

SICK **CoLa Kommunikationsbaustein**

CCOM_TCP_CP Funktionsbaustein für
Siemens S7-Steuerungen



Inhaltsverzeichnis

1 Zu diesem Dokument	3
1.1 Funktion dieses Dokuments	3
1.2 Zielgruppe	3
2 Allgemeines	4
3 Hardwarekonfiguration	5
3.1 Unterstützte SPS-Steuerungen	5
3.2 Konfiguration in Step7	5
4 Bausteinbeschreibung	9
4.1 Bausteinspezifikationen	9
4.2 Arbeitsweise	10
4.2.1 Empfangen von Leseergebnissen (RD)	10
4.2.2 Gerätekommunikation über CoLa Kommandos (REQ)	10
4.2.3 Timing	11
4.3 Verhalten im Fehlerfall	11
4.4 Parameter	12
4.5 Fehlercodes	15
5 Beispiel	16

1 Zu diesem Dokument

Bitte lesen Sie dieses Kapitel sorgfältig, bevor Sie mit dieser Betriebsanleitung und den SICK CoLa Kommunikationsbaustein arbeiten.

1.1 Funktion dieses Dokuments

Diese Betriebsanleitung beschreibt den Umgang mit dem SICK CCOM_TCP_CP Funktionsbaustein. Sie leitet das technische Personal des Maschinenherstellers bzw. Maschinenbetreibers zur Projektierung und Inbetriebnahme des Funktionsbausteins an.

1.2 Zielgruppe

Diese Betriebsanleitung richtet sich an fachkundiges Personal wie z.B. Techniker oder Ingenieure.

2 Allgemeines

Der Funktionsbaustein CCOM_TCP_CP unterstützt beim Datenaustausch zwischen SICK Geräten und S7-Steuerungen. Der Baustein unterstützt eine TCP-Anbindung über ein Simatic CP-Modul (Kommunikationsprozessor). Steuerungen mit integrierter Ethernet-Schnittstelle werden nicht unterstützt.

Mit diesem Baustein können die folgenden SICK Sensoren angesteuert werden:

- CLV6xx
- LECTOR62x
- RFH62x
- RFU63x

Die folgende Abbildung zeigt die Darstellung des Funktionsbausteins in der Funktionsplan Ansicht (FUP).

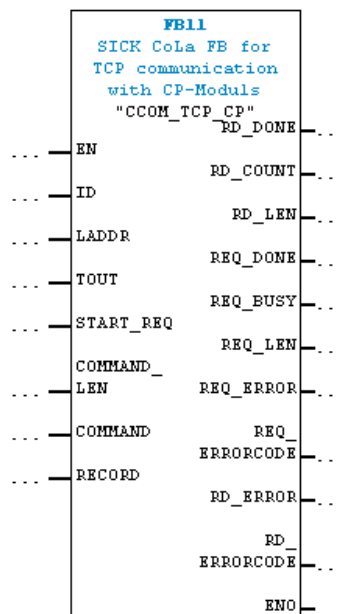


Abbildung 1: Darstellung des Funktionsbausteins in der FUP-Ansicht

Optionale Funktionen:

- Senden von CoLa¹ Kommandos an ein SICK Sensor
- Empfangen von CoLa Antworten eines SICK Sensor
- Empfangen von geräteseitig gesendeten Telegrammen (im SOPAS² Ausgabeformat konfigurierbar)

¹ Die command language (CoLa) ist ein SICK internes Protokoll zur Kommunikation mit SOPAS Geräten

² SOPAS-ET ist ein Engineering Tool zum parametrieren von SICK Sensoren

3 Hardwarekonfiguration

3.1 Unterstützte SPS-Steuerungen

Der Funktionsbaustein darf nur mit Simatic S7-Steuerungen der 300er und der 400er Familie betrieben werden. Es werden nur Steuerungen unterstützt, die für die TCP Kommunikation ein CP-Modul verwenden. Steuerungen mit integrierter Ethernet-Schnittstelle werden nicht unterstützt.

3.2 Konfiguration in Step7

Bevor der Funktionsbaustein verwendet werden kann, muss in eine TCP-Verbindung zum Sensor hergestellt werden. Öffnen Sie hierfür das NetPro Verbindungstool im Simatic Manager (Simatic Manager → Extras → Netz konfigurieren).

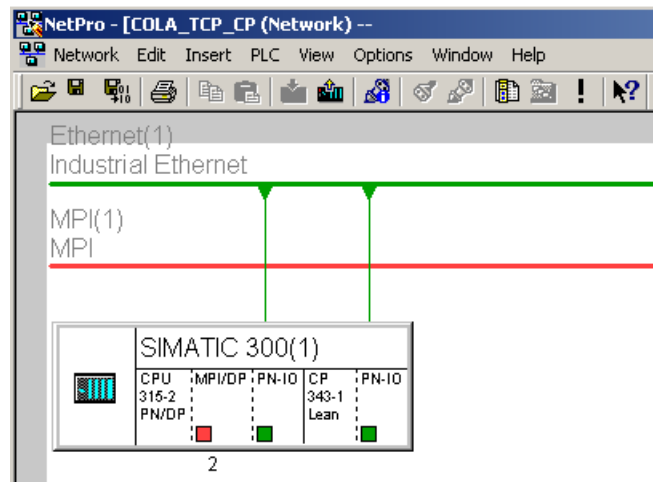


Abbildung 2: NetPro (Step7)

Markieren Sie die CPU in ihrer S7-Station legen sie unter „Einfügen → Neue Verbindung“ eine unspezifizierte TCP-Verbindung an.

