

SICK **CoLa Kommunikationsbaustein**

SICK_CCOM_EIP Add-On Instruction für
Rockwell Automation Steuerungen



Versionshistorie

Version	Datum	Beschreibung
V1.0	14.08.2012	Initiale Version
V1.1	07.11.2013	Fix Arithmetik Error alle 32768 eingehende Leseergebnisse
V1.2	26.11.2015	Fix Arithmetik Error alle 128 eingehende Telegramme

Inhaltsverzeichnis

1 Zu diesem Dokument	3
1.1 Funktion dieses Dokuments	3
1.2 Zielgruppe	3
2 Allgemeines	4
2.1 Bausteinspezifikationen	4
3 Einbinden der AOI in RSLogix5000	5
3.1 SOPAS Gerätekonfiguration	5
3.2 Hardwarekonfiguration	6
3.3 AOI Import	8
3.4 Arbeitsweise	8
3.4.1 Empfangen von Leseergebnissen (Read)	9
3.4.2 Gerätekommunikation über CoLa Kommandos (Req)	9
3.4.3 Verhalten im Fehlerfall	9
4 Parameter	10
5 Fehlercodes	13
6 Beispiele	14
6.1 Auslesen des Leseergebnisses (Read only)	14
6.2 Senden eines CoLa Kommandos	17

1 Zu diesem Dokument

Bitte lesen Sie dieses Kapitel sorgfältig, bevor Sie mit dieser Betriebsanleitung und der SICK_CCOM_EIP Add-On Instruction arbeiten.

1.1 Funktion dieses Dokuments

Diese Betriebsanleitung beschreibt den Umgang mit der SICK_CCOM_EIP Add-On Instruction. Sie leitet das technische Personal des Maschinenherstellers bzw. Maschinenbetreibers zur Projektierung und Inbetriebnahme der Add-On Instruction an.

1.2 Zielgruppe

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

2 Allgemeines

Die Add-On Instruction (AOI) wird zur Kommunikation zwischen einer Rockwell Steuerung und einem SICK Autolident EtherNet/IP Sensor verwendet. Der Sensor muss hierfür in das EtherNet/IP Umfeld der Steuerung eingebunden werden. Die Kommunikation erfolgt zyklisch über die Prozessdaten (implizite Kommunikation).

Die AOI kann für die folgenden SICK Geräte verwendet werden:

- Lector62x
- RFU63x
- CLV69x
- RFH62x (unterstützt nur das Empfangen von Leseergebnissen)
- CLV62x / 65x (unterstützt nur das Empfangen von Leseergebnissen)

Die AOI unterstützt bei den folgenden Funktionen:

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

2.1 Bausteinspezifikationen

AOI Name:	SICK_CCOM_EIP
Version:	1.2
Routinenname:	Logic
Verwendete UDTs:	keine
Bausteinanruf:	Zyklisch
Erstellsprache:	Strukturierter Text (ST)

Die SICK_CCOM_EIP AOI ist eine asynchron arbeitende Routine d.h. die Bearbeitung erstreckt sich über mehrere SPS Zyklen. Die folgende Abbildung zeigt die Darstellung der AOI in der Function Block Diagram (FBD) Ansicht.



Abbildung 1: Darstellung der SICK_CCOM_EIP AOI

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

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

3 Einbinden der AOI in RSLogix5000

Die AOI kann mit allen Rockwell Steuerungen verwendet werden, die RSLogix5000 V16 oder höher einsetzen.

Die Implementierung des SICK_CCOM_EIP Bausteins wird über eine Add-On Instruction (AOI) gehandhabt. Die AOI beinhaltet eine Programmroutine, die an einer beliebigen Stelle im Anwenderprogramm zyklisch aufgerufen werden muss.

3.1 SOPAS Gerätekonfiguration

Um den EtherNet/IP Bus im SICK Gerät zu aktivieren, muss in SOPAS-ET unter dem Menüpunkt **Network / Interfaces / IOs** → **Ethernet** → **EtherNet/IP** die folgenden Einstellungen vorgenommen werden:

- EtherNet/IP enabled: Aktivieren
- Communication Mode: with Handshake
- Assembly Output Size at PLC: 10..500
- Assembly Input Size at PLC: 10..500

