Technical Information

# SICK
# CoLa Communication Function Block
## (Only for CLV62x / 65x and RFH6xx devices)

SICK_CCOM_CLV_RFH_EIP Add-On Instruction fo
Rockwell Automation Controls

SICK
Sensor Intelligence.

# Version history

| Version | Date | Remarks |
|---|---|---|
| V1.0 | 07.12.2012 | Initial version |
| V1.1 | 07.11.2013 | Fix arithmetic error every 32768 incoming reading results |

# Table of Contents

# 1 About this document

Please read this chapter carefully before you start working with this operating instructions and the SICK_CCOM_CLV_RFH_EIP Add-On Instruction.

## 1.1 Function of this document

This operating instructions describes how to use the SICK_CCOM_CLV_RFH_EIP Add-On Instruction. It is used for guiding technical personnel working for the machine manufacturer / operator in project planning and commissioning the Add-On Instruction.

## 1.2 Target group

This operating instructions is aimed for specialists, such as technicians and engineers.

# 2 General information

The Add-On Instruction (AOI) is used for the communication between a Rockwell control and a SICK AutoIdent EtherNet/IP sensor. The sensor has to be embedded into the EtherNet/IP surrounding of the control. The communication is done cyclically via process data (implies communication).

The AOI can be used for the following SICK devices:
- RFH62x RFID interrogators
- CLV62x / 65x barcode reader

The AOI supports the following functions:
- Sending of CoLa[i] commands to a SICK sensor
- Receiving of CoLa answers of a SICK sensor
- Receiving of telegrams sent from the device (can be configures in the SOPAS[ii] output format)

Please note:
If you need a PLC solution for a SICK Lector or a RFU device, please use the SICK_CCOM_EIP AOI.

## 2.1 Specification of the function block

| | |
|---|---|
| AOI Name: | SICK_CCOM_CLV_RFH_EIP |
| Version: | 1.1 |
| Routine name: | Logic |
| Used UDTs: | none |
| Call up: | periodically |
| Programming language: | Structured text (ST) |

The SICK_CCOM_CLV_RFH_EIP AOI is an asynchronous working routine which means that the handling is done over several PLC cycles. The following image shows the AOI in the view of the function block diagram.


Image 1: Diagram of SICK_CCOM_CLV_RFH_EIP AOI

[i] The command language (CoLa) is a SICK internal protocol for the communication with SOPAS devices
[ii] SOPAS-ET is an engineering tool for the configuration of SICK sensors

# 3 Embedding of AOI in RSLogix5000

AOI can be used with all Rockwell controls using RSLogix5000 V16 or higher.

The implementation of SICK_CCOM_CLV_RFH_EIP function block is done via the Add-On Instruction (AOI). The AOI contains a program routine, which has to be called up periodically at any position in the user program.

## 3.1 SOPAS device configuration

In order to activate EtherNet/IP Bus in a SICK device, the following settings have to be activated in SOPAS-ET at the menu point **Network / Interfaces /IOs → Ethernet → EtherNet/IP**:

- EtherNet/IP enabled: activate
- Communication Mode: with Handshake

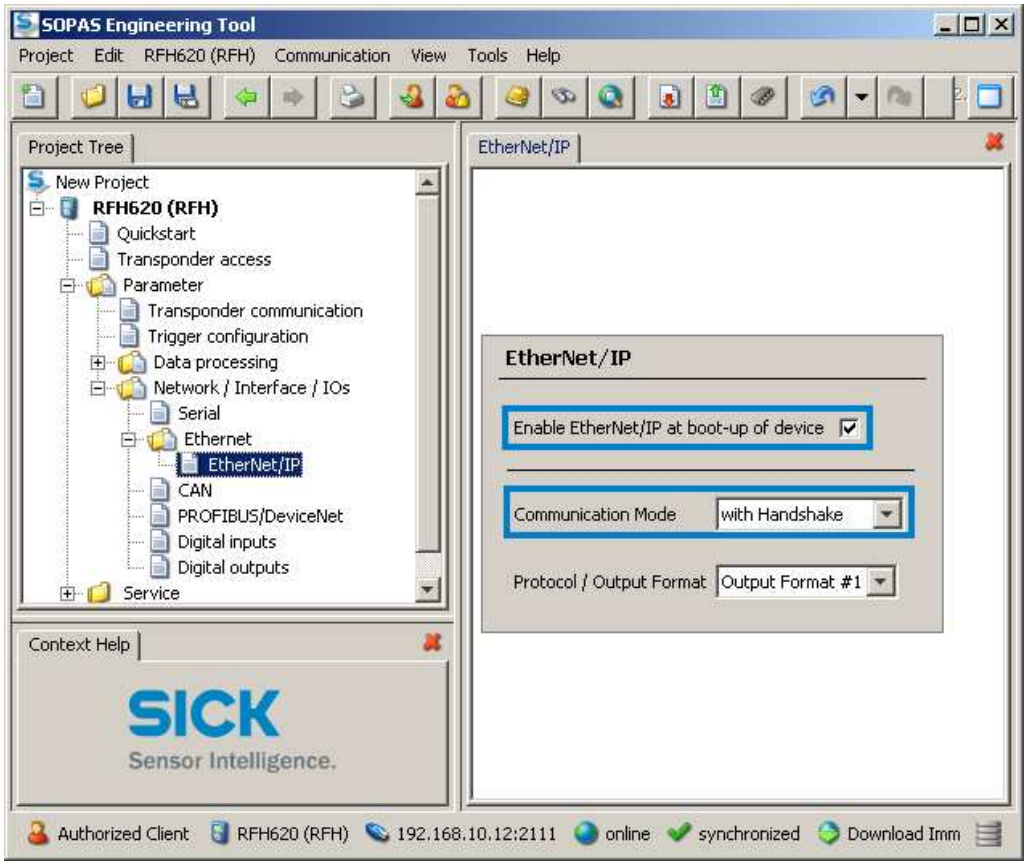


*Image 2: Activating EtherNet/IP communication in SOPAS-ET*

The AOI communicates via process data with the SICK Sensor (implicit EtherNet/IP communication). The Input-Assembly and the Output-Assembly contain the process data of the sensor. The length of the assemblies indicates how much data can be transferred in one bus cycle. At the CLV62x /65x and RFH6xx devices, the assembly sizes are fixed 200 Byte. If the content of the reading result exceed 200 Byte, the telegram will be fragmented automatically. The fragments are reassembled by the AOI.

## 3.2 Hardware configuration

In order to access the input / output assemblies with RSLogix5000, you first have to project the used sensor.

Click with the right mouse button the symbol **Ethernet** and choose the selection **New Module…**.

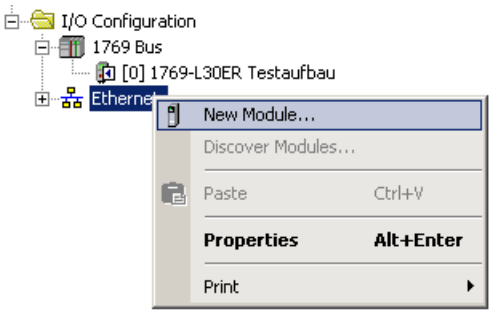


*Image 3: Insert new Ethernet module in RSLogix5000*

Select the module **ETHERNET-MODULE (Generic Ethernet Module)** in the dialogue **Select Module** and then click **Create** in order to add the module to the hardware configuration.