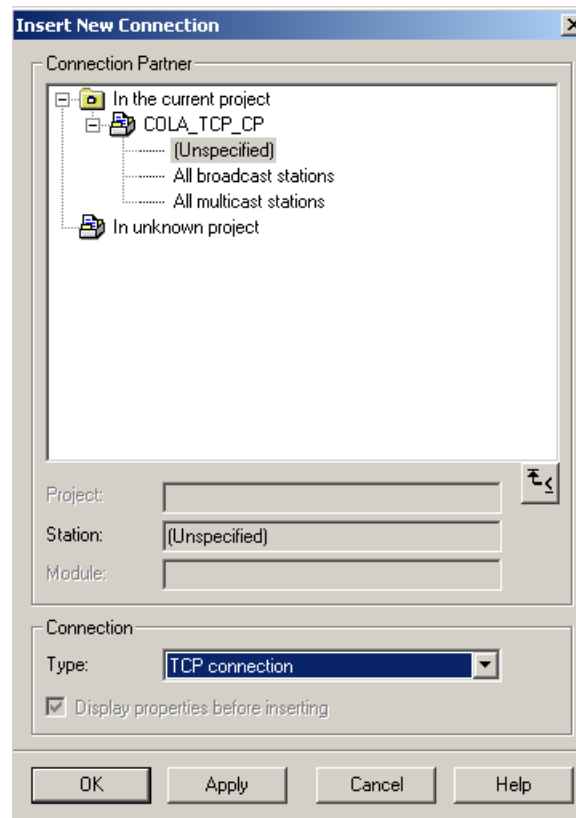


Abbildung 3: Anlegen einer TCP-Verbindung in NetPro (Step7)

Im Eigenschaftsdialog der TCP-Verbindung muss im Register „Allgemein“ die Checkbox „Aktiver Verbindungsaufbau“ aktiviert werden. Auf der Rechten Seite im Dialog werden die Verbindungsparameter für die TCP-Verbindung angezeigt. Diese Parameterwerte müssen dem CCOM_TCP_CP Funktionsbaustein beim Bausteinaufruf übergeben werden (Parameter: ID / LADDR).

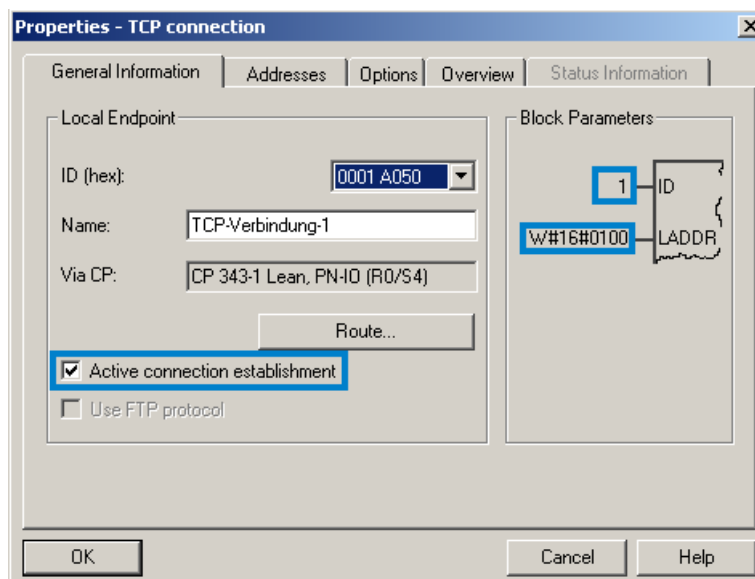


Abbildung 4: NetPro TCP-Verbindungseinstellungen (Step7)

Im Register „Adressen“ muss die IP-Adresse und der verwendete Port des SICK Sensors angegeben werden. Standardmäßig verwenden SICK-Sensoren zum kommunizieren den Port 2112.

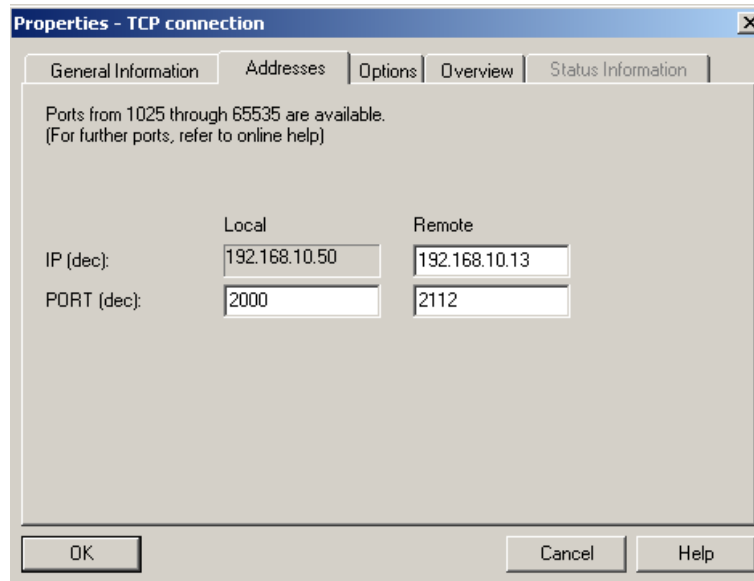


Abbildung 5: Adressen der TCP-Verbindung (Step7)