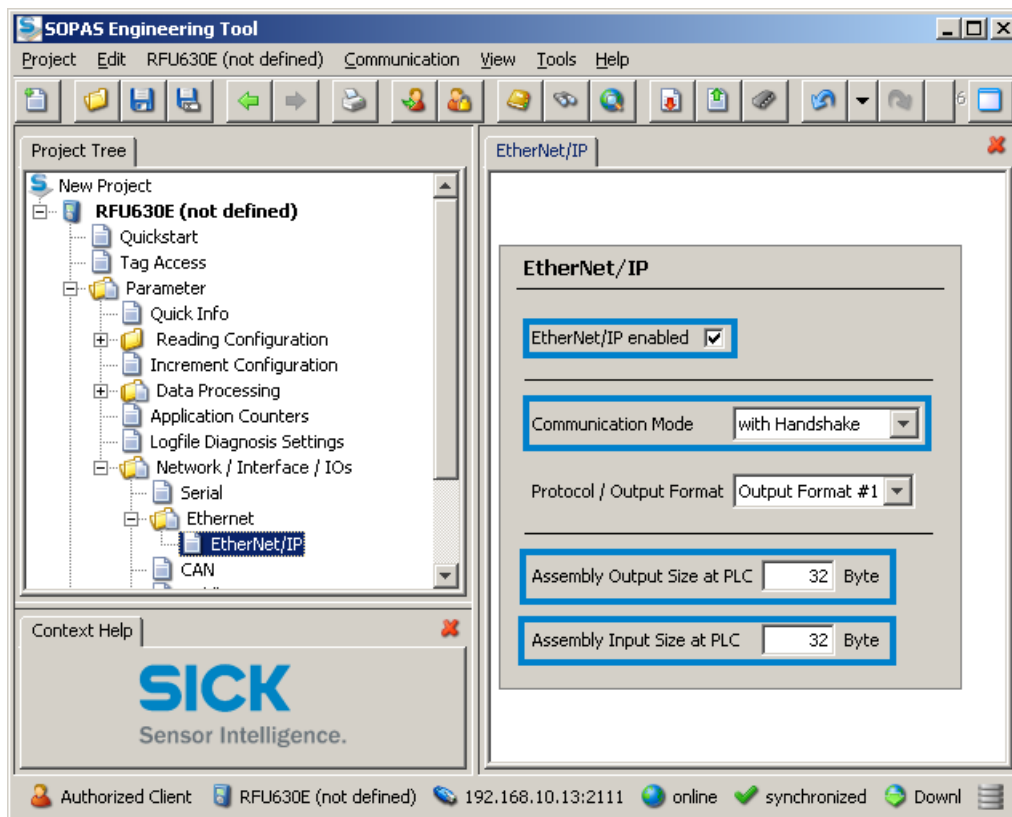


Abbildung 2: Aktivieren der EtherNet/IP Kommunikation in SOPAS-ET

Die AOI kommuniziert über Prozessdaten mit dem SICK Sensor (Implizite EtherNet/IP Kommunikation). Die Input-Assembly und die Output-Assembly beinhalten die Prozessdaten des Sensors. Die Länge der Assemblies gibt an, wie viele Daten in einem Buszyklus übertragen werden können. Die AOI unterstützt Prozessdatenlängen von bis zu 500 Bytes. Ist der Telegraminhalt länger als die Länge der Prozessdaten, wird das Telegram fragmentiert übertragen.

3.2 Hardwarekonfiguration

Um mit RSLogix5000 auf die Input- / Output- Assemblies zugreifen zu können, muss zunächst der verwendete Sensor in projiziert werden.

Klicken Sie mit der rechten Maustaste auf das Symbol **Ethernet** und wählen Sie die Auswahl **New Module....**

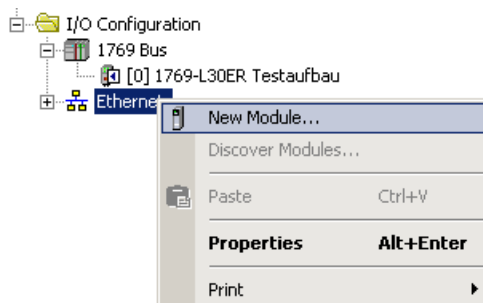


Abbildung 3: Neues Ethernetmodul in RSLogix5000 einfügen

Wählen Sie im Dialog **Select Module** das Modul **ETHERNET-MODULE (Generic Ethernet Module)** aus und klicken Sie anschließend auf **Create** um das Modul in die Hardwarekonfiguration aufzunehmen.

