

# **SICK** **CoLa Kommunikationsbaustein**

CCOM\_PNDP Funktionsbaustein für  
Siemens Step7 Steuerungen



# Inhaltsverzeichnis

<b>1 Zu diesem Dokument .....</b>	<b>3</b>
1.1 Funktion dieses Dokuments .....	3
1.2 Zielgruppe .....	3
<b>2 Allgemeines .....</b>	<b>4</b>
<b>3 Hardwarekonfiguration .....</b>	<b>5</b>
3.1 Unterstützte SPS-Steuerungen .....	5
3.2 Unterstützte Feldbus Gateways / Sensoren .....	5
3.3 Konfiguration in Step7 .....	5
<b>4 Bausteinbeschreibung .....</b>	<b>7</b>
4.1 Bausteinspezifikationen .....	7
4.2 Arbeitsweise .....	8
4.2.1 Empfangen von Leseergebnissen (RD) .....	8
4.2.2 Gerätekommunikation über CoLa Kommandos (REQ) .....	8
4.2.3 Timing .....	9
4.3 Verhalten im Fehlerfall .....	9
4.4 Rücksetzen der Kommunikation .....	9
4.5 Parameter .....	10
4.6 Fehlercodes .....	13
<b>5 Beispiel .....</b>	<b>14</b>

# **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\_PNDP 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\_PNDP unterstützt beim Datenaustausch zwischen SICK Geräten und S7-Steuerungen. Es werden nur S7-300/S7-400 Steuerungen unterstützt, die mit einer integrierten Feldbus Schnittstelle ausgerüstet sind (Profibus / Profinet).

Mit diesem Baustein können die folgenden SICK Sensoren angesteuert werden:

- CLV6xx
- LECTOR62x
- RFH62x
- RFU63x
- MSC800

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

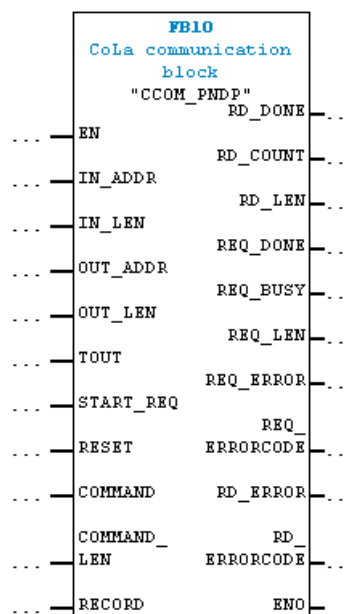


Abbildung 1: Darstellung des Funktionsbausteins in FUP

### Optionale Funktionen:

- Senden von CoLa<sup>i</sup> Kommandos an ein SICK Sensor
- Empfangen von CoLa Antworten eines SICK Sensor
- Empfangen von geräteseitig gesendeten Telegrammen (im SOPAS<sup>ii</sup> Ausgabeformat konfigurierbar)

<sup>i</sup> Die command language (CoLa) ist ein SICK internes Protokoll zur Kommunikation mit SOPAS Geräten

<sup>ii</sup> 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 400er Familie betrieben werden. Es werden nur Steuerungen unterstützt, die die verwendete Feldbuschnittstelle direkt integriert haben. Ein Datenaustausch über einen Kommunikationsprozessor (CP- Baugruppe) wird nicht unterstützt.

### 3.2 Unterstützte Feldbus Gateways / Sensoren

Der SICK Sensor kommuniziert über einen Feldbus (Profibus/Profinet) mit der Steuerung. Sollte der Sensor die oben genannten Feldbusse nicht unterstützen, können Gateway-Module eingesetzt werden.

Folgende Gateways werden vom Funktionsbaustein unterstützt:

- CDM 425 (Profinet)
- CDF 600 (Profibus)
- CDM 420 + CMF400 (Profibus)

### 3.3 Konfiguration in Step7

Bevor der Funktionsbaustein verwendet werden kann, muss in der Step7 Hardwarekonfiguration der entsprechende Sensor bzw. das entsprechende Gateway projektiert werden. Der erste Schritt ist, die entsprechende Gerätestammdatei (GSD / GSDML) in die Step7 Hardwarebibliothek zu importieren.