Im Register „Optionen“ ist die Betriebsart „Send/Recv“ einzustellen.

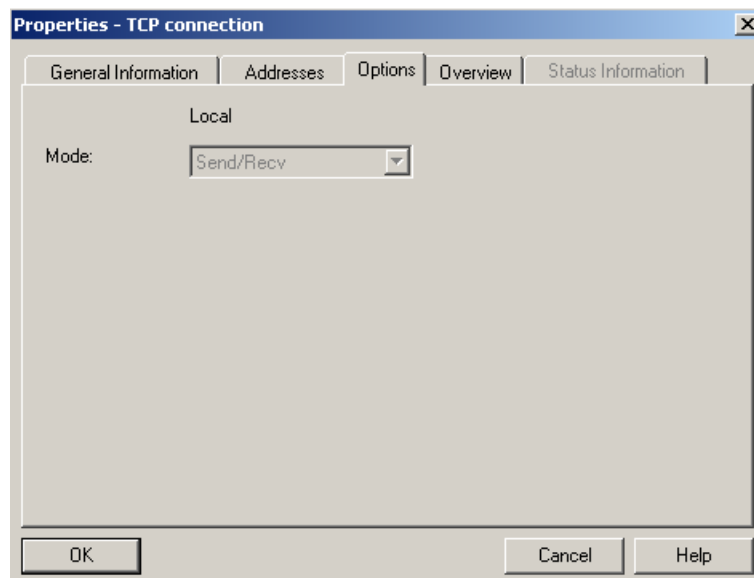


Abbildung 6: Betriebsart der TCP-Verbindung (Step7)

Nachdem der Eigenschaftsdialog mit „OK“ geschlossen wird, wird automatisch eine TCP-Verbindung in der Verbindungstabelle angezeigt. Speichern und übersetzen Sie nun die Station und laden Sie anschließend die Verbindung in Ihre S7-Steuerung. Ggf. ist ein Neustart des Sensors nötig, um die TCP-Verbindung herzustellen.

Um die projektierte TCP-Verbindung zu diagnostizieren kann unter „Zielsystem → Verbindungsstatus aktivieren“ der Verbindungsstatus abgefragt werden.

