen zu können, enthält jede Konstante einen der folgenden Namenspräfixe:

- **STATUS** → 16#7000 .. 16#75FF → LCOM_STATUS_FIRST_CALL
- **WARN** → 16#7600 .. 16#7FFF → LCOM_WARN_SEND_DATA_LENGTH
- **ERR** → 16#8200 .. 16#83FF → LCOM_ERR_INTERFACE_ID
16#8600 .. 16#87FF → LCOM_ERR_SEND

Die verwendeten Statusmeldungen sind nachstehend aufgeführt. Die Meldungen werden an der Ausgangsvariable *status* angezeigt und im Diagnosepuffer eingetragen.

Tabelle 2-9 Statusmeldungen

<p>16#7000 LCOM_STATUS_NO_CALL</p>	<p>Kein Auftrag in Bearbeitung. Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4</i> T#0ms</p>
<p>16#7001 LCOM_STATUS_FIRST_CALL</p>	<p>Erster Aufruf nach Eingang eines neuen Auftrags (steigende Flanke an <i>enable</i>). Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4</i> T#0ms</p>
<p>16#7002 LCOM_STATUS_INTERMEDIATE_CALL</p>	<p>Folgeaufruf während der aktiven FB Bearbeitung. Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4</i> T#0ms</p>
<p>16#7003 LCOM_STATUS_CLIENT_TRY_CONNECT</p>	<p>FB versucht sich als TCP-Client zu verbinden. Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4</i> T#0ms</p>
<p>16#7004 LCOM_STATUS_SERVER_PORT_OPEN</p>	<p>FB ist als TCP-Server konfiguriert und wartet auf eine TCP- Client Anfrage. Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4</i> T#0ms</p>
<p>16#7005 LCOM_STATUS_TCP_CONNECTED</p>	<p>Die TCP-Verbindung zum Partner ist aufgebaut. Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4</i> T#0ms</p>

16#7006 LCOM_STATUS_LCOM_CONNECTED_V1	Der FB hat erfolgreich die Konfigurationsdaten über das LCom Protokoll V1 mit dem Partner ausgetauscht. Die maximale mögliche Datenlänge beträgt 64 KByte. Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms
16#7007 LCOM_STATUS_LCOM_CONNECTED_V2	Der FB hat erfolgreich die Verbindung aufgebaut und die Konfigurationsdaten (LCom Protokoll V2) mit dem Partner ausgetauscht. Die maximale mögliche Datenlänge beträgt 16 MByte. Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms
16#7008 LCOM_STATUS_DISCONNECTING	Der FB baut die Verbindung ab. Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms
16#7600 LCOM_WARN_SEND_DATA_LENGTH	Die angegebene Sendelänge (<i>sendDataLength</i>) ist zu groß und wird vom FB automatisch begrenzt. Die maximal mögliche Sendelänge wird durch die Größe des eigenen Sendepuffers (<i>sendBuffer</i>) bestimmt. Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> <i>sendDataLength</i> <i>r32AdditionalValue2:</i> <i>sendBuffer</i> <i>r32AdditionalValue3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms
16#7610 LCOM_WARN_LIFE_SIGN_LIMITATION	Die angegebene Zeit (<i>configurations.sConnection.tLifeSignCycleTime</i>) für die Verbindungsüberwachung ist zu groß und wird automatisch auf 65535 ms begrenzt. Diagnoseeintrag: <i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> Begrenzung auf 65535 <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> Wert des Anwenders

<p>16#7611 LCOM_WARN_SEND_CYCLE_LIMITATION</p>	<p>Die angegebene Zeit für den Sendezyklus (<i>configurations.sSender.tCycleTime</i>) ist zu groß und wird automatisch auf 65535 ms begrenzt.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> Begrenzung auf 65535 <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> Wert des Anwenders</p>
<p>16#7612 LCOM_WARN_ACK_TIMEOUT_LIMITATION</p>	<p>Die konfigurierte Überwachungszeit (<i>configurations.sSender.tAckTimeout</i>) ist zu groß und wird automatisch auf 65535 ms begrenzt.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> Begrenzung auf 65535 <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> Wert des Anwenders</p>
<p>16#7613 LCOM_WARN_SEND_CYCLE_EXPIRED</p>	<p>Der konfigurierte Sendezyklus (<i>configurations.sSender.tCycleTime</i>) kann nicht eingehalten werden, weil der vorherige Sendevorgang noch nicht abgeschlossen ist.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> Wert des Anwenders <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> Bisherige Zeit des vorherigen Sendeauftrags</p>
<p>16#7614 LCOM_WARN_RECEIVE_BUFFER_LEN</p>	<p>Mit LCom Protokoll: (<i>configurations.sConnection.u8ComService = LCOM_LCOM_CONNECTION</i>):</p> <p>Die angegebene Sendelänge (<i>sendDataLength</i>) ist größer als der Empfangspuffer des Partners.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> <i>sendDataLength</i> <i>r32AdditionalValue2:</i> Empfangspuffer der Partnerseite <i>r32AdditionalValue3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p> <p>Ohne LCom Protokoll: (<i>configurations.sConnection.u8ComService = LCOM_TCP_CONNECTION</i>):</p> <p>Die angegebene <i>readDataLength</i> am FB ist größer als</p>