Der Funktionsbaustein ist speziell für den Handshake mode ausgelegt. Bitte nur Module aus der Kategorie „Handshake v2.1“ verwenden, die mit einer Länge zwischen 8...128 Bytes definiert sind. Die verwendeten Adressen dürfen im Peripheriebereich oder außerhalb projektiert werden. Eine Adresszuweisung auf Peripheriebereiche, denen ein Teilprozessabbild mit OB6x-Anbindung (Taktsynchronalarmlenken) zugeordnet ist, darf nicht verwendet werden.

Abbildung 2 zeigt eine Beispielkonfiguration mit einem CDM-425 Profinet Gateway.

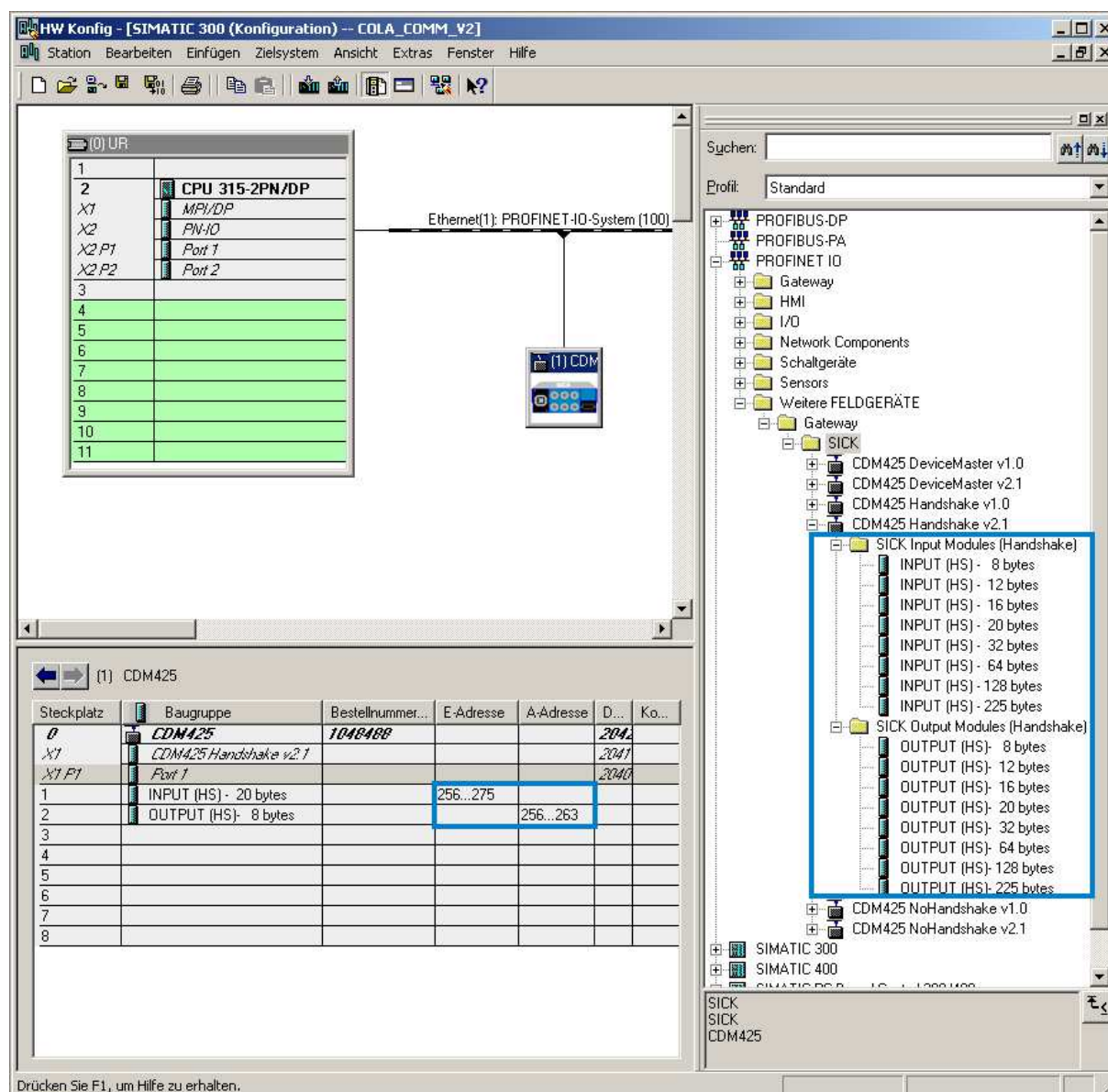


Abbildung 2: Step7 Hardwarekonfiguration

## 4 Bausteinbeschreibung

Der Funktionsbaustein CCOM\_PNDP (FB10) vereinfacht die Benutzung von SICK Sensoren an S7-Steuerungen. Der Baustein ermöglicht das Senden und Empfangen von CoLa Telegrammen über eine in der Hardwarekonfiguration projektierten Profibus/Profinet 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 Baustein fragmentiert die Daten automatisch, sobald diese nicht in einem Zyklus übertragen / empfangen werden können.