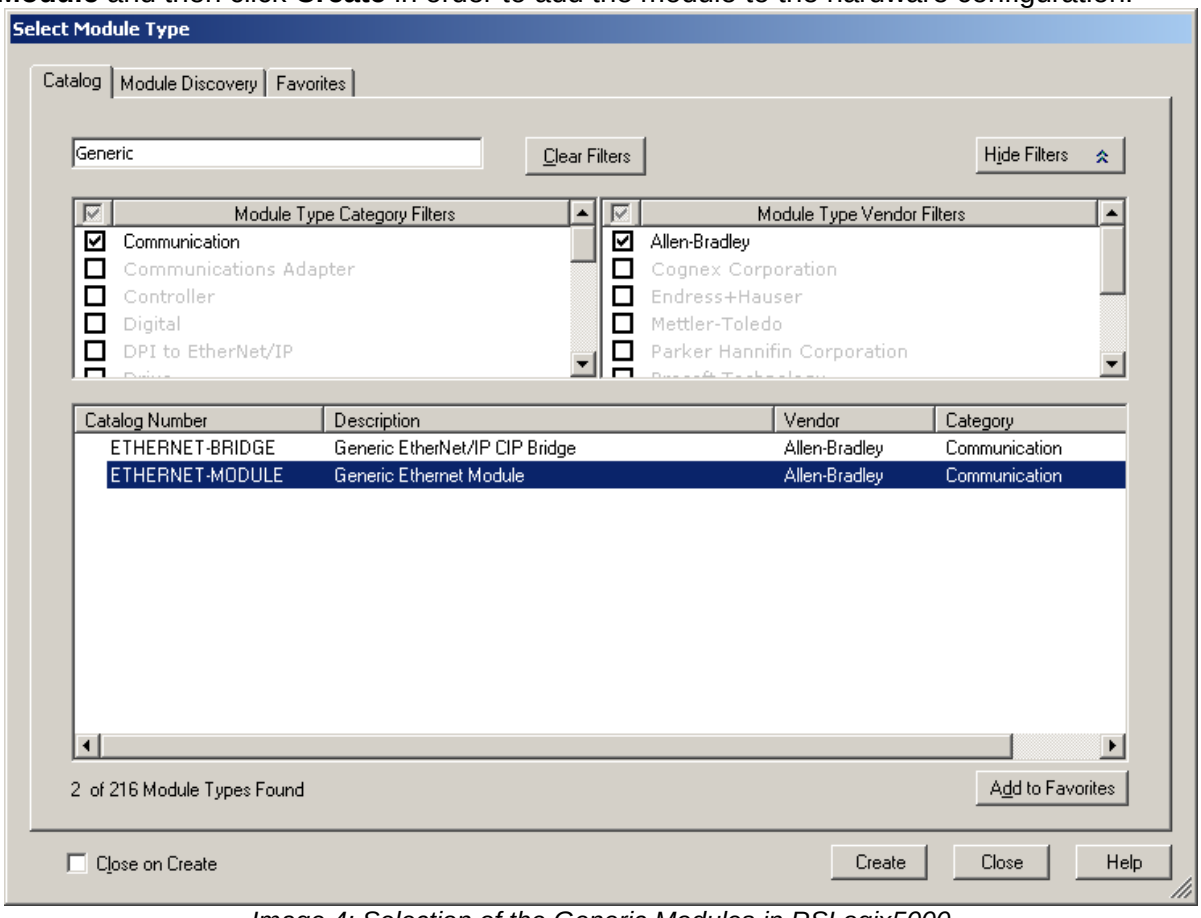


*Image 4: Selection of the Generic Modules in RSLogix5000*

In the dialogue **New Module** please insert the settings for **Input**, **Output**, and **Configuration**.

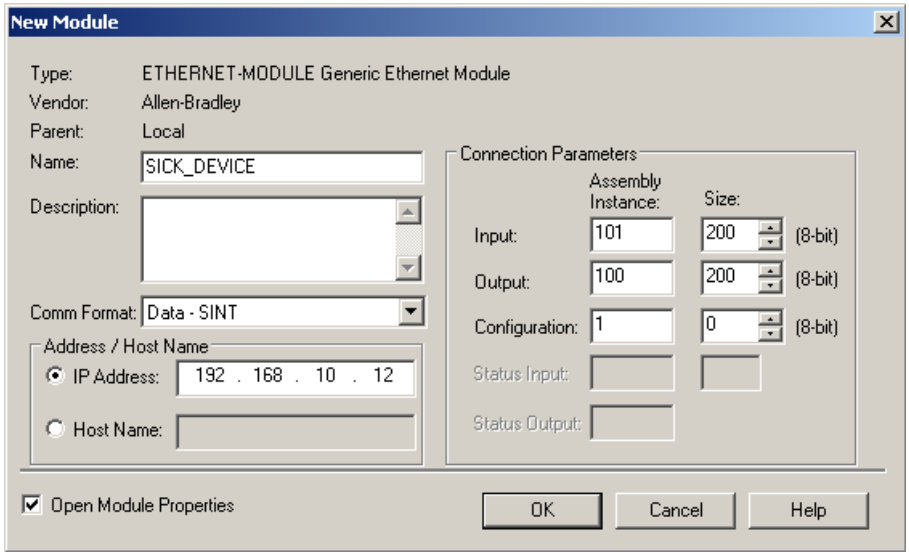


*Image 5: Assembly settings of the SICK sensor*

Example:

- **Name:** SICK_DEVICE (name can be selected arbitrarily)
- **Comm Format:** Data – SINT
- **IP Address:** 192.168.10.12 (IP-Address of the SICK sensor)
- **Input Assembly Instance:** 101
- **Input Assembly Size:** 200
- **Output Assembly Instance:** 100
- **Input Assembly Size:** 200
- **Configuration Assembly Instance:** 1
- **Configuration Assembly Size:** 0

Please load the configuration into the PLC as follows:

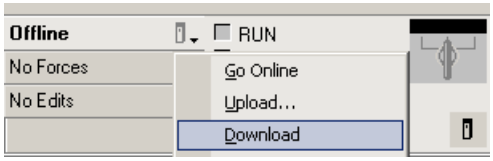


*Image 6: Download of the PLC configuration*

The status display (Run Mode, Controller OK and I/O) signalises if the connection to the sensor has been done successfully.

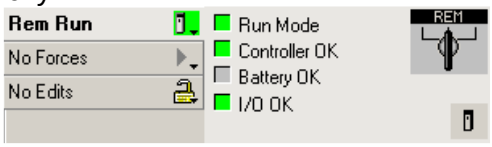


*Image 7: Control of the communication*

## 3.3 AOI Import

In order to use SICK_CCOM_CLV_RFH_EIP AOI in the user program, you first have to import the **File → Import Component → Add-On Instruction…** into an existing project.