	<p>der eigene Empfangspuffer (receiveBuffer). Die <i>readDataLength</i> wird auf die Größe des eigenen Empfangspuffers begrenzt.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> Eigener Empfangspuffer <i>r32AdditionalValue2:</i> <i>readDataLength</i> <i>r32AdditionalValue3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#7615 LCOM_WARN_NO_TIME_SYNC</p>	<p>Es wird keine Uhrzeitsynchronisation durchgeführt → siehe Kapitel 2.2.6 Uhrzeitsynchronisation.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..4:</i> 0.0</p>
<p>16#8200 LCOM_ERR_COM_SERVICE</p>	<p>Der angegebene Kommunikationsdienst (<i>configurations.sConnection.u8ComService</i>) ist unzulässig → siehe Systemfunktion <i>tcpOpenClient/tcpOpenServer</i>.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1</i> Wert des Anwenders <i>r32AdditionalValue2...3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8201 LCOM_ERR_CONNECTION_ID</p>	<p>Die angegebene Verbindungsreferenz ist unzulässig</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1</i> Wert des Anwenders <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8202 LCOM_ERR_INTERFACE_ID</p>	<p>Die angegebene Hardwareerkennung für die lokale Schnittstelle ist unzulässig</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1</i> Wert des Anwenders <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>

<p>16#8203 LCOM_ERR_LOCAL_PORT</p>	<p>Der angegebene lokale Port (<i>configurations.sConnection.au8LocalPort</i>) ist unzulässig → siehe Systemfunktion <i>tcpOpenClient/tcpOpenServer</i>.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> Status von <i>tcpOpenClient/tcpOpenServer</i> <i>r32AdditionalValue1</i> Wert des Anwenders <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8204 LCOM_ERR_PARTNER_IP</p>	<p>Die angegebene Partner IP Adresse (<i>configurations.sConnection.u16PartnerIP</i>) ist unzulässig → siehe Systemfunktion <i>tcpOpenClient</i>.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> Status von <i>tcpOpenClient</i> <i>r32AdditionalValue1</i> Wert des Anwenders <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8205 LCOM_ERR_TIME_SYNC_MODE</p>	<p>Der angegebene Modus für die Uhrzeitsynchronisation (<i>configurations.sTimeSync.u8SendMode</i>) ist unzulässig → siehe Kapitel 2.2.6 Uhrzeitsynchronisation.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1</i> Wert des Anwenders <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8206 LCOM_ERR_TIME_CYCLE_TIME</p>	<p>Die angegebene Zykluszeit für die Uhrzeitsynchronisation (<i>configurations.sTimeSync.tCycleTime</i>) ist unzulässig → siehe Kapitel 2.2.6 Uhrzeitsynchronisation.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3</i> 0.0 <i>tAdditionalValue4:</i> Wert des Anwenders</p>
<p>16#8207 LCOM_ERR_TIME_OF_DAY</p>	<p>Der angegebene Zeitpunkt für die Uhrzeitsynchronisation (<i>configurations.sTimeSync.todSendAtTimeOfDay</i>) ist unzulässig → siehe Kapitel 2.2.6 Uhrzeitsynchronisation.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>