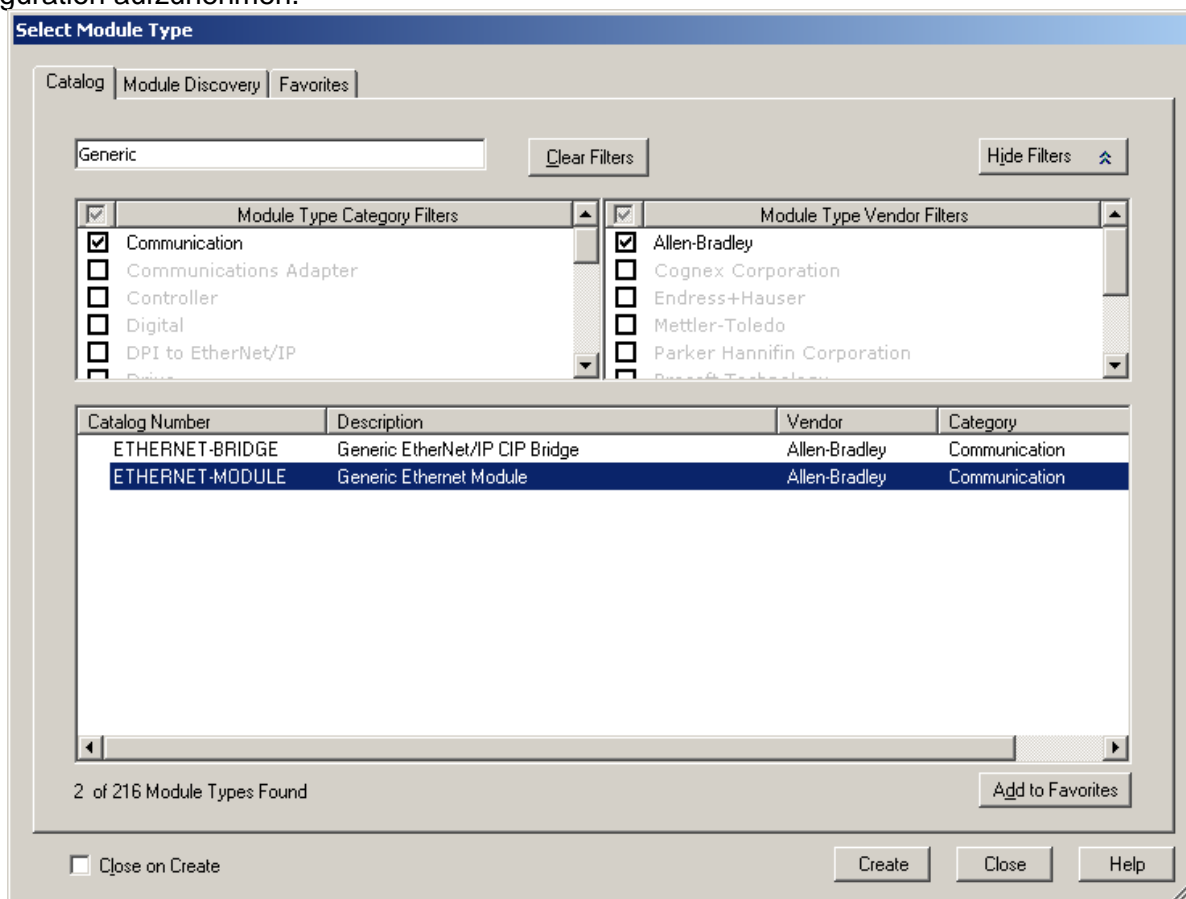


Abbildung 4: Auswahl des Generic Modules in RSLogix5000

Geben Sie im Dialog **New Module** die Einstellungen für **Input**, **Output**, sowie **Configuration** ein.

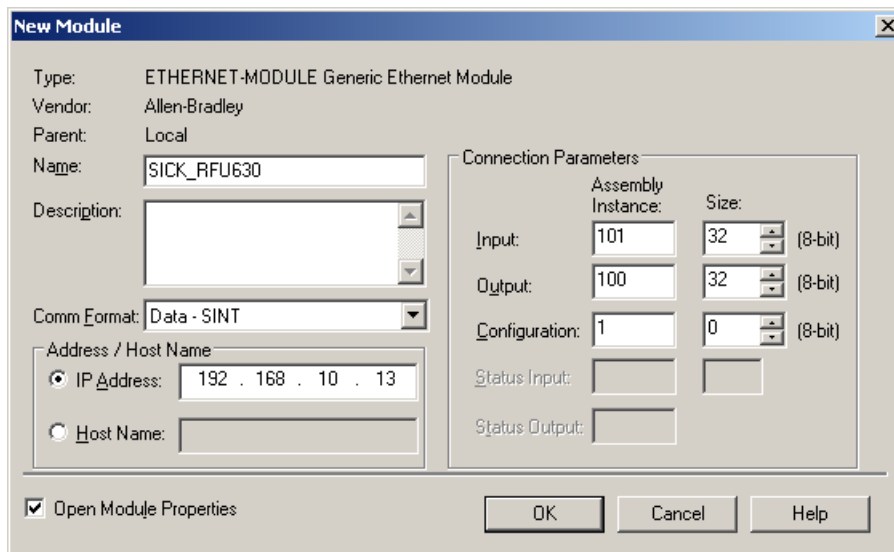


Abbildung 5: Assembly Einstellungen des SICK Sensors

Beispiel:

- **Name:** SICK_RFU630 (Name ist frei wählbar)
- **Comm Format:** Data – SINT
- **IP Address:** 192.168.10.13 (IP-Adresse des SICK Sensors)
- **Input Assembly Instance:** 101
- **Input Assembly Size:** 32 (Die Size muss mit der in SOPAS konfigurier-ten Datenlänge identisch sein siehe Kapitel 3.1)
- **Output Assembly Instance:** 100
- **Input Assembly Size:** 32 (Die Size muss mit der in SOPAS konfigurier-ten Datenlänge identisch sein siehe Kapitel 3.1)
- **Configuration Assembly Instance:** 1
- **Configuration Assembly Size:** 0

Laden Sie die Konfiguration wie folgt in die Steuerung.

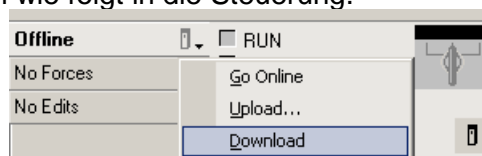


Abbildung 6: Download der SPS Konfiguration

Die Statusanzeigen (Run Mode, Controller OK und I/O) signalisieren, ob die Verbindung zum Sensor erfolgreich hergestellt wurde.

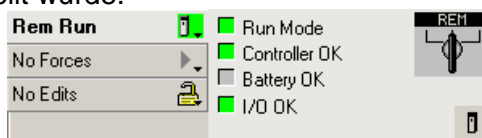


Abbildung 7: Kontrolle der Kommunikation

3.3 AOI Import

Um die SICK_CCOM_EIP AOI im Anwenderprogramm benutzen zu können, muss diese zunächst über **File → Import Component → Add-On Instruction...** in ein bestehendes Projekt importiert werden.