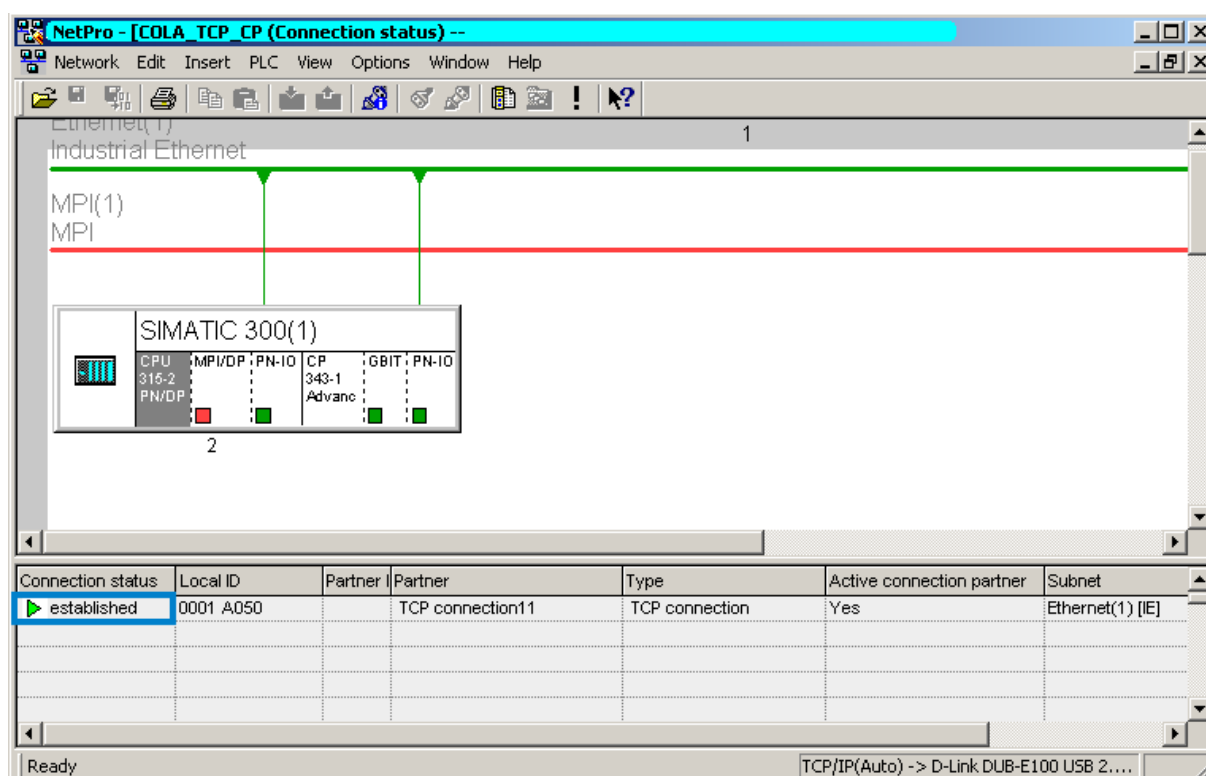


Abbildung 7: Verbindungsstatus der TCP-Verbindung (Step7)

4 Bausteinbeschreibung

Der Funktionsbaustein CCOM_TCP_CP (FB11) vereinfacht die Benutzung von SICK Sensoren an S7-Steuerungen. Der Baustein ermöglicht das Senden und Empfangen von CoLa Kommandos über eine in NetPro projektierte TCP-Verbindung.

Der Baustein unterstützt bei folgenden Aufgaben:

- Senden von CoLa Kommandos an ein SICK Sensor
- Empfangen von CoLa Antworten eines SICK Sensor
- Empfangen von geräteseitig gesendeten Telegrammen (im SOPAS Ausgabeformat konfigurierbar)

Der Funktionsbaustein ist ein asynchron arbeitender FB d.h. die Bearbeitung erstreckt sich über mehrere FB-Aufrufe. Der Funktionsbaustein muss hierfür zyklisch im Anwenderprogramm aufgerufen werden.

Der CCOM_TCP_CP Baustein (FB11) kapselt den Siemens Funktionsbaustein AG_RECV_TCP_xVar (FB103) und die Funktionen AG_SEND (FC5) / AG_RECV (FC6), die für die Kommunikation zwischen SPS und Sensor verwendet werden.

4.1 Bausteinspezifikationen

Bausteinnummer:	FB11
Bausteinname:	CCOM_TCP_CP
Version:	1.0
Aufgerufene Bausteine:	FC5 (AG_SEND) FC6 (AG_RECV) FB103 (AG_RECV_TCP_xVAR) SFB4 (TON)
Verwendete Datenbausteine:	-
Bausteinaufruf:	Zyklisch
Verwendete Merker:	keine
Verwendete Zähler:	keine
Verwendetes Register:	AR1, AR2 (für Multiinstanzen)
Erstellsprache:	Step7-AWL

Die Bausteine FC5, FC6 und FB103 werden von der Siemens Bibliothek bereitgestellt.

4.2 Arbeitsweise

Um den CCOM_TCP_CP Baustein einsetzen zu können, müssen zunächst die folgenden Parameter angegeben werden.

ID: Verbindungs-ID der TCP-Verbindung. Der Parameter wird in den NetPro Verbindungseigenschaften angegeben (siehe Abbildung 4).

LADDR: Baugruppen-Anfangsadresse. Bei der Konfiguration des CP-Moduls mit der Step7 Hardware Konfiguration wird die Baugruppen-Anfangsadresse in der Konfigurationstabelle angegeben. Der Parameter wird ebenfalls in den NetPro Verbindungseigenschaften angezeigt (siehe Abbildung 4).

COMMAND: Der Pointer zeigt auf den Datenbereich, in den das CoLa Kommando abgelegt ist. Der Datenbereich muss vom Programmierer selbst erstellt werden (z.B. Datenbaustein mit einem Array of CHAR).

COMMAND_LEN: Zeichenlänge des zu übertragenden CoLa Kommandos.

RECORD: Der Pointer zeigt auf den Datenbereich, in den die vom Gerät gesendeten Telegramme abgelegt werden. Der Datenbereich muss vom Programmierer selbst erstellt werden (z.B. Datenbaustein mit einem Array of BYTE).

4.2.1 Empfangen von Leseergebnissen (RD)

Daten die geräteseitig gesendet werden (RD), werden in den Record geschrieben, sobald der Funktionsbaustein neue Daten empfangen hat. Das Bit RD_DONE zeigt für einen SPS Zyklus den Empfang neuer Daten an. Sobald neue Daten empfangen wurden, wird der Zähler RD_COUNT inkrementiert. Die jeweilige Bytelänge des zuletzt empfangenden Telegramms kann dem Parameter RD_LEN entnommen werden.

4.2.2 Gerätekommunikation über CoLa Kommandos (REQ)

Bei der Kommunikation via CoLa Kommandos, wird das im COMMAND definierte Kommando zum Gerät übertragen. Die resultierende Antwort wird in den vom Pointer RECORD definierten Bereich abgelegt. CoLa Kommandos werden immer mit Steuerzeichen ([ETX], [STX] Rahmung) übertragen.