<p>16#8208 LCOM_ERR_SEND_DATA_TYPE</p>	<p>Der übergebene Wert an der Eingangsvariable <i>sendBuffer</i> ist kein Typ ARRAY of BYTE und unzulässig.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8209 LCOM_ERR_RECEIVE_DATA_TYPE</p>	<p>Der übergebene Wert an der Ausgangsvariable <i>receiveBuffer</i> ist kein Typ ARRAY of BYTE und unzulässig.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8210 LCOM_ERR_LIFE_SIGN_CYCLE</p>	<p>Die angegebene Zeit (<i>configurations.sConnection.tLifeSignCycleTime</i>) für die Verbindungsüberwachung ist unzulässig → siehe Kapitel 2.2.1 Kommunikationsaufbau.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3</i> 0.0 <i>tAdditionalValue4:</i> Wert des Anwenders</p>
<p>16#8211 LCOM_ERR_SEND_CYCLE_TIME</p>	<p>Der angegebene Sendezyklus (<i>configurations.sSender.tCycleTime</i>) ist unzulässig → siehe Kapitel 2.2.3 Daten senden.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3</i> 0.0 <i>tAdditionalValue4:</i> Wert des Anwenders</p>
<p>16#8212 LCOM_ERR_ACK_TIMEOUT</p>	<p>Die konfigurierte Überwachungszeit (<i>configurations.sSender.tAckTimeout</i>) ist unzulässig → siehe Kapitel 2.2.3 Daten senden.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1..3</i> 0.0 <i>tAdditionalValue4:</i> Wert des Anwenders</p>

16#8600 LCOM_ERR_OPEN_CLIENT	<p>Es ist ein Fehler bei der Verwendung der Systemfunktion tcpOpenClient aufgetreten, siehe <i>i32SubFunctionErrorID</i> im Diagnosebuffer und Systemdokumentation. Die Verbindung wird neu aufgebaut.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> Status von tcpOpenClient <i>r32AdditionalValue1:</i> 0.0 <i>r32AdditionalValue2:</i> 0.0 <i>r32AdditionalValue3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
16#8601 LCOM_ERR_OPEN_SERVER	<p>Es ist ein Fehler bei der Verwendung der Systemfunktion tcpOpenServer aufgetreten, siehe <i>i32SubFunctionErrorID</i> im Diagnosebuffer und Systemdokumentation. Die Verbindung wird neu aufgebaut.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> Status von tcpOpenServer <i>r32AdditionalValue1:</i> Lokaler Port <i>r32AdditionalValue2:</i> 0.0 <i>r32AdditionalValue3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
16#8602 LCOM_ERR_SEND	<p>Es ist ein Fehler bei der Verwendung der Systemfunktion tcpSend aufgetreten, siehe <i>i32SubFunctionErrorID</i> im Diagnosebuffer und Systemdokumentation. Die Verbindung wird neu aufgebaut.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> Status von tcpSend <i>r32AdditionalValue1:</i> 0.0 <i>r32AdditionalValue2:</i> Länge der Sendedaten <i>r32AdditionalValue3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
16#8603 LCOM_ERR_RECEIVE	<p>Es ist ein Fehler bei der Verwendung der Systemfunktion tcpReceive aufgetreten, siehe <i>subFunctionErrorID</i> im Diagnosebuffer und Systemdokumentation. Die Verbindung wird neu aufgebaut.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> Status von tcpReceive <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>

<p>16#8604 LCOM_ERR_CLOSE_CONNECTION</p>	<p>Es ist ein Fehler bei der Verwendung der Systemfunktion tcpCloseConnection aufgetreten, siehe <i>subFunctionErrorID</i> im Diagnosepuffer und Systemdokumentation. Die Verbindung wird neu aufgebaut.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> Status von tcpCloseConnection <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8605 LCOM_ERR_CLOSE_SERVER</p>	<p>Es ist ein Fehler bei der Verwendung der Systemfunktion tcpCloseServer aufgetreten, siehe <i>subFunctionErrorID</i> im Diagnosepuffer und Systemdokumentation.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> Status von tcpCloseServer <i>r32AdditionalValue1..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8610 LCOM__ERR_LIFE_SIGN_EXPIRED</p>	<p>Das Lebenszeichen des Partners wurde zu lange nicht empfangen. Die Verbindung wird neu aufgebaut.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> Lebenszeichenzyklus <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> Zeit in der der FB keine Daten vom Partner empfangen hat.</p>
<p>16#8611 LCOM_ERR_ACK_TIMEOUT_EXPIRED</p>	<p>Die Überwachungszeit eines gesendeten Datenpaketes ist abgelaufen, ohne dass eine Bestätigung vom Partner empfangen wurde. Die Verbindung wird neu aufgebaut.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> Überwachungszeit <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> Zeit in der der FB keine Daten vom Partner empfangen hat.</p>