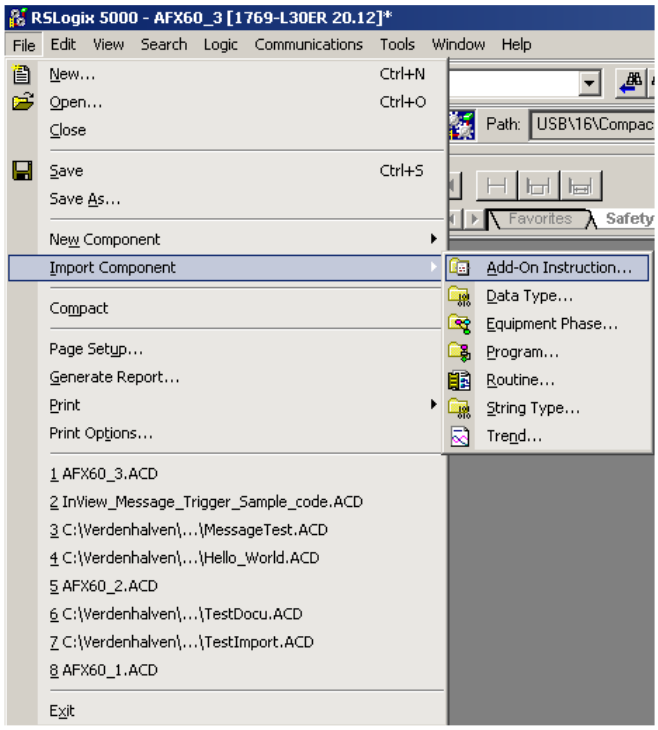


*Image 8: Import of SICK_CCOM_CLV_RFH_EIP Add-On instruction*

## 3.4 Mode of operation

In order to use SICK_CCOM_CLV_RFH_EIP function block, the following function block parameters have to be set first:

arrInputAssembly: Reference to the Input Assembly Array, which is created automatically in the controller tags at the device projection. The Array must have a size of 200 Byte.

arrOutputAssembly: Reference to the Output Assembly Array, which is created automatically in the controller tag at the device projection. The Array must have a size of 200 Byte.

arrRecord: Reference to the Record-Array. Here, the telegrams sent from the device are stored. The array from data type SINT has to be set up in the user program and is then transferred to the AOI. The length of the array depends on the maximal data length. If, e.g. data with a max. length of 200 Bytes is received, the Record-Array has to be set up with a min. length of 200 Bytes.

arrCommand: Reference to the command array which contains the CoLa command that has to be transferred.

iCommandLength: Length of the CoLa command which has to be transferred (e.g. 'sRI0' = length 4)

### 3.4.1 Receipt of read results (Read)

Mutually sent telegrams (read results) are stored in the array (arrRecord), as soon as the AOI has received new data. The Bit bReadDone = TRUE shows the receipt of new data for one PLC cycle. As soon as new data has been received the counter iReadCount will be incremented. The byte length of the last received telegram can be seen in the parameter iReadLength.

### 3.4.2 Device communication via CoLa commands (Req)

At the communication via CoLa commands, the command which is defined in the array (arrCommand) will be transferred to the device. The resulting answer is stored in the data field defined within the parameter arrRecord.

The transfer is started by triggering the parameter bReq with a positive edge. If no valid answer to the sent CoLa command has been received, this is signalised by the parameter bReqBusy. If no answer has been received within the Timeout (iTimeout), the handling will be cancelled with a Timeout error. The output parameter bReqDone = TRUE indicates that an answer to the CoLa command has been received.

### 3.4.3 Behaviour in the case of an error

In the case of an error the Errorbit (bError) signalises the error. In this case an error code will be given out via the parameter iErrorcode. The error bit remains set until a new order has been started. If the function block is only used for the reading of read results (iCommandLength = 0), the error bit will be set back when receiving new data automatically.

# 4 Parameter

| Parameter | Declaration | Type of data | Description |
|---|---|---|---|
| arrInputAssembly | INPUT | SINT[1] | Reference to the Input Assembly Array, which is created automatically in the controller tags during the configuration. The Array must have a size of 200 Byte.<br><br>Example:<br>arrInputAssembly:= SICK_DEVICE:I.Data |
| arrOutputAssembly | INPUT | SINT[1] | Reference to the Output Assembly Array, which is created automatically in the controller tags during the configuration. . The Array must have a size of 200 Byte.<br><br>Example:<br>arrOutputAssembly:= SICK_DEVICE:O.Data |
| arrRecord | INPUT | SINT[1] | Reference to the Record Array. Here, the telegrams sent from the device are stored. The length of the array depends on the max. data length.<br><br>Example:<br>The sensor sends max. 50 Byte data (read result or CoLa telegram answer). In this case the record has to have a minimal length of 50 bytes.<br><br>**The content of the record is only valid if bReadDone or bReqDone signalises an increasing edge. It is recommended to copy the content of the record as soon as it contains valid data.** |
| arrCommand | INPUT | SINT[1] | Reference to the command-Array which contains the CoLa command that has to be transferred. |
| iTimeout | INPUT | INT | Time in [ms], after a Timeout error has occurred.<br><br>By default, the Timeout time is set to five seconds.<br><br>Please note that some CoLa commands need a longer time for processing (e.g. storage commands). |