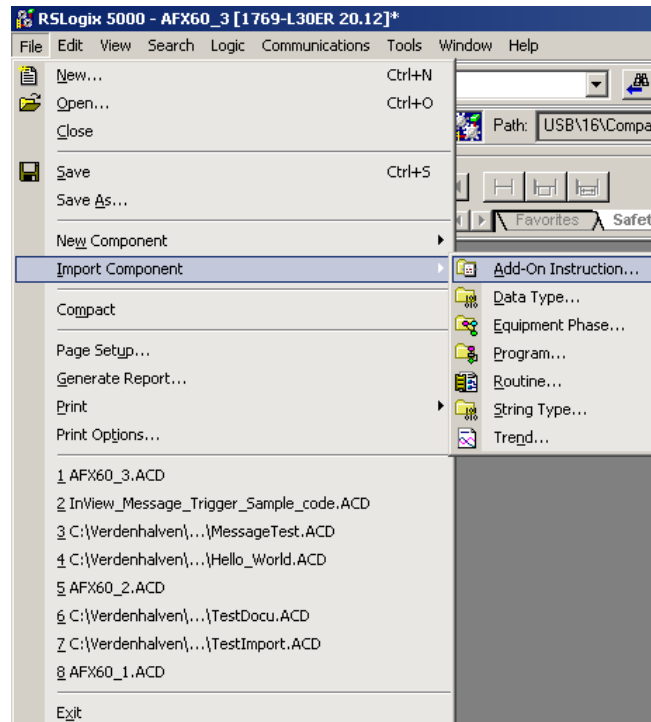


Abbildung 8: Import der SICK_CCOM_EIP Add-On Instruction

3.4 Arbeitsweise

Um den SICK_CCOM_EIP Baustein einsetzen zu können, müssen zunächst die folgenden Bausteinparameter beschaltet werden:

arrInputAssembly: Verweist auf das Input Assembly Array, welches automatisch bei der Geräteprojektierung in den Controller Tags angelegt wird.

arrOutputAssembly: Verweist auf das Output Assembly Array, welches automatisch bei der Geräteprojektierung in den Controller Tags angelegt wird.

arrRecord: Verweist auf das Record-Array. Hier werden die vom Gerät gesendeten Telegramme abgelegt. Das Array vom Datentyp SINT muss im Anwenderprogramm erstellt und der AOI übergeben werden. Die Länge des Array ist abhängig von der maximal zu erwartenden Datenlänge. Werden z.B. Daten mit einer maximalen Länge von 200 Bytes empfangen, muss das Record-Array mindestens mit einer Länge von 200 Bytes angelegt werden.

arrCommand: Verweist auf das Kommando-Array, welches das zu übertragende CoLa Kommando enthält.

iCommandLength: Zeichenlänge des zu übertragenden CoLa Kommandos (z.B. 'sRIO' = Länge 4)

3.4.1 Empfangen von Leseergebnissen (Read)

Geräteseitig gesendete Telegramme (Leseergebnisse) werden in das Array (arrRecord) abgelegt, sobald die AOI neue Daten empfangen hat. Das Bit bReadDone = TRUE zeigt für einen SPS Zyklus den Empfang neuer Daten an. Sobald neue Daten empfangen wurden, wird der Zähler iReadCount inkrementiert. Die jeweilige Bytelänge des zuletzt empfangenden Telegramms kann dem Parameter iReadLength entnommen werden.

3.4.2 Gerätekommunikation über CoLa Kommandos (Req)

Bei der Kommunikation via CoLa Kommandos, wird das im Array (arrCommand) definierte Kommando zum Gerät übertragen. Die resultierende Antwort wird in das vom Parameter arrRecord definierte Datenfeld abgelegt.

Die Übertragung wird gestartet indem der Parameter bReq mit einer positiven Flanke antriggert wird. Solange noch keine gültige Antwort auf das gesendete CoLa Kommando eingetroffen ist, wird dies über den Parameter bReqBusy signalisiert. Sollte innerhalb der Timeout Zeit (iTimeout) keine Antwort eingetroffen sein, wird die Bearbeitung mit einem Timeout Fehler abgebrochen. Der Ausgangsparameter bReqDone = TRUE zeigt an, dass auf ein CoLa Kommando eine Antwort empfangen wurde.

3.4.3 Verhalten im Fehlerfall

Im Fehlerfall signalisiert das Errorbit (bError), dass ein Fehler aufgetreten ist. In diesem Fall wird über den Parameter iErrorCode ein Fehlercode ausgegeben. Das Errorbit bleibt solange gesetzt, bis ein neuer Auftrag gestartet wird. Wenn der Funktionsbaustein nur zum auslesen von Leseergebnissen verwendet wird (iCommandLength = 0), wird das Errorbit automatisch beim Empfang neuer Daten zurückgesetzt.

