

SICK	Application Notes		IOLG2
	Date: 14.6.2016	last mod.: 14.06.2016 16:15:00	Proj. Nr.

How to read and write ISDUs with IOLG2 on EIP

To be able to read and write ISDU data using the IOLG2 IO-Link Master in an Ethernet/IP network you need to integrate the master into your Studio 5000 project.

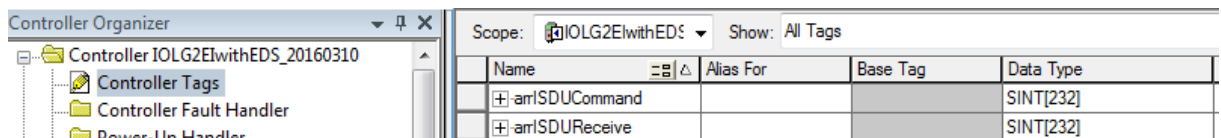
The first steps and the basic configuration is shown in this video tutorial:

www.sick.com/de/de/p/p389646 → Videos

The basic principle of the ISDU communication is shown in chapter 6.3 (page 29) in the user manual. The user manual is found under the same link as above → Downloads → Literature.

Step by step: Using Messages in Locix Designer (Studio 5000):

1. Create two arrays as Controller or Local Tag
 - a. One to receive the data out of an ISDU
 - b. One to send an ISDU command

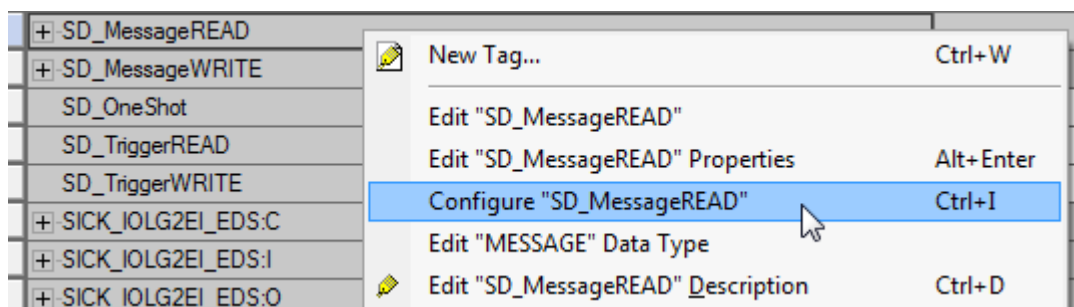


Since the maximum possible size of an ISDU is 232Byte we use SINT arrays of 232 Byte size in this example.

2. Create two messages in your Tag list:

+	SD_MessageREAD		MESSAGE
+	SD_MessageWRITE		MESSAGE

3. Configure the messages as shown in the screenshots:



You find the used values in the user manual.

Attribute 3 = READ
2 = WRITE

Source Element is the Array you created before for sending commands and data.

Source Length is the number of bytes you like to read or write. For reading it doesn't matter if the value is bigger than the ISDU you are reading. For writing it has to match.

Destination Element is the Array you created to receive ISDU data.

Use the button "Browse" to select the Path of the IOLG2 IO-Link Master.

The message for writing needs to be configured in the same way (besides **Attribute** and **Source Length**)

4. The message needs to be called in your source code.
 - a. Using ST coding

```

//Use message boxes in ST
IF SD_TriggerREAD THEN
  MSG(SD_MessageREAD);
ELSIF SD_TriggerWRITE THEN
  MSG(SD_MessageWRITE);
END_IF;
SD_TriggerREAD := 0;
SD_TriggerWRITE := 0;

```

- b. Using Ladder diagram



5. Define the ISDU command
 - a. READ - Example

Name	Value	Force Mask	Style	Data Type	Description
arrISDUCommand	{...}	{...}	Hex	SINT[232]	
arrISDUCommand[0]	16#18		Hex	SINT	ISDU Index low
arrISDUCommand[1]	16#00		Hex	SINT	ISDU Index high
arrISDUCommand[2]	16#00		Hex	SINT	Subindex
arrISDUCommand[3]	16#00		Hex	SINT	Data 0
arrISDUCommand[4]	16#00		Hex	SINT	Data 1
arrISDUCommand[5]	16#00		Hex	SINT	Data ...

- b. WRITE - Example

Name	Value	Force Mask	Style	Data Type	Description
arrISDUCommand	{...}	{...}	Hex	SINT[232]	
arrISDUCommand[0]	16#18		Hex	SINT	ISDU Index low
arrISDUCommand[1]	16#00		Hex	SINT	ISDU Index high
arrISDUCommand[2]	16#00		Hex	SINT	Subindex
arrISDUCommand[3]	'T'		ASCII	SINT	Data 0
arrISDUCommand[4]	'E'		ASCII	SINT	Data 1
arrISDUCommand[5]	'C'		ASCII	SINT	Data ...
arrISDUCommand[6]	'H'		ASCII	SINT	
arrISDUCommand[7]	' '		ASCII	SINT	
arrISDUCommand[8]	'C'		ASCII	SINT	
arrISDUCommand[9]	'O'		ASCII	SINT	
arrISDUCommand[10]	'N'		ASCII	SINT	
arrISDUCommand[11]	' '		ASCII	SINT	
arrISDUCommand[12]	'L'		ASCII	SINT	
arrISDUCommand[13]	'U'		ASCII	SINT	
arrISDUCommand[14]	'T'		ASCII	SINT	
arrISDUCommand[15]	'M'		ASCII	SINT	
arrISDUCommand[16]	16#00		Hex	SINT	

SICK	Application Notes		IOLG2
	Date: 14.6.2016	last mod.: 14.06.2016 16:15:00	Proj. Nr.

6. Start the READ or WRITE operation by enabling your trigger.

Here the example for the ST code snippet from above:

SD_TriggerREAD	[1]	Decimal	BOOL	Trigger ISDU reading
SD_TriggerWRITE	[0]	Decimal	BOOL	Trigger ISDU writing

7. For READ operations you find the data in your receive array

Name	Value	Force Mask	Style	Data Type
arrISDUReceive	{ ... }	{ ... }	Hex	SINT[232]
+ arrISDUReceive[0]	16#00		Hex	SINT
+ arrISDUReceive[1]	'I'		ASCII	SINT
+ arrISDUReceive[2]	'E'		ASCII	SINT
+ arrISDUReceive[3]	'C'		ASCII	SINT
+ arrISDUReceive[4]	'H'		ASCII	SINT
+ arrISDUReceive[5]	'_'		ASCII	SINT
+ arrISDUReceive[6]	'C'		ASCII	SINT
+ arrISDUReceive[7]	'O'		ASCII	SINT
+ arrISDUReceive[8]	'N'		ASCII	SINT
+ arrISDUReceive[9]	' '		ASCII	SINT
+ arrISDUReceive[10]	'L'		ASCII	SINT
+ arrISDUReceive[11]	'U'		ASCII	SINT
+ arrISDUReceive[12]	'T'		ASCII	SINT
+ arrISDUReceive[13]	'M'		ASCII	SINT

Reading and writing of ISDU data is easier if you use function block which is provided by SICK. On how to use these function blocks please refer to the manual in the ZIP package of the function block itself.