| Parameter | Declaration | Type of data | Description |
|---|---|---|---|
| arrControl | INPUT | SINT[3] | Control Array for triggering the sensor via field bus.<br><br>arrControl[0] = Control Byte 1<br>arrControl[1] = Control Byte 2<br>arrControl[2] = Status Byte of the CM protocol<br><br>Example:<br>In order to trigger the sensor via field bus, the Bit arrControl[0].0 has to be set. In order to do so, the trigger source in SOPAS has to be set to „fieldbus trigger“.<br><br>The definition of further Control-Bits can be seen in the operating instructions of the respective sensor. |
| bReq | INPUT | BOOL | Positive edge: Sends the through the parameters arrCommand and iCommandLength defined CoLa command to the connected SICK sensor and waits for the corresponding answer. |
| iCommandLength | INPUT | INT | Character length of the CoLa command which has to be transferred.<br><br>Example:<br>If the command ’sRI0’ is sent, the iCommandLength has to have the value 4. |
| bReadDone | OUTPUT | BOOL | Positive edge: A from the device sent read result has been received (formatting can be seen in the SOPAS output format).<br><br>If a read result has been received, the bit is set for always one PLC cycle. The read result is written into the Record-Array. |
| iReadLength | OUTPUT | INT | Shows the byte length of the read result that has to be received. |
| iReadCount | OUTPUT | INT | Counts the number of read results which are received. The counter counts from 0…32767 (Dec.). In the case of an overflow, the counter starts at cero once again. |
| bReqDone | OUTPUT | BOOL | Positive edge: An answer to a sent CoLa command has been received.<br><br>TRUE: Editing is finished<br>FALSE: Editing is not yet finished<br><br>The command answer is written into the record array. |

| Parameter | Declara-tion | Type of data | Description |
|---|---|---|---|
| bReqBusy | OUTPUT | BOOL | Indicates if a request is in progress.<br><br>FALSE: Request is not in progress<br>TRUE: Request is in progress |
| iReqLength | OUTPUT | INT | Shows the byte length of the CoLa telegram answer that has to be received. |
| bError | OUTPUT | BOOL | Error status:<br><br>FALSE: No error<br>TRUE: Interruption with error |
| iErrorcode | OUTPUT | DINT | Error status (see error codes). |

# 5 Error Codes

The parameter iErrorcode contains the following error information:

| Error code | Short description | Description |
|---|---|---|
| 16#0000 | No error | No error |
| 16#0001 | Timeout | The request has not been carried out within the chosen timeout.<br><br>Reasons can be:<br>- Device is not connected with PLC<br>- False communication parameter<br>- Use of CoLa commands that do not send back an answer (echo)<br>- Time for editing the commando > Timeout time |
| 16#0002 | arrInputAssembly <> 200 | Invalid length of the Input Assembly.<br><br>Valid value area:<br>[200] |
| 16#0003 | arrOutputAssembly <> 200 | Invalid length of Output Assembly.<br><br>Valid value area:<br>[200] |
| 16#0004 | 200 < iCommandLength <= 0 Byte | Invalid length of the command length.<br><br>Valid value area:<br>[1..200] |
| 16#0005 | arrRecord < 4 Byte | The length of the Record-Array (arrRecord) is smaller than 4 Bytes. |
| 16#0006 | iCommandLength > arrCommand | The command length (iCommandLength) is longer than the Command-Array (arrCommand). |
| 16#0007 | Internal error | Fragmentation error. Access to invalid storage area. |
| 16#0008 | Incoming CoLa-Telegram > arrRecord | The incoming CoLa-Telegram is longer than the Record-Array (arrRecord). |
| iRead-Length = -1 | Incoming read result > arrRecord | The incoming read result is longer than the Record-Array (arrRecord). |

# 6 Examples

## 6.1 Output of the read results (Read only)

Image 10 shows how to receive a read result with SICK_CCOM_CLV_RFH_EIP AOI. This example can be used for all SICK AutoIdent sensors.

The sensor (in this case CLV620) is directly triggered via the control byte (arrControl[0].0). The function block can be operated independent from the trigger source.