4 Parameter

Parameter	Deklara- tion	Daten- typ	Beschreibung
arrInputAssembly	INPUT	SINT[1]	<p>Verweist auf das Input Assembly Array, welches automatisch bei der Geräteprojektierung in den Controller Tags angelegt wird.</p> <p>Beispiel: arrInputAssembly:= MySickDevice:I.Data</p>
arrOutputAssembly	INPUT	SINT[1]	<p>Verweist auf das Output Assembly Array, welches automatisch bei der Geräteprojektierung in den Controller Tags angelegt wird.</p> <p>Beispiel: arrOutputAssembly:= MySickDevice:O.Data</p>
arrRecord	INPUT	SINT[1]	<p>Verweist auf das Record Array. Hier werden die vom Gerät gesendeten Telegramme abgelegt. Die Länge des übergebenen Arrays ist abhängig von der maximal zu erwartenden Datenlänge.</p> <p>Beispiel: Der Sensor sendet maximal 50 Byte Daten (Lesergebnis, oder CoLa Telegramantwort). Der Record muss in diesem Fall Minimum eine Länge von 50 Bytes haben.</p> <p>Der Record Inhalt ist nur dann gültig, sobald bReadDone oder bReqDone eine steigende Flanke signalisiert. Es empfiehlt sich den Inhalt des Records zu kopieren, sobald dieser gültige Daten beinhaltet.</p>
arrCommand	INPUT	SINT[1]	<p>Verweist auf das Kommando-Array, welches das zu übertragende CoLa Kommando enthält.</p>
iTimeout	INPUT	INT	<p>Zeit in [ms], nachdem ein Timeout-Fehler ausgelöst wird.</p> <p>Die Timeout Zeit ist standardmäßig auf fünf Sekunden eingestellt.</p> <p>Bitte beachten Sie, dass einige CoLa Kommandos längere Bearbeitungszeit benötigen (z.B. Speicherkommandos).</p>

Parameter	Deklara- tion	Daten- typ	Beschreibung
arrControl	INPUT	SINT[3]	<p>Control Array zum triggern des Sensors über den Feldbus.</p> <p>arrControl[0] = Control Byte 1 arrControl[1] = Control Byte 2 arrControl[2] = Status Byte des CM-Protokolls</p> <p>Beispiel: Zum Triggern des Sensors über den Feldbus, muss das Bit arrControl[0].0 gesetzt werden. Voraussetzung hierfür ist, dass die Triggerquelle in SOPAS auf „Feldbustrigger“ eingestellt ist.</p> <p>Die Definition der weiteren Control-Bits entnehmen Sie bitte der Betriebsanleitung des jeweiligen Sensors.</p>
bReq	INPUT	BOOL	Positive Flanke: Sendet das durch die Parameter arrCommand und iCommandLength definierte CoLa Kommando an den angeschlossenen SICK Sensor und wartet auf die entsprechende Antwort.
iCommandLength	INPUT	INT	<p>Zeichenlänge des zu übertragenden CoLa Kommandos.</p> <p>Beispiel: Wird das Kommando 'sRI0' gesendet, so muss iCommandLength den Wert 4 haben.</p>
bReadDone	OUTPUT	BOOL	<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 wird in das Record-Array geschrieben.</p>
iReadLength	OUTPUT	INT	Zeigt die Bytelänge des empfangenden Leseergebnisses an.
iReadCount	OUTPUT	INT	Zählt die Anzahl der Leseergebnisse die empfangen werden. Der Zähler zählt von 0...32767 (Dez.). Bei einem Überlauf fängt der Zähler wieder bei null an.
bReqDone	OUTPUT	BOOL	<p>Positive Flanke: Es wurde eine Antwort auf ein gesendetes CoLa Kommando empfangen.</p> <p>TRUE: Bearbeitung abgeschlossen FALSE: Bearbeitung noch nicht abgeschlossen</p> <p>Die Kommandoantwort wird in das Record-Array geschrieben.</p>

Parameter	Deklara- tion	Daten- typ	Beschreibung
bReqBusy	OUTPUT	BOOL	Zeigt an, ob ein Request-Auftrag in Bearbeitung ist. FALSE: Kein Auftrag in Bearbeitung TRUE: Auftrag in Bearbeitung
iReqLength	OUTPUT	INT	Gibt die Bytelänge der empfangenden CoLa Telegramantwort an.
bError	OUTPUT	BOOL	Fehler Status: FALSE: Kein Fehler TRUE: Abbruch mit Fehler
iErrorCode	OUTPUT	DINT	Fehlerstatus (siehe Fehlercodes).

5 Fehlercodes

Der Parameter iErrorcode enthält die folgenden Fehlerinformationen:

Fehlercode	Kurzbeschreibung	Beschreibung
16#0000	Kein Fehler	Kein Fehler
16#0001	Timeout	<p>Der Auftrag konnte innerhalb der gewählten Timeoutzeit nicht ausgeführt werden.</p> <p>Dies könnte folgende Ursache haben:</p> <ul style="list-style-type: none"> - Gerät ist nicht mit der SPS Verbunden - Fehlerhafte Kommunikationsparameter - Verwendung von CoLa Kommandos, die keine Antwort (Echo) zurücksenden - Bearbeitungszeit des Kommandos > Timeout Zeit
16#0002	500 < arrInputAssembly < 10	<p>Ungültige Länge der Input Assembly.</p> <p>Gültiger Wertbereich: [10...500]</p>
16#0003	500 < arrOutputAssembly < 10	<p>Ungültige Länge der Output Assembly.</p> <p>Gültiger Wertbereich: [10...500]</p>
16#0004	iCommandLength <= 0 Byte	Die Länge des zu übertragenden CoLa-Kommandos ist <=0 Bytes lang.
16#0005	arrRecord < 4 Byte	Die Länge des Record-Arrays (arrRecord) ist kleiner 4 Bytes.
16#0006	iCommandLength > arrCommand	Die Kommandolänge (iCommandLength) ist länger als das Command-Array (arrCommand).
16#0007	Internen Fehler	Fragmentierungsfehler. Zugriff auf ungültigen Speicherbereich.
16#0008	Eingehendes CoLa-Telegram > arrRecord	Das eingehende CoLa-Telegram ist länger als das Record-Array (arrRecord).
iRead- Length = -1	Eingehendes Leseergebnis > arrRecord	Das eingehende Leseergebnis ist länger als das Record-Array (arrRecord).