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\_PNDP Baustein (FB10) kapselt die Siemens Funktionsbausteine DPRD\_DAT (SFC14) und DPWR\_DAT (SFC15), die für die Kommunikation zwischen SPS und Sensor verwendet werden.

### 4.1 Bausteinspezifikationen

Bausteinnummer:	FB10
Bausteinname:	CCOM_PNDP
Version:	2.0
Aufgerufene Bausteine:	SFC14 (DPRD_DAT) SFC15 (DPWR_DAT) SFC20 (BLKMOV) SFB4 (TON)
Verwendete Datenbausteine:	-
Bausteinanruf:	Zyklisch
Verwendete Merker:	keine
Verwendete Zähler:	keine
Verwendetes Register:	AR1, AR2 (für Multiinstanzen)
Erstellsprache:	Step7-AWL

Die im Funktionsbaustein verwendeten Systemfunktionen (SFCs) müssen auf der jeweils verwendeten Steuerung vorhanden sein.

## 4.2 Arbeitsweise

Um den CCOM\_PNDP Baustein einsetzen zu können, müssen zunächst die folgenden Parameter angegeben werden.

IN\_ADDR: Projektierte Anfangsadresse aus dem E-Bereich des verwendeten Moduls. Die Eingangsadresse wird mit der Hardwareprojektierung festgelegt (siehe *Abbildung 2*).

IN\_LEN: Länge des verwendeten Input-Moduls in der Hardwarekonfiguration. Die Länge des Eingangsmoduls wird mit der Hardwareprojektierung festgelegt (siehe *Abbildung 2*).

OUT\_ADDR: Projektierte Anfangsadresse aus dem A-Bereich des verwendeten Moduls. Die Ausgangsadresse wird mit der Hardwareprojektierung festgelegt (siehe Kapitel 3.3).

OUT\_LEN: Länge des verwendeten Output-Moduls in der Hardwarekonfiguration. Die Länge des Ausgangsmoduls wird mit der Hardwareprojektierung festgelegt (siehe Kapitel 3.3).

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). Das Kommando muss ohne [STX] / [ETX] Rahmung angegeben werden.

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.

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

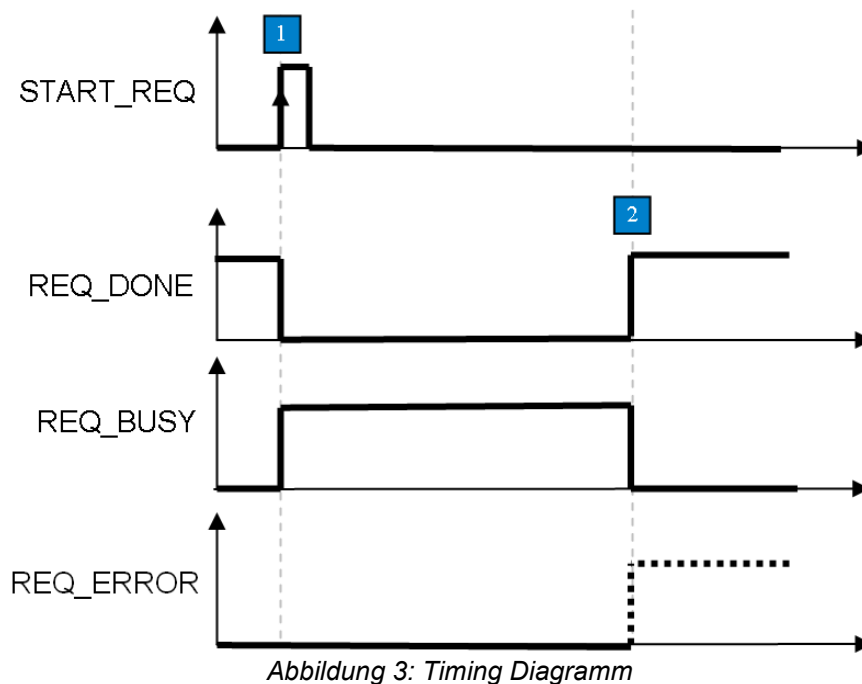


Abbildung 3: Timing Diagramm

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 Rücksetzen der Kommunikation