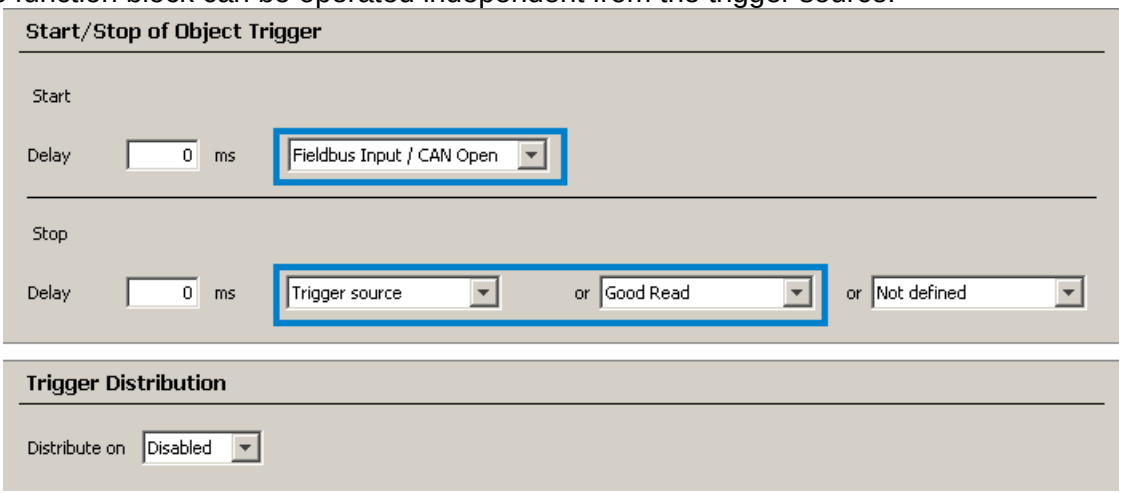


*Image 9: Trigger setting of the CLV in SOPAS-ET*

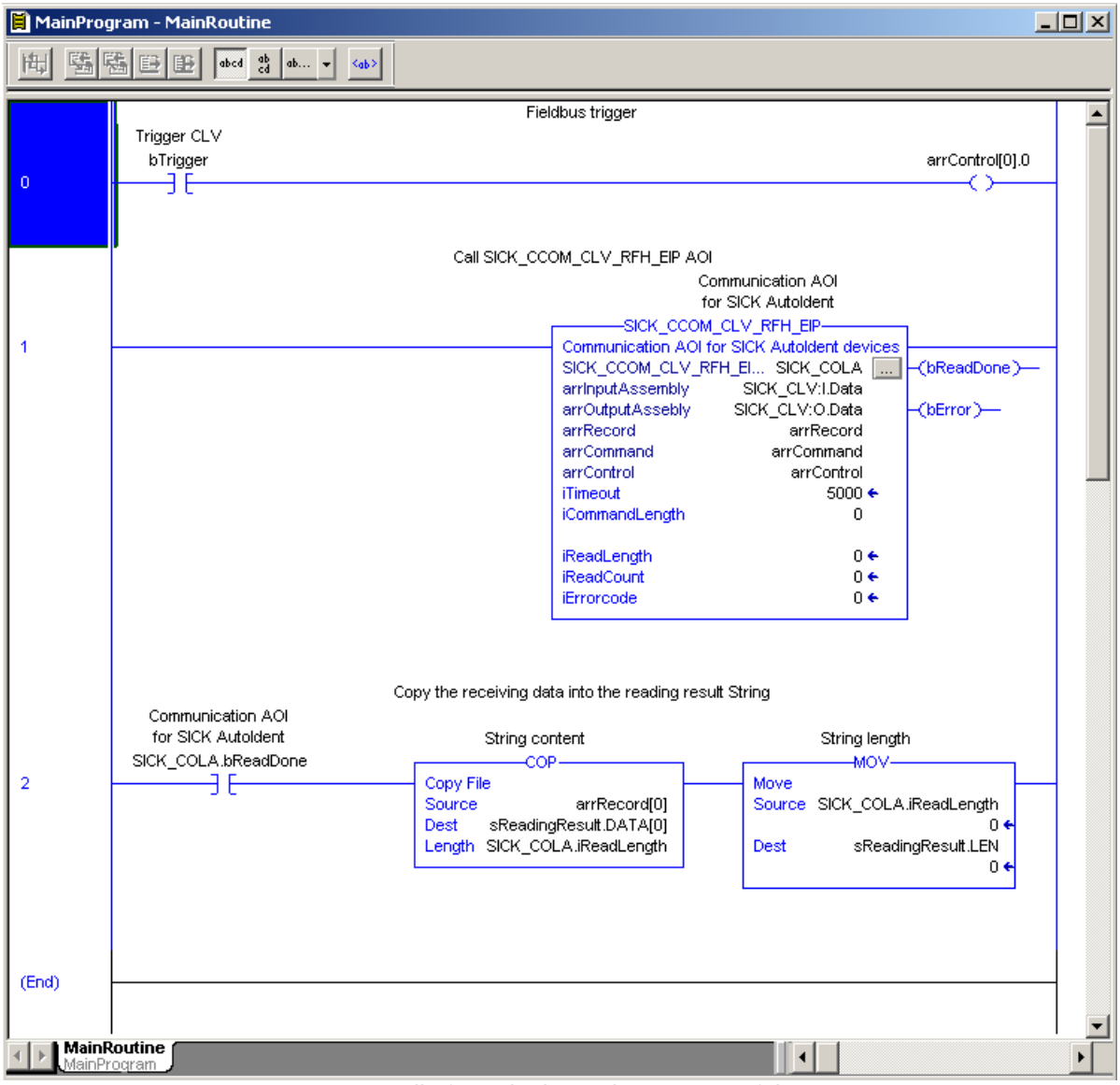


*Image 10: Call of AOI in the Main Program of the PLC*

As soon as the variable bReadDone signalises TRUE, the content of the record (arrRecord) will be copied into the result string (sReadingResult). The direct use of the record as read result is not recommended since it only contains valid values during bReadDone = TRUE.