<p>16#8612 LCOM_ERR_ACK_MESSAGE_NUMBER</p>	<p>Es wurde eine unzulässige Acknowledge Nummer vom Partner empfangen, die Verbindung wird neu aufgebaut.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> Empfangene acknowledge message Nummer <i>r32AdditionalValue2:</i> Send acknowledge message Nummer <i>r32AdditionalValue3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8613 LCOM_ERR_WRONG_LCOM_HEADER</p>	<p>Die LCom (Protokoll-) Versionen der Bibliothek LCom passt nicht zum Partner oder es wurde eine unzulässige (Protokoll-) Versionen empfangen. Die Verbindung wird neu aufgebaut.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> empfangene Version <i>r32AdditionalValue2:</i> lokale, aktive Version <i>r32AdditionalValue3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>
<p>16#8614 LCOM_ERR_WRONG_TELEGRAM_ID</p>	<p>Es wurde ein Telegramm empfangen, welches eine nicht erwartete ID hat. Die Verbindung wird neu aufgebaut.</p> <p>Diagnoseeintrag:</p> <p><i>dtTimestamp:</i> Zeitpunkt des Ereignisses <i>bolsActive:</i> Zustand <i>i32SubFunctionErrorID:</i> 0 <i>r32AdditionalValue1:</i> empfangene ID <i>r32AdditionalValue2..3:</i> 0.0 <i>tAdditionalValue4:</i> T#0ms</p>

2.4 Übersicht der Strukturen und Konstanten

Die nachfolgende Tabelle zeigt die Struktur von sLCOM_ConfigType.

Tabelle 2-10 Struktur sLCOM_ConfigType

Name	Typ	Startwert	Kommentar
sConnection	Struct		
u8ComService	USInt	2	<p>Konfiguration des Kommunikationsprotokolls.</p> <p>1, LCOM_TCP_CONNECTION: Das Standard TCP Protokoll wird verwendet.</p> <p>2, LCOM_LCOM_CONNECTION(default): Das LCom Protokoll wird verwendet. Mit dem LCom Protokoll werden zusätzliche Kommunikationsfunktionalitäten ermöglicht</p>
bolsClient	Bool	FALSE	<p>TRUE: Aktiver Verbindungsaufbau als TCP-Client.</p> <p>FALSE: Passiver Verbindungsaufbau als TCP-Server.</p>
u16LocalPort	UInt	3456	Lokaler Port, siehe Systemfunktion tcpOpenClient/Server.
u16PartnerPort	UInt	3456	<p>Nur bei aktivem Verbindungsaufbau (<i>configurations.sConnection.bolsClient = TRUE</i>):</p> <p>Port der Partnerseite, siehe Systemfunktion tcpOpenClient.</p>
au8PartnerIP	Array of USInt	-	IP Adresse vom Partner.
boAcceptUnknownPartner	Bool	TRUE	<p>Nur bei passivem Verbindungsaufbau (<i>configurations.sConnection.bolsClient = FALSE</i>):</p> <p>TRUE: Vorher nicht konfigurierter Verbindungspartner wird akzeptiert.</p> <p>FALSE: Es wird nur der konfigurierte Verbindungspartner akzeptiert.</p>
tLifeSignCycleTime	Time	T#1s	<p>Lebenszeichenzyklus</p> <p>T#1ms .. T#24d20h30m20s630ms</p>

Name	Typ	Startwert	Kommentar
sSender	Struct		
tCycleTime	Time	T#1s	Sendezyklus T#0ms .. T#24d20h30m20s630ms
tAckTimeout	Time	T#1s	Nur mit LCom-Protokoll, (<i>configurations.sConnection.u8ComService = LCOM_LCOM_CONNECTION</i>): Die Variable <i>ackTimeout</i> bestimmt die Überwachungszeit nach der eine Quittierung des gesendeten Pakets spätestens erwartet wird. Bei Überschreiten dieser Zeit wird die Verbindung geschlossen und neu aufgebaut. T#1ms .. T#24d20h30m20s630ms
sTimeSync	struct		Nur mit LCom-Protokoll, (<i>configurations.sConnection.u8ComService = LCOM_LCOM_CONNECTION</i>):
boUsePartnerTimestamps	Bool	FALSE	TRUE: Die empfangene Zeit wird als Systemzeit übernommen. FALSE: Empfangene Uhrzeittelegramme werden ignoriert.
u8SendMode	USInt	0	Sendemodus Zeitsynchronisation 0, LCOM_NO_TIME_SYNC: inaktiv 1, LCOM_CYCLE_TIME_SYNC: zyklische Zeitsynchronisation 2, LCOM_TIME_OF_DAY_TIME_SYNC: Uhrzeit, wann die Zeitsynchronisation erfolgen soll.
tCycleTime	Time	T#1h	Nur bei sendMode 1, LCOM_CYCLE_TIME_SYNC: Sendezykluszeit der Zeitstempel T#1ms .. T#24d20h30m20s630ms
todSendAtTimeOfDay	Time_Of_Day	TOD#05:0 0:0.000	Nur bei sendMode 2, LCOM_TIME_OF_DAY_TIME_SYNC: Uhrzeit, bei der ein Zeitsynchronisationstelegramm verschickt wird.