Sie starten die Übertragung, indem Sie den Parameter START_REQ mit einer positiven Flanke antriggern. Solange noch keine gültige Antwort auf das gesendete CoLa Kommando eingetroffen ist, wird dies über den Parameter REQ_BUSY signalisiert. Sollte innerhalb der Timeout Zeit (TOUT) keine Antwort eingetroffen sein, wird die Bearbeitung mit einem Timeout Fehler (REQ_ERRORCODE) abgebrochen. Der Ausgangsparameter REQ_DONE zeigt an, dass auf ein CoLa Kommando eine Antwort empfangen wurde (REQ_DONE = TRUE).

4.2.3 Timing

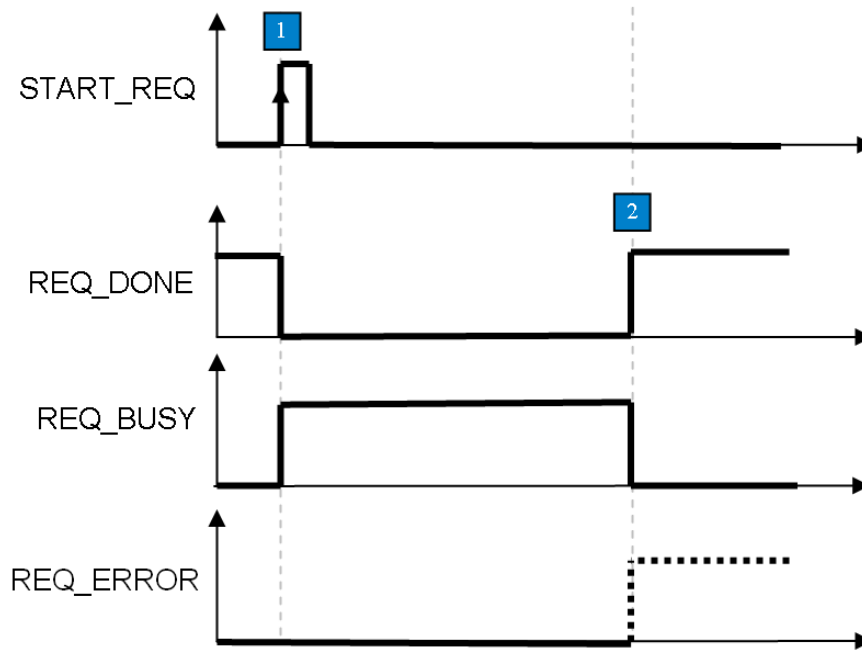


Abbildung 8: Timing Diagramm

1: Anforderung durch eine positive Flanke an START_REQ. Das vom Parameter COMMAND referenzierte CoLa Kommando wird zum Sensor gesendet. Es kann immer nur ein Kommando zeitgleich gesendet werden.

2: Wenn das Kommando versendet ist und die Antwort empfangen wurden, wird die Aktion mit „REQ_DONE“ beendet. Wenn die Aktion fehlerhaft verläuft, wird mit „REQ_ERROR“ beendet. Bei Abbruch mit „REQ_ERROR“ enthält „REQ_ERRORCODE“ den aufgetretenen Fehlercode.

4.3 Verhalten im Fehlerfall

Im Fehlerfall signalisieren die Errorbits REQ_ERROR oder RD_ERROR, dass ein Fehler aufgetreten ist. In diesem Fall wird über den Parameter REQ_ERRORCODE bzw. RD_ERRORCODE ein Fehlercode ausgegeben. Das Errorbit REQ_ERROR bleibt solange gesetzt, bis ein neuer Auftrag gestartet wird. Der Parameter RD_ERROR ist immer nur für einen SPS-Zyklus aktiv und wird danach zurückgesetzt, sollte der Fehler nicht weiter bestehen.

4.4 Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E,M,D,L, Konst.	Enable Eingang (KOP und FUP).
ID	INPUT	INT	E,M,D,L, Konst.	Verbindungs-ID zur projektierten TCP-Verbindung (siehe NetPro Verbindungseinstellungen Abbildung 4).
LADDR	INPUT	WORD	E,M,D,L, Konst.	Baugruppen-Anfangsadresse des CP-Moduls (siehe NetPro Verbindungseinstellungen Abbildung 4).
TOUT	INPUT	TIME	E,M,D,L, Konst.	<p>Zeit, nachdem ein Timeout-Fehler ausgelöst wird.</p> <p>Wenn dieser Parameter nicht beschaltet ist, beträgt die Timeout Zeit standardmäßig 5Sekunden.</p> <p>Bitte beachten Sie, dass einige CoLa Kommandos eine längere Bearbeitungszeit benötigen (z.B. Speicherkommandos).</p>
START_REQ	INPUT	BOOL	E,M,D,L	Positive Flanke: Senden ein CoLa Kommando und wartet auf die entsprechende Antwort.
COMMAND	INPUT	ANY	D	<p>Pointer auf den Bereich, welcher das zu übertragende CoLa Kommando enthält. Es ist nur das Datentyp BYTE zulässig.</p> <p><u>Hinweis:</u> Beachten Sie, dass der Parameter immer die vollständige Angabe der DB-Parameter erfordert (Bsp.: P#DB13.DBX0.0 BYTE 100). Das Weglassen einer expliziten DB-Nr. ist unzulässig und führt zu einem Bausteinfehler.</p>
COMMAND_LEN	INPUT	INT	E,M,D,L, Konst.	Anzahl der Bytes, die das zu sendende CoLa Kommando enthält, worauf der Pointer #COMMAND referenziert.