Über das RESET Bit ist es möglich die Kommunikation zwischen dem Gateway / Sensor und der PLC zurückzusetzen. Hierbei werden die ersten 8 Bytes der Peripherieausgangsseite für eine Sekunde mit null initialisiert. Der Reset-Befehl wird ausgeführt, sobald das RESET = TRUE und START\_REQ mit einer positiven Flanke angetriggert wird. Das REQ\_BUSY Bit signalisiert, dass der Befehl bearbeitet wird. Ist die Reset-Routine abgeschlossen, wird das REQ\_DONE Bit gesetzt.

## 4.5 Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	INPUT	BOOL	E,M,D,L, Konst.	Enable
IN_ADDR	INPUT	WORD	E,M,D,L, Konst.	Projektierte Anfangsadresse aus dem E-Bereich des verwendeten Moduls.
IN_LEN	INPUT	INT	E,M,D,L, Konst.	Länge des verwendeten Input-Moduls in der Hardwarekonfiguration.
OUT_ADDR	INPUT	WORD	E,M,D,L, Konst.	Projektierte Anfangsadresse aus dem A-Bereich des verwendeten Moduls.
OUT_LEN	INPUT	INT	E,M,D,L, Konst.	Länge des verwendeten Output-Moduls in der Hardwarekonfiguration.
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.
RESET	INPUT	BOOL	E,M,D,L, Konst.	Rücksetzen der Kommunikation (HS-Counter des Datenflussprotokolls)
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>Das Kommando muss ohne [STX] / [ETX] Rahmung angegeben werden.</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>
RD_ERROR	OUTPUT	BOOL	A,M,D,L	<p>RD Fehler Status:</p> <p>0: Kein Fehler 1: Abbruch mit Fehler</p>

Parameter	Dekla- ration	Daten- typ	Speicher- bereich	Beschreibung
REQ_ERRO R_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).

## 4.6 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"> <li>- Gerät ist nicht mit der SPS Verbunden</li> <li>- Kommunikationsparameter fehlerhaft</li> <li>- Verwendung von CoLa Kommandos, die keine Antwort (Echo) zurücksenden</li> <li>- Bearbeitungszeit des Kommandos &gt; Timeout Zeit</li> </ul>
W#16#0006	Ungültige Kommandolänge	Die Länge des zu sendenden Kommandos ist länger als die angegebene Kommandolänge (COMMAND_LEN).
W#16#0007	SFC20 Fehler	Der Baustein SFC20 (BLKMOV) meldet einen Bausteinfehler. Der Fehlercode wird in der Variable „nStatusBLKMOV“ des instanziierten Datenbausteins angezeigt.  Zur Interpretation des Fehlercodes verwenden Sie bitte das Step7 Hilfesystem.
W#16#000A	Empfangendes Telegramm > #RECORD Länge	Das empfangende Telegramm ist länger als die angegebene #RECORD Länge.
W#16#000B	Ungültiges Input Modul	Die angegebene Länge des Input Moduls (IN_LEN) ist ungültig.  Gültiger Wertebereich: [8..128]
W#16#000C	Ungültiges Output Modul	Die angegebene Länge des Output Moduls (OUT_LEN) ist ungültig.  Gültiger Wertebereich: [8..128]
W#16#000D	Interner Bausteinfehler	Interner Bausteinfehler
W#16#8XXX	SFC14 / SFC15 Fehler	Der Baustein SFC14 (DPRD_DAT) / SFC15 (DPWR_DAT) meldet einen Bausteinfehler.  Zur Interpretation des Fehlercodes verwenden Sie bitte das Step7 Hilfesystem.

## 5 Beispiel

Abbildung 4 zeigt eine Beispielbeschaltung des CCOM\_PNDP FBs. In der Hardwarekonfiguration ist ein SICK Gerät mit einer Prozessdatenbreite von 20 Byte Input und 8 Byte Output projektiert. Die gewählten E/A-Module sind jeweils auf die Anfangsadresse 256 (W#16#100) adressiert (siehe *Abbildung 2*).