Die nachfolgende Tabelle zeigt die Struktur von sLCOM_DiagnosticsType.

Tabelle 2-11: Struktur sLCOM_DiagnosticsType

Name	Typ	Kommentar
sLocalConfig	Struct	
sConnection	Struct	
u8ComService	USInt	1: Das Standard TCP Protokoll wird verwendet. 2: Das LCom Protokoll wird verwendet.
bolsClient	Bool	TRUE: Aktiver Verbindungsaufbau als TCP-Client. FALSE: Passiver Verbindungsaufbau als TCP-Server.
u16LocalPort	UInt	Lokaler Port.
u16PartnerPort	UInt	Port der Partnerseite.
au8PartnerIP	Array of USInt	IP Adresse des Partners.
boAcceptUnknownPartner	Bool	TRUE: Vorher nicht konfigurierter Verbindungspartner wird akzeptiert. FALSE: Es wird nur der konfigurierte Verbindungspartner akzeptiert.
boUseLComProtocol	Bool	TRUE: Das LCom-Protokoll wird verwendet. FALSE: Das LCom-Protokoll wird nicht verwendet.
tLifeSignCycleTime	Time	Lokaler Lebenszeichenzyklus.
sSender	Struct	
tCycleTime	Time	Lokaler Sendezyklus.
tAckTimeout	Time	Die Variable <i>ackTimeout</i> bestimmt die lokale Überwachungszeit nach der eine Quittierung des gesendeten Pakets spätestens erwartet wird. Bei Überschreiten dieser Zeit wird die Verbindung geschlossen und neu aufgebaut.
sTimeSync	Struct	
boUsePartnerTimestamps	Bool	TRUE: Die empfangene Zeit wird als Systemzeit übernommen. FALSE: Empfangene Uhrzeittelegramme werden ignoriert.
u8SendMode	USInt	Sendemodus Zeitsynchronisation 0, LCOM_NO_TIME_SYNC: inaktiv 1, LCOM_CYCLE_TIME_SYNC: zyklische Zeitsynchronisation. 2, LCOM_TIME_OF_DAY_TIME_SYNC: Uhrzeit, wann die Zeitsynchronisation erfolgen soll.

Name	Typ	Kommentar
tCycleTime	Time	Sendezykluszeit der Zeitstempel
todSendAtTimeOfDay	Time_Of_Day	Uhrzeit, bei der ein Synchronisations - telegramm verschickt wird.
u32SizeOfSendBuffer	UDInt	Größe des lokalen Sendepuffers in Bytes.
u32SizeOfReceiveBuffer	UDInt	Größe des lokalen Empfangspuffers in Bytes.
sPartnerConfig	Struct	
sConnection	Struct	
tLifeSignCycleTime	Time	Partner Lebenszeichenzyklus.
sSender	Struct	
tCycleTime	Time	Partner Sendezyklus.
tAckTimeout	Time	Die Variable <i>ackTimeout</i> bestimmt die Partner Überwachungszeit nach der eine Quittierung des gesendeten Pakets spätestens erwartet wird. Bei Überschreiten dieser Zeit wird die Verbindung vom Partner geschlossen und neu aufgebaut.
sTimeSync	Struct	
boUsePartnerTimestamps	Bool	TRUE: Die empfangene Zeit wird als Systemzeit übernommen. FALSE: Empfangene Uhrzeittelegramme werden ignoriert.
u8SendMode	USInt	Sendemodus Zeitsynchronisation 0, LCOM_NO_TIME_SYNC: inaktiv 1, LCOM_CYCLE_TIME_SYNC: zyklische Zeitsynchronisation. 2, LCOM_TIME_OF_DAY_TIME_SYNC: Uhrzeit, wann die Zeitsynchronisation erfolgen soll.
u32SizeOfSendBuffer	UDInt	Größe des Sendepuffers in Bytes des Partners.
u32SizeOfReceiveBuffer	UDInt	Größe des Empfangspuffers in Bytes des Partners.
sStatistics	Struct	Statistik
r32AvgCallCycle	Real	Mittelwert zwischen zwei FB-Aufrufen [ms].
r32AvgReceiveMsgCycle	Real	Mittelwert in welchem Zyklus Daten empfangen wurden [ms].
tMaxReceiveMsgCycle	Time	Maximal aufgetretener Empfangszyklus.
boLComProtocolUsed	Bool	TRUE: Das LCom-Protokoll wird verwendet. Mit dem LCom-Protokoll werden zusätzliche Kommunikations-funktionalitäten ermöglicht. FALSE: Das Transportprotokoll TCP wird verwendet.
u8ActiveLComVersion	USInt	Aktive LCom-Protokollversion.
r32AvgMsgSendingTime	Real	Mittelwert wie lange ein Sendevorgang (<i>sending</i> = TRUE) dauert [ms].