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RECORD	INPUT	ANY	D	<p>Pointer auf den Bereich, in denen die vom Gerät gesendeten Telegramme abgelegt werden sollen. Es ist nur der Datentyp BYTE zulässig.</p> <p><u>Hinweis:</u> Beachten Sie, dass der Parameter immer die vollständige Angabe der DB-Parameter erfordert (Bsp.: P#DB13.DBX0.0 BYTE 100). Das Weglassen einer expliziten DB-Nr. ist unzulässig und führt zu einem Bausteinfehler.</p>
RD_DONE	OUTPUT	BOOL	A,M,D,L	<p>Positive Flanke: Ein vom Gerät gesendetes Leseergebnis wurde empfangen (Formatierung siehe SOPAS Ausgabeformat).</p> <p>Wenn ein Leseergebnis empfangen wurde, ist das Bit für jeweils einen SPS-Zyklus gesetzt. Das Leseergebnis steht im vom Parameter #RECORD referenzierten Speicherbereich zur Verfügung.</p>
RD_COUNT	OUTPUT	BYTE	A,M,D,L	Zählt die Anzahl der Leseergebnisse die empfangen wurden. Der Zähler zählt von 0...255 (Dez.). Bei einem Überlauf fängt der Zähler wieder bei null an.
RD_LEN	OUTPUT	INT	A,M,D,L	Gibt die Bytelänge des empfangenden Leseergebnisses an.
REQ_DONE	OUTPUT	BOOL	A,M,D,L	<p>Zeigt an, ob ein CoLa Kommando abgeschickt und eine Antwort empfangen wurde.</p> <p>TRUE: Bearbeitung abgeschlossen FALSE: Bearbeitung noch nicht abgeschlossen</p> <p>Die Kommandoantwort steht im, vom Parameter #RECORD referenzierten, Speicherbereich zur Verfügung.</p>
REQ_BUSY	OUTPUT	BOOL	A,M,D,L	REQ Auftrag ist in Bearbeitung.
REQ_LEN	OUTPUT	INT	A,M,D,L	Länge eines Antworttelegramms in BYTE.
REQ_ERROR	OUTPUT	BOOL	A,M,D,L	<p>REQ Fehler Status:</p> <p>0: Kein Fehler 1: Abbruch mit Fehler</p>

Parameter	Dekla- ration	Daten- typ	Speicher- bereich	Beschreibung
RD_ERROR	OUTPUT	BOOL	A,M,D,L	RD Fehler Status: 0: Kein Fehler 1: Abbruch mit Fehler
REQ_ERROR CODE	OUTPUT	WORD	A,M,D,L	REQ Fehlerstatus (siehe Fehlercodes).
RD_ERROR CODE	OUTPUT	WORD	A,M,D,L	RD Fehlerstatus (siehe Fehlercodes).
ENO	OUTPUT	BOOL	A,M,D,L	Enable Ausgang (KOP und FUP).

4.5 Fehlercodes

Die Parameter REQ_ERRORCODE und RD_ERRORCODE enthalten die folgenden Fehlerinformationen:

Fehlercode	Kurzbeschreibung	Beschreibung
W#16#0000	Kein Fehler	Kein Fehler
W#16#0001	Ungültiger Speicherbereich #RECORD Pointer angegeben	Ungültiger Speicherbereich des angegebenen ANY-Pointers. Dem Pointer muss ein DB zugeordnet werden.
W#16#0002	Ungültige Pointerlänge #RECORD angegeben	Die Länge des referenzierten Datenbaustein ist kürzer als die vom Pointer vorgegebene Länge.
W#16#0003	Ungültiger Speicherbereich #COMMAND Pointer angegeben	Ungültiger Speicherbereich des angegebenen ANY-Pointers. Dem Pointer muss ein DB zugeordnet werden.
W#16#0004	Ungültige Pointerlänge #COMMAND angegeben	Die Länge des referenzierten Datenbaustein ist kürzer als die vom Pointer vorgegebene Länge.
W#16#0005	Timeout	Der Auftrag konnte innerhalb der gewählten Timeoutzeit nicht ausgeführt werden. Dies könnte folgende Ursache haben: <ul style="list-style-type: none"> - Gerät ist nicht mit der SPS Verbunden - Kommunikationsparameter fehlerhaft - Verwendung von CoLa Kommandos, die keine Antwort (Echo) zurücksenden - Bearbeitungszeit des Kommandos > Timeout Zeit
W#16#0006	Ungültige Kommandolänge	Die Länge des zu sendenden Kommandos ist länger als die angegebene Kommandolänge (COMMAND_LEN).
W#16#0007	Ungültiges CoLa Kommando	Das angegebene CoLa Kommando hat keine [STX] [ETX] Rahmung.
W#16#000A	Empfangendes Telegramm > #RECORD Länge	Das empfangende Telegramm ist länger als die angegebene #RECORD Länge.
W#16#000B	Telegramm ohne Steuerzeichen empfangen	<ul style="list-style-type: none"> - Es wurde ein Telegramm ohne [STX] [ETX] Rahmung empfangen. - Das empfangende Telegramm ist länger als die angegebene #RECORD Länge.
W#16#8XXX	AG_RECV_TCP_xVAR / AG_SEND Fehler	Fehlerbeschreibung siehe Step7 Hilfe.