### Programmaufruf:

```
OB1 : "Main Program Sweep (Cycle)"
```

Bitte beachten:

Es werden nur S7-300/S7-400 Steuerungen mit integrierter Profinet/Profibus Schnittstelle unterstützt

---

Please note:

This function block may only be operated with S7-300/S7-400 controllers with integrated Profinet/Profibus interfaces

**Netzwerk 1:** AUFRUF CCOM\_PNDP FB | CALL CCOM\_PNDP FB

Aufruf des CoLa Funktionsbausteins

---

Call of the CoLa function block

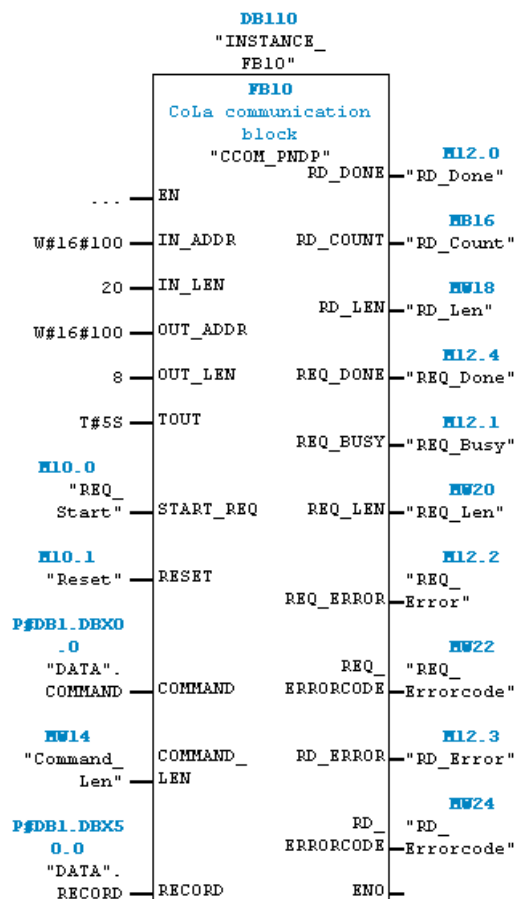




Abbildung 4: Aufruf des CCOM\_PNDP FBs im OB1

**Variablentabelle zum ausführen eines CoLa Kommandos:**

// CCOM PNDP integrated function block					
M	10.0	"REQ_Start"	BOOL	 true	
M	10.1	"Reset"	BOOL	false	
// Reading Result Status					
M	12.0	"RD_Done"	BOOL	false	
MB	16	"RD_Count"	DEZ	101	
M	12.3	"RD_Error"	BOOL	false	
MW	24	"RD_Errorcode"	HEX	W#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	W#16#0000	
// Command					
MW	14	"Command_Len"	DEZ	13	13
DB1.DBB	0	"DATA".COMMAND[1]	ZEICHEN	's'	's'
DB1.DBB	1	"DATA".COMMAND[2]	ZEICHEN	'M'	'M'
DB1.DBB	2	"DATA".COMMAND[3]	ZEICHEN	'N'	'N'
DB1.DBB	3	"DATA".COMMAND[4]	ZEICHEN	' '	' '
DB1.DBB	4	"DATA".COMMAND[5]	ZEICHEN	'm'	'm'
DB1.DBB	5	"DATA".COMMAND[6]	ZEICHEN	'T'	'T'
DB1.DBB	6	"DATA".COMMAND[7]	ZEICHEN	'C'	'C'
DB1.DBB	7	"DATA".COMMAND[8]	ZEICHEN	'g'	'g'
DB1.DBB	8	"DATA".COMMAND[9]	ZEICHEN	'a'	'a'
DB1.DBB	9	"DATA".COMMAND[10]	ZEICHEN	't'	't'
DB1.DBB	10	"DATA".COMMAND[11]	ZEICHEN	'e'	'e'
DB1.DBB	11	"DATA".COMMAND[12]	ZEICHEN	'o'	'o'
DB1.DBB	12	"DATA".COMMAND[13]	ZEICHEN	'n'	'n'
DB1.DBB	13	"DATA".COMMAND[14]	ZEICHEN	B#16#00	
DB1.DBB	14	"DATA".COMMAND[15]	HEX	B#16#00	
DB1.DBB	15	"DATA".COMMAND[16]	ZEICHEN	B#16#00	
DB1.DBB	16	"DATA".COMMAND[17]	ZEICHEN	B#16#00	
DB1.DBB	17	"DATA".COMMAND[18]	ZEICHEN	B#16#00	
DB1.DBB	18	"DATA".COMMAND[19]	ZEICHEN	B#16#00	
DB1.DBB	19	"DATA".COMMAND[20]	ZEICHEN	B#16#00	

Das CoLa Kommando (hier: 'sMN mTCgateon') wird ausgeführt, sobald das Bit „REQ\_START“ mit einer positiven Flanke angesteuert wird. Dem Parameter Command\_Len wird die Länge des Kommandos übergeben (hier: 13 Zeichen).