Image 11 shows the read result of the CLV. The content of the read result can be configured via the SOPAS output format (see Image 12).

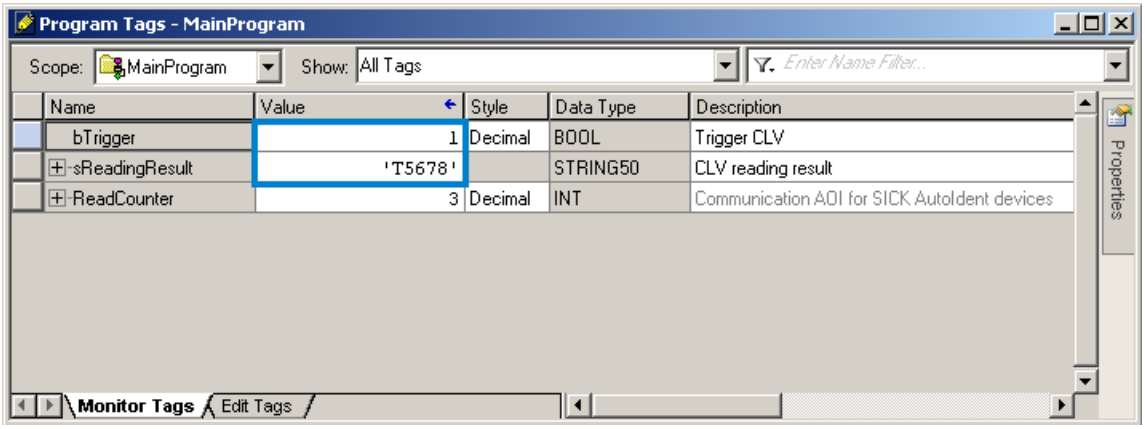


*Image 11: Call of the read result*

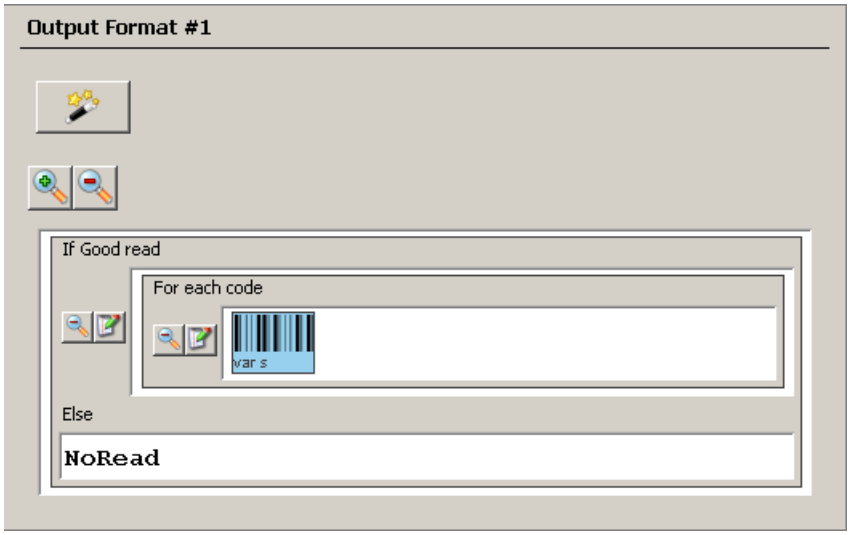


*Image 12: Configuration of the output format in SOPAS-ET*

## 6.2 Sending of a CoLa command

Image 13 shows how you can send a device command with SICK_CCOM_CLV_RFH_EIP AOI parallel to the receipt of read results. This example can only be used in combination with the devices CLV62x / 65x or RFH6xx.