5 Beispiel

Abbildung 9 zeigt eine Beispielbeschaltung des CCOM_TCP_CP FBs. Die TCP Verbindung zum SICK Sensor wurde im Vorfeld mit NetPro projektiert und die Verbindungsparameter (ID) und (LADDR) dem Funktionsbaustein übergeben.

Programmaufruf:

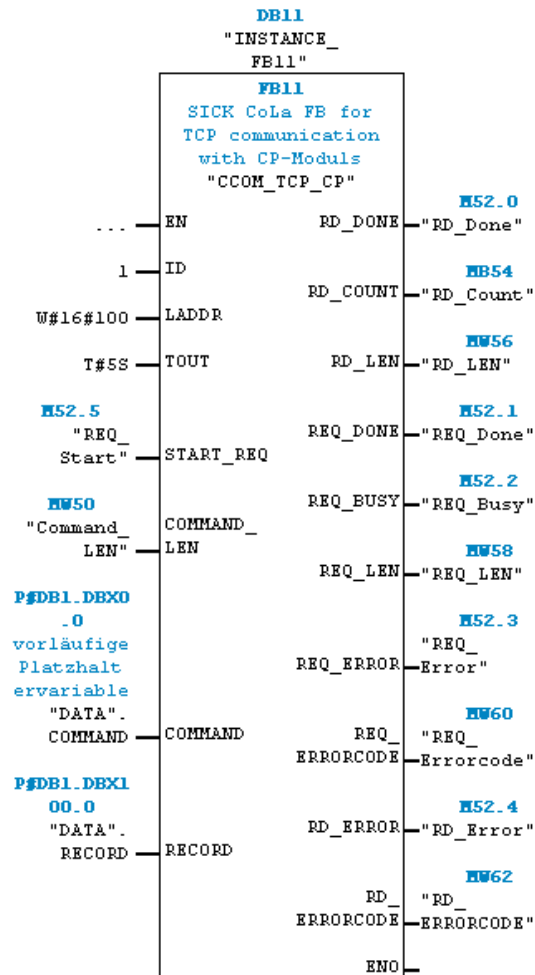




Abbildung 9: Beispielbeschaltung CCOM_TCP_CP FB

Variablen-tabelle zum ausführen eines CoLa Kommandos:

// CCOM TCP CP function block				
M 10.0	"REQ_Start"	BOOL	 true	
// Reading Result Status				
M 12.0	"RD_Done"	BOOL	false	
MB 16	"RD_Count"	DEC	23	
M 12.3	"RD_Error"	BOOL	false	
MW 24	"RD_Errorcode"	HEX	VW#16#0000	
// Requesting Result Status				
M 12.4	"REQ_Done"	BOOL	 true	
M 12.1	"REQ_Busy"	BOOL	false	
M 12.2	"REQ_Error"	BOOL	false	
MW 22	"REQ_Errorcode"	HEX	VW#16#0000	
// Command				
MW 14	"Command_Len"	DEC	15	15
DB1.DBB 0	"DATA".COMMAND[1]	CHARACTER	'_'	'_'
DB1.DBB 1	"DATA".COMMAND[2]	CHARACTER	's'	's'
DB1.DBB 2	"DATA".COMMAND[3]	CHARACTER	'm'	'm'
DB1.DBB 3	"DATA".COMMAND[4]	CHARACTER	'n'	'n'
DB1.DBB 4	"DATA".COMMAND[5]	CHARACTER	' '	' '
DB1.DBB 5	"DATA".COMMAND[6]	CHARACTER	'm'	'm'
DB1.DBB 6	"DATA".COMMAND[7]	CHARACTER	't'	't'
DB1.DBB 7	"DATA".COMMAND[8]	CHARACTER	'c'	'c'
DB1.DBB 8	"DATA".COMMAND[9]	CHARACTER	'g'	'g'
DB1.DBB 9	"DATA".COMMAND[10]	CHARACTER	'a'	'a'
DB1.DBB 10	"DATA".COMMAND[11]	CHARACTER	't'	't'
DB1.DBB 11	"DATA".COMMAND[12]	CHARACTER	'e'	'e'
DB1.DBB 12	"DATA".COMMAND[13]	CHARACTER	'o'	'o'
DB1.DBB 13	"DATA".COMMAND[14]	CHARACTER	'n'	'n'
DB1.DBB 14	"DATA".COMMAND[15]	CHARACTER	'l'	'l'
DB1.DBB 15	"DATA".COMMAND[16]	CHARACTER	B#16#00	
DB1.DBB 16	"DATA".COMMAND[17]	CHARACTER	B#16#00	
DB1.DBB 17	"DATA".COMMAND[18]	CHARACTER	B#16#00	
DB1.DBB 18	"DATA".COMMAND[19]	CHARACTER	B#16#00	
DB1.DBB 19	"DATA".COMMAND[20]	CHARACTER	B#16#00	

Das CoLa Kommando (hier: '[STX]sMN mTCgateon[ETX]') wird ausgeführt, sobald das Bit „START_REQ“ mit einer positiven Flanke angesteuert wird.