Name	Typ	Kommentar
tMaxMsgSendingTime	Time	Maximaler Wert eines Sendevorgangs (<i>sending</i> = TRUE) [ms].
r32AvgMsgReceivingTime	Real	Mittelwert wie lange ein Empfangsvorgang dauert [ms].
tMaxMsgReceivingTime	Time	Maximaler Wert eines Empfangsvorgang [ms].
u32NumberOfSentMessages	UDInt	Anzahl der gesendeten Nachrichten, seit der L/H Flanke am <i>enable</i> Eingang.
u32NumberOfReceivedMessages	UDInt	Anzahl der empfangenen Nachrichten, seit der L/H Flanke am <i>enable</i> Eingang.
u16TotalAckTimeouts	UInt	Anzahl in der die Überwachungszeit (sLCOM_DiagnosticsType.sLocalConfig.sSender.tAckTimeout) überschritten wurde.
u16TotalSendCycleViolations	UInt	Anzahl in der der Sendezyklus (sLCOM_DiagnosticsType.sLocalConfig.sSender.tCycleTime) überschritten wurde
u16TotalReceiveCycleViolations	UInt	Anzahl in der der Empfangszyklus (sLCOM_DiagnosticsType.sPartnerConfig.sSender.tCycleTime) überschritten wurde
u16TotalReconnects	UInt	Anzahl wie oft der FB die TCP Verbindung neu aufgebaut hat.
dtLastConnect	DT	Zeitpunkt, wenn die Verbindung erfolgreich zum Partner aufgebaut wurde.
dtLastTimeSync	DT	Zeitpunkt der letzten erfolgten Zeitsynchronisierung.
u8BufferIndex	USInt	Index auf den letzten Diagnoseeintrag.
asBuffer	Array[0..63] of Struct	Diagnosepuffer
b16Status	Word	Statusmeldung vom FB
dtTimestamp	DT	Zeitstempel bei Auftritt der Meldung.
bolsActive	Bool	TRUE: Zustand ist noch vorhanden. FALSE: Zustand ist nicht mehr vorhanden.
i32SubFunctionErrorID	DInt	ggf. Rückgabewert einer Systemfunktion
r32AdditionalValue1	Real	Zusatzwert 1
r32AdditionalValue2	Real	Zusatzwert 2
r32AdditionalValue3	Real	Zusatzwert 3
tAdditionalValue4	Time	Zusatzwert 4

Die Tabelle "LCom Anwenderkonstanten" zeigt die Übersicht aller Konstanten der Bausteinbibliothek **LCOM**.