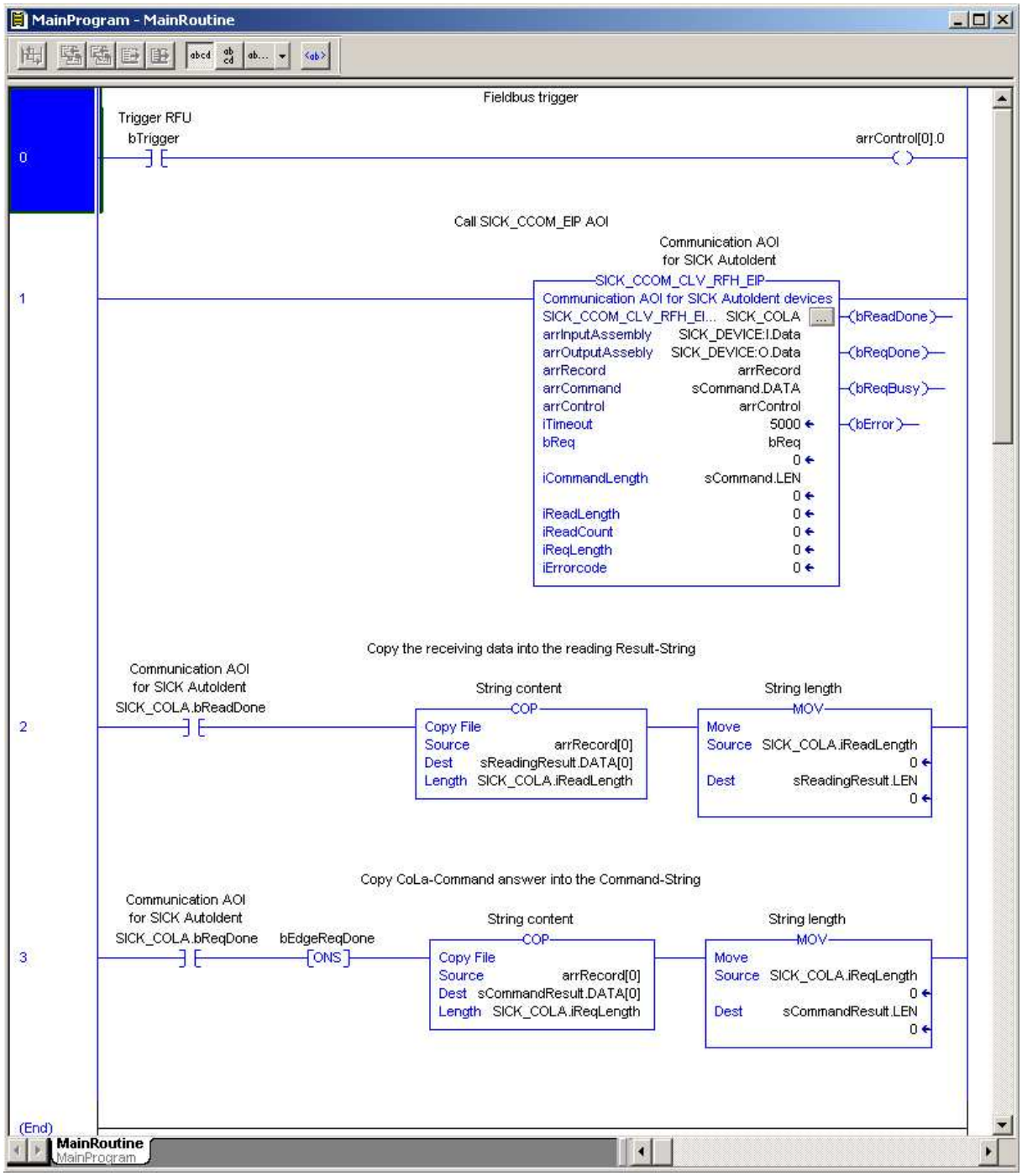


*Image 13: Call up of AOI in the Main Program of the PLC*

As soon as the record contains valid data (bReadDone = TRUE / bReqDone = increasing edge) the content of the records will be copied into the variables sReadingResult or sCommandResult.

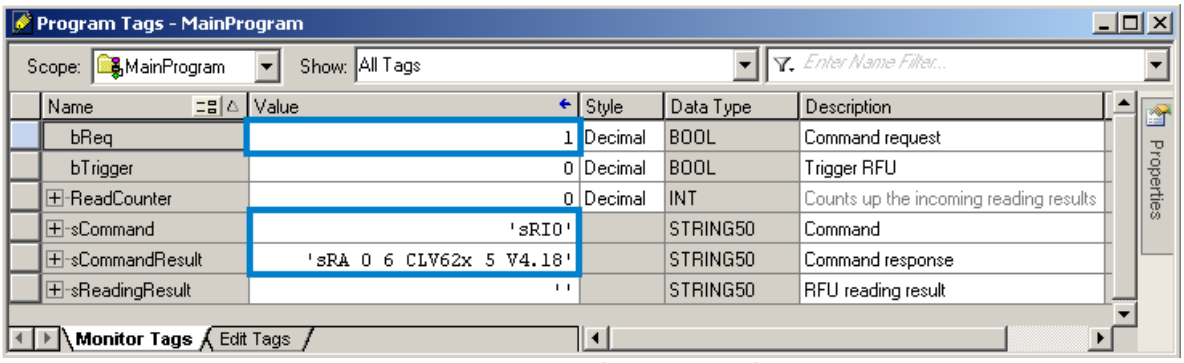


*Image 14: RSLogix5000 Tag view*

Image 14 shows how to send a CoLa command to a device (here CLV620) via AOI. In this example the CoLa command 'sRI0' (device identification) is sent to CLV. In this case the following parameters have to be set:

sCommand: 'sRI0'
iCommandLength: 4 (can be taken from the string variable sCommand.LEN)

The command is carried out as soon as the variable bReq is triggered with a positive edge. As soon as the output bit bReqDone signalises a positive edge, the record can be copied and used.

Parallel to the sending, the AOI can also receive and show read results sent from device.