**Variablentabelle für eingehende Kommandoantworten:**

// Record

MV	18	"RD_Len"	DEZ	31
MV	20	"REQ_Len"	DEZ	15
DB1.DBB	100	"DATA".RECORD[1]	ZEICHEN	's'
DB1.DBB	101	"DATA".RECORD[2]	ZEICHEN	'A'
DB1.DBB	102	"DATA".RECORD[3]	ZEICHEN	'N'
DB1.DBB	103	"DATA".RECORD[4]	ZEICHEN	' '
DB1.DBB	104	"DATA".RECORD[5]	ZEICHEN	'm'
DB1.DBB	105	"DATA".RECORD[6]	ZEICHEN	't'
DB1.DBB	106	"DATA".RECORD[7]	ZEICHEN	'c'
DB1.DBB	107	"DATA".RECORD[8]	ZEICHEN	'g'
DB1.DBB	108	"DATA".RECORD[9]	ZEICHEN	'a'
DB1.DBB	109	"DATA".RECORD[10]	ZEICHEN	't'
DB1.DBB	110	"DATA".RECORD[11]	ZEICHEN	'e'
DB1.DBB	111	"DATA".RECORD[12]	ZEICHEN	'o'
DB1.DBB	112	"DATA".RECORD[13]	ZEICHEN	'n'
DB1.DBB	113	"DATA".RECORD[14]	ZEICHEN	' '
DB1.DBB	114	"DATA".RECORD[15]	ZEICHEN	'l'
DB1.DBB	115	"DATA".RECORD[16]	ZEICHEN	B#16#00
DB1.DBB	116	"DATA".RECORD[17]	ZEICHEN	B#16#00
DB1.DBB	117	"DATA".RECORD[18]	ZEICHEN	B#16#00
DB1.DBB	118	"DATA".RECORD[19]	ZEICHEN	B#16#00
DB1.DBB	119	"DATA".RECORD[20]	ZEICHEN	B#16#00

Die Kommandoantwort (REQ) auf ein gesendete Kommando (hier: 'sAN mTCgateon 1') steht im Record Bereich zur Verfügung, sobald 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

MW	18	"RD_Len"	DEZ	9
MW	20	"REQ_Len"	DEZ	15
DB1.DBB	100	"DATA".RECORD[1]	ZEICHEN	'1'
DB1.DBB	101	"DATA".RECORD[2]	ZEICHEN	'2'
DB1.DBB	102	"DATA".RECORD[3]	ZEICHEN	'3'
DB1.DBB	103	"DATA".RECORD[4]	ZEICHEN	'4'
DB1.DBB	104	"DATA".RECORD[5]	ZEICHEN	'5'
DB1.DBB	105	"DATA".RECORD[6]	ZEICHEN	'6'
DB1.DBB	106	"DATA".RECORD[7]	ZEICHEN	'7'
DB1.DBB	107	"DATA".RECORD[8]	ZEICHEN	'8'
DB1.DBB	108	"DATA".RECORD[9]	ZEICHEN	'9'
DB1.DBB	109	"DATA".RECORD[10]	ZEICHEN	'1'
DB1.DBB	110	"DATA".RECORD[11]	ZEICHEN	'e'
DB1.DBB	111	"DATA".RECORD[12]	ZEICHEN	'o'
DB1.DBB	112	"DATA".RECORD[13]	ZEICHEN	'n'
DB1.DBB	113	"DATA".RECORD[14]	ZEICHEN	' '
DB1.DBB	114	"DATA".RECORD[15]	ZEICHEN	'1'
DB1.DBB	115	"DATA".RECORD[16]	ZEICHEN	B#16#00
DB1.DBB	116	"DATA".RECORD[17]	ZEICHEN	B#16#00
DB1.DBB	117	"DATA".RECORD[18]	ZEICHEN	B#16#00
DB1.DBB	118	"DATA".RECORD[19]	ZEICHEN	B#16#00
DB1.DBB	119	"DATA".RECORD[20]	ZEICHEN	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.