Tabelle 2-12 LCom Anwenderkonstanten

Name	Typ	Wert	Kommentar
LCOM_NO_TIME_SYNC	USInt	0	siehe Kapitel 2.2.6
LCOM_TCP_CONNECTION	USInt	1	siehe Kapitel 2.2.1
LCOM_CYCLE_TIME_SYNC	USInt	1	siehe Kapitel 2.2.6
LCOM_LCOM_CONNECTION	USInt	2	siehe Kapitel 2.2.1
LCOM_TIME_OF_DAY_TIME_SYNC	USInt	2	siehe Kapitel 2.2.6
LCOM_STATUS_NO_CALL	Word	16#7000	siehe Kapitel 2.3.1
LCOM_STATUS_FIRST_CALL	Word	16#7001	siehe Kapitel 2.3.1
LCOM_STATUS_INTERMEDIATE_CALL	Word	16#7002	siehe Kapitel 2.3.1
LCOM_STATUS_CLIENT_TRY_CONNECT	Word	16#7003	siehe Kapitel 2.3.1
LCOM_STATUS_SERVER_PORT_OPEN	Word	16#7004	siehe Kapitel 2.3.1
LCOM_STATUS_TCP_CONNECTED	Word	16#7005	siehe Kapitel 2.3.1
LCOM_STATUS_LCOM_CONNECTED_V1	Word	16#7006	siehe Kapitel 2.3.1
LCOM_STATUS_LCOM_CONNECTED_V2	Word	16#7007	siehe Kapitel 2.3.1
LCOM_STATUS_DISCONNECTING	Word	16#7008	siehe Kapitel 2.3.1
LCOM_WARN_SEND_DATA_LENGTH	Word	16#7600	siehe Kapitel 2.3.1
LCOM_WARN_LIFE_SIGN_LIMITATION	Word	16#7610	siehe Kapitel 2.3.1
LCOM_WARN_SEND_CYCLE_LIMITATION	Word	16#7611	siehe Kapitel 2.3.1
LCOM_WARN_ACK_TIMEOUT_LIMITATION	Word	16#7612	siehe Kapitel 2.3.1
LCOM_WARN_SEND_CYCLE_EXPIRED	Word	16#7613	siehe Kapitel 2.3.1
LCOM_WARN_RECEIVE_BUFFER_LEN	Word	16#7614	siehe Kapitel 2.3.1
LCOM_WARN_NO_TIME_SYNC	Word	16#7615	siehe Kapitel 2.3.1
LCOM_ERR_COM_SERVICE	Word	16#8200	siehe Kapitel 2.3.1
LCOM_ERR_CONNECTION_ID	Word	16#8201	siehe Kapitel 2.3.1
LCOM_ERR_INTERFACE_ID	Word	16#8202	siehe Kapitel 2.3.1
LCOM_ERR_LOCAL_PORT	Word	16#8203	siehe Kapitel 2.3.1
LCOM_ERR_PARTNER_IP	Word	16#8204	siehe Kapitel 2.3.1
LCOM_ERR_TIME_SYNC_MODE	Word	16#8205	siehe Kapitel 2.3.1
LCOM_ERR_TIME_CYCLE_TIME	Word	16#8206	siehe Kapitel 2.3.1
LCOM_ERR_TIME_OF_DAY	Word	16#8207	siehe Kapitel 2.3.1
LCOM_ERR_SEND_DATA_TYPE	Word	16#8208	siehe Kapitel 2.3.1
LCOM_ERR_RECEIVE_DATA_TYPE	Word	16#8209	siehe Kapitel 2.3.1
LCOM_ERR_LIFE_SIGN_CYCLE	Word	16#8210	siehe Kapitel 2.3.1
LCOM_ERR_SEND_CYCLE_TIME	Word	16#8211	siehe Kapitel 2.3.1
LCOM_ERR_ACK_TIMEOUT	Word	16#8212	siehe Kapitel 2.3.1
LCOM_ERR_OPEN_CLIENT	Word	16#8600	siehe Kapitel 2.3.1
LCOM_ERR_OPEN_SERVER	Word	16#8601	siehe Kapitel 2.3.1
LCOM_ERR_SEND	Word	16#8602	siehe Kapitel 2.3.1
LCOM_ERR_RECEIVE	Word	16#8603	siehe Kapitel 2.3.1
LCOM_ERR_CLOSE_CONNECTION	Word	16#8604	siehe Kapitel 2.3.1
LCOM_ERR_CLOSE_SERVER	Word	16#8605	siehe Kapitel 2.3.1

Name	Typ	Wert	Kommentar
LCOM_ERR_LIFE_SIGN_EXPIRED	Word	16#8610	siehe Kapitel 2.3.1
LCOM_ERR_ACK_TIMEOUT_EXPIRED	Word	16#8611	siehe Kapitel 2.3.1
LCOM_ERR_ACK_MESSAGE_NUMBER	Word	16#8612	siehe Kapitel 2.3.1
LCOM_ERR_WRONG_LCOM_HEADER	Word	16#8613	siehe Kapitel 2.3.1
LCOM_ERR_WRONG_TELEGRAM_ID	Word	16#8614	siehe Kapitel 2.3.1

3 Arbeiten mit der Bibliothek

Was steht hier?

Dieses Kapitel besteht aus Anleitungen zur Integration der Bibliothek LCom in Ihr SIMOTION SCOUT-Projekt.

3.1 Bibliotheksaufbau

Die Bibliothek LCom ist in folgende Units unterteilt.

Tabelle 3-1 Units FBLLCom_Communication

Unitname	Beschreibung
aVersion	Unit für Änderungsverfolgung
cProtected	Unit für Know-how-geschützte Konstanten
cPublic	Unit für vom Anwender änderbare Konstanten sowie Fehlerkonstanten
dProtected	Unit für Know-how-geschützte Datentypen
fCom	Unit mit dem Know-how- geschützten Funktionsbaustein FBLLCom_Communication
fSub	Unit mit Hilfsfunktionalitäten für die Kommunikation (Know-howgeschützt) Es sind nur die extern zu benutzenden Funktionen dieser Bibliothek dokumentiert. Andere Funktionalitäten werden nur intern in der Bibliothek genutzt und sind daher nicht dokumentiert.