6 Beispiele

6.1 Auslesen des Leseergebnisses (Read only)

Abbildung 10 zeigt, wie man mit der SICK_CCOM_EIP AOI ein Leseergebnis empfangen kann. Dieses Beispiel kann für alle SICK Autolident Sensoren verwendet werden.

Der Sensor (hier CLV620) wird über das Control Byte (arrControl[0].0) direkt getriggert. Der Baustein kann unabhängig der Triggerquelle betrieben werden.

Start/Stop of Object Trigger

Start

Delay ms Fieldbus Input / CAN Open

Stop

Delay ms Trigger source or Good Read or Not defined

Trigger Distribution

Distribute on Disabled

Abbildung 9: Triggereinstellung des CLVs in SOPAS-ET

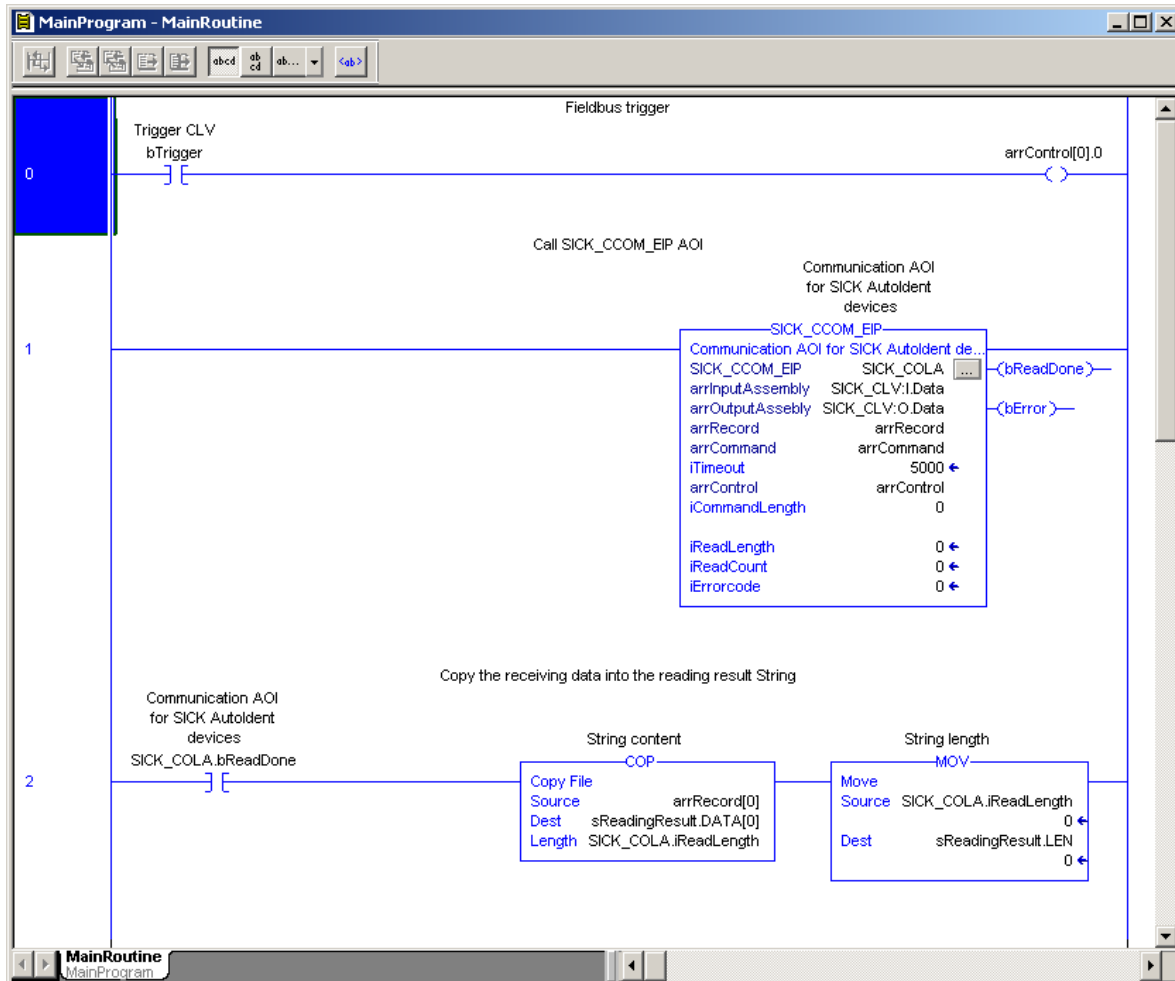


Abbildung 10: Aufruf der AOI im Main Programm der Steuerung

Nachdem die Variable `bReadDone = TRUE` signalisiert, wird der Inhalt des Records (`arrRecord`) in den Result-String (`sReadingResult`) kopiert. Die direkte Verwendung des Records als Leseergebnis ist nicht zu empfehlen, da dieser nur zum Zeitpunkt `bReadDone = TRUE` gültige Werte enthält.