Variablentabelle für eingehende Kommandoantworten:

// Record				
MW	18	"RD_Len"	DEC	50
MW	20	"REQ_Len"	DEC	17
DB1.DBB	100	"DATA".RECORD[1]	CHARACTER	'j'
DB1.DBB	101	"DATA".RECORD[2]	CHARACTER	's'
DB1.DBB	102	"DATA".RECORD[3]	CHARACTER	'A'
DB1.DBB	103	"DATA".RECORD[4]	CHARACTER	'N'
DB1.DBB	104	"DATA".RECORD[5]	CHARACTER	' '
DB1.DBB	105	"DATA".RECORD[6]	CHARACTER	'm'
DB1.DBB	106	"DATA".RECORD[7]	CHARACTER	'T'
DB1.DBB	107	"DATA".RECORD[8]	CHARACTER	'C'
DB1.DBB	108	"DATA".RECORD[9]	CHARACTER	'g'
DB1.DBB	109	"DATA".RECORD[10]	CHARACTER	'a'
DB1.DBB	110	"DATA".RECORD[11]	CHARACTER	't'
DB1.DBB	111	"DATA".RECORD[12]	CHARACTER	'e'
DB1.DBB	112	"DATA".RECORD[13]	CHARACTER	'o'
DB1.DBB	113	"DATA".RECORD[14]	CHARACTER	'n'
DB1.DBB	114	"DATA".RECORD[15]	CHARACTER	' '
DB1.DBB	115	"DATA".RECORD[16]	CHARACTER	'i'
DB1.DBB	116	"DATA".RECORD[17]	CHARACTER	'l'
DB1.DBB	117	"DATA".RECORD[18]	CHARACTER	B#16#00
DB1.DBB	118	"DATA".RECORD[19]	CHARACTER	B#16#00
DB1.DBB	119	"DATA".RECORD[20]	CHARACTER	B#16#00

Die Kommandoantwort (REQ) auf ein gesendete Kommando (hier: '[STX]sAN mTCgateon 1[ETX]') steht im Record Bereich zur Verfügung, wenn sich der Wert des Ausgangsbits „REQ_DONE“ von FALSE auf TRUE ändert (positive Flanke). Der Parameter „REQ_LEN“ gibt an, wie viele Bytes empfangen wurden bzw. gültig sind.

Variablentabelle für eingehende Leseergebnisse:

// Record

MV	18	"RD_Len"	DEC	11
MV	20	"REQ_Len"	DEC	17
DB1.DBB	100	"DATA".RECORD[1]	CHARACTER	'1'
DB1.DBB	101	"DATA".RECORD[2]	CHARACTER	'1'
DB1.DBB	102	"DATA".RECORD[3]	CHARACTER	'2'
DB1.DBB	103	"DATA".RECORD[4]	CHARACTER	'3'
DB1.DBB	104	"DATA".RECORD[5]	CHARACTER	'4'
DB1.DBB	105	"DATA".RECORD[6]	CHARACTER	'5'
DB1.DBB	106	"DATA".RECORD[7]	CHARACTER	'6'
DB1.DBB	107	"DATA".RECORD[8]	CHARACTER	'7'
DB1.DBB	108	"DATA".RECORD[9]	CHARACTER	'8'
DB1.DBB	109	"DATA".RECORD[10]	CHARACTER	'9'
DB1.DBB	110	"DATA".RECORD[11]	CHARACTER	'L'
DB1.DBB	111	"DATA".RECORD[12]	CHARACTER	'e'
DB1.DBB	112	"DATA".RECORD[13]	CHARACTER	'o'
DB1.DBB	113	"DATA".RECORD[14]	CHARACTER	'n'
DB1.DBB	114	"DATA".RECORD[15]	CHARACTER	' '
DB1.DBB	115	"DATA".RECORD[16]	CHARACTER	'1'
DB1.DBB	116	"DATA".RECORD[17]	CHARACTER	'L'
DB1.DBB	117	"DATA".RECORD[18]	CHARACTER	B#16#00
DB1.DBB	118	"DATA".RECORD[19]	CHARACTER	B#16#00

Daten die geräteseitig gesendet werden (RD), werden in den Record geschrieben, sobald der Funktionsbaustein neue Daten empfangen hat. Das Bit „RD_DONE“ zeigt für einen SPS Zyklus den Empfang neuer Daten an (Signalwechsel von FALSE auf TRUE). Sobald neue Daten empfangen wurden, wird der Zähler RD_COUNT inkrementiert. Der Parameter „RD_LEN“ gibt an, wie viele Bytes empfangen wurden bzw. gültig sind.