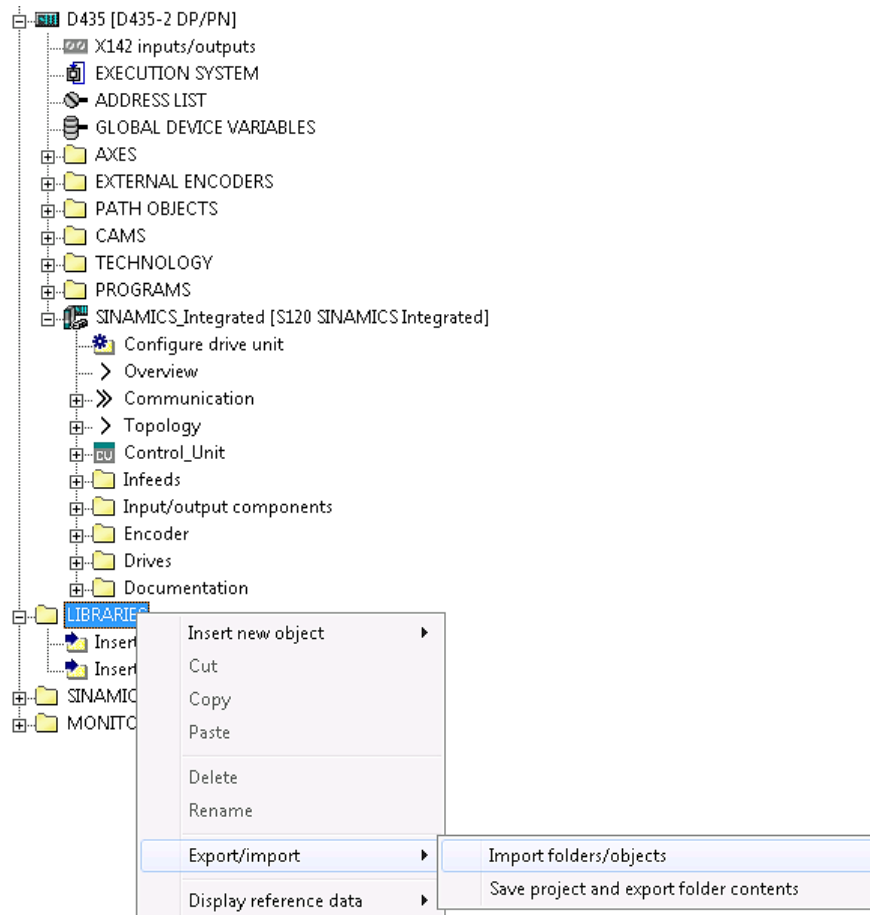
Die Konstanten der Unit cPublic können durch den Anwender geändert werden.

3.2 Integration der Bibliothek in SCOUT

Die Bibliothek LCom wird als XML-Export ausgeliefert. Diese kann im SIMOTION SCOUT oder SCOUT TIA in das SIMOTION Projekt importiert werden.

1. Bitte entpacken Sie die Bibliothek LCom_V2_x_x.zip in ein lokales Verzeichnis.
2. Öffnen Sie mittels Rechtsklick auf den Ordner Bibliotheken im SCOUT das Dialog-Fenster.
3. Wählen Sie unter Export/Import: Import aus und navigieren zu Ihrem lokalen Verzeichnis, in welchem Sie LCom entpackt haben.
4. Selektieren Sie die Datei mit der Endung .xml
5. Die Bibliothek wird jetzt eingebunden.

Abbildung 3-1 LCom Integration in STEP 7



4 Literaturhinweise

Tabelle 4-1 Literaturhinweise

	Themengebiet
\1\	Siemens Industry Online Support http://support.automation.siemens.com
\2\	Link auf die Beitragsseite des Anwendungsbeispiels https://support.industry.siemens.com/cs/ww/de/view/48955385

5 Ansprechpartner

Siemens AG
 Digital Industries Division
 Factory Automation
 Production Machines
 DI FA PMA APC
 Frauenaauracher Str. 80
 91056 Erlangen, Deutschland

mailto: profinet.team.motioncontrol.i-dt@siemens.com

6 Historie

Tabelle 6-1 Historie

Version	Datum	Änderung
V2.0	04/2017	Erste Ausgabe
V2.0	05/2019	Korrekturen im Dokument
V2.0	10/2019	Hinzufügen von Statusmeldung / Anwenderkonstante ERR_LCOM_CLOSE_SERVER