Abbildung 11 zeigt das Leseergebnis des CLVs an. Der Inhalt des Leseergebnisses kann über das SOPAS Ausgabeformat frei konfiguriert werden (siehe Abbildung 12).

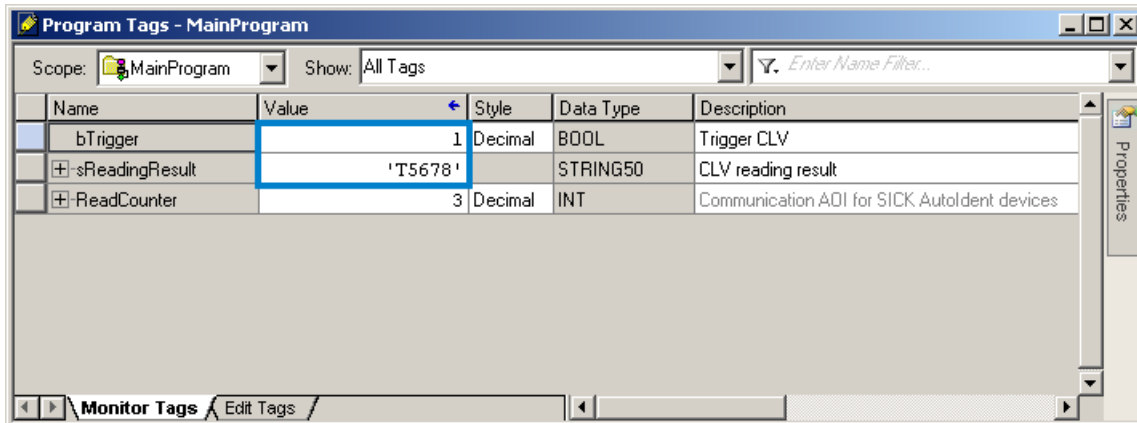


Abbildung 11: Abruf des Leseergebnisses

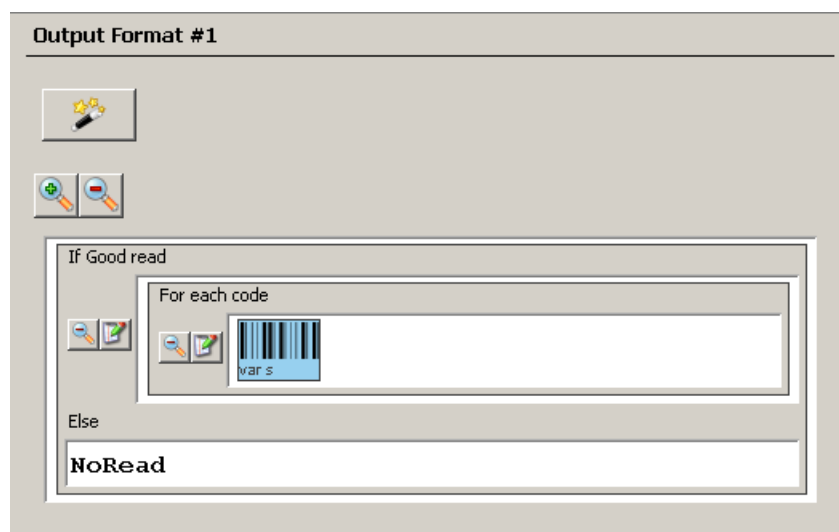


Abbildung 12: Konfiguration Ausgabeformats in SOPAS-ET

6.2 Senden eines CoLa Kommandos

Abbildung 13 zeigt, wie man mit der SICK_CCOM_EIP AOI parallel zum Empfang von Lesergebnissen ein Gerätekommando abschicken kann. Dieses Beispiel kann nur in Verbindung mit den Geräten RFU63X oder Lector62X verwendet werden.

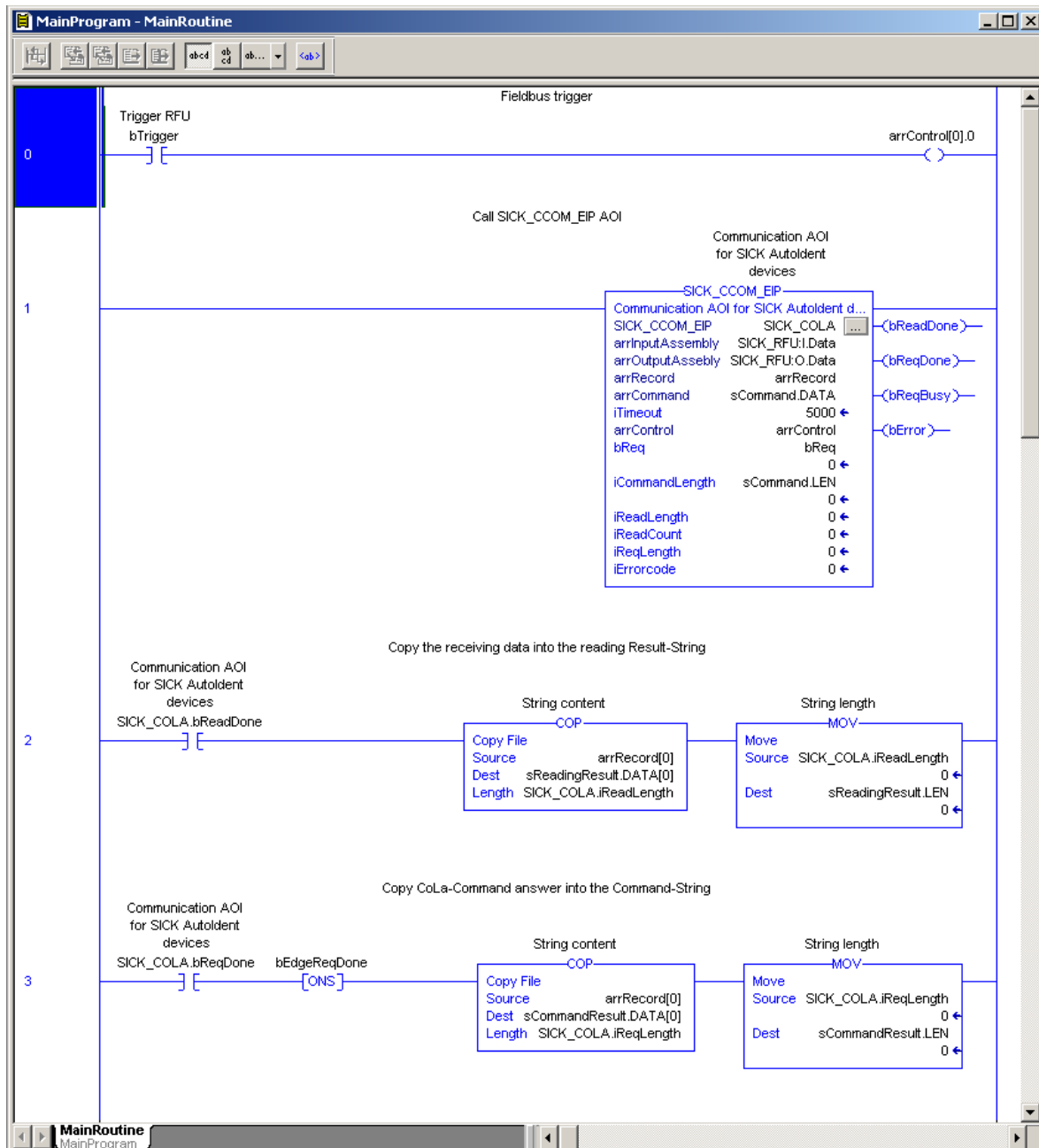


Abbildung 13: Aufruf der AOI im Main Programm der Steuerung

Sobald der Record gültige Daten enthält (bReadDone = TRUE / bReqDone = steigende Flanke), wird der Inhalt des Records in die Variablen sReadingResult oder sCommandResult kopiert.

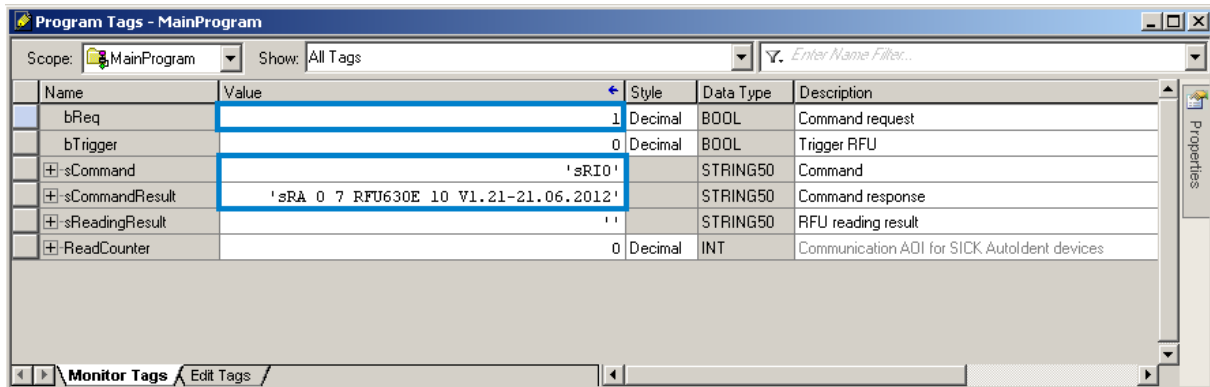


Abbildung 14: RSLogix5000 Tag Ansicht

Abbildung 14 zeigt die, wie man über die AOI ein CoLa-Kommando an ein Gerät (hier RFU630) schicken kann. In diesem Beispiel wird das CoLa-Kommando 'sRI0' (Geräteidentifikation) an den RFU gesendet. Hierfür müssen die folgenden Parameter beschrieben werden:

sCommand: 'sRI0'
iCommandLength: 4 (Kann aus der String-Variable sCommand.LEN entnommen werden)

Das Kommando wird ausgeführt, sobald die Variable bReq mit einer positiven Flanke angeregt wird. Sobald das Ausgangsbit bReqDone eine positive Flanke signalisiert, kann der Record umkopiert und verwendet werden.

Parallel zum senden kann die AOI ebenfalls die vom Gerät gesendeten Leseergebnisse empfangen und anzeigen.