Description

EtherNet/IP

process interface

Version 1.0

Contents

1. Preliminary note

This document serves to explain an EtherNet/IP communication between an ifm vision sensor with EtherNet/IP interface and a PLC with EtherNet/IP capacity.

2. General

The Ethernet Industrial Protocol (EtherNet/IP) is an open standard for industrial networks. EtherNet/IP serves for the transmission of cyclic I/O data as well as acyclic parameter data. EtherNet/IP provides a broad basis for effective data communication in the industry.

EtherNet/IP extends Ethernet by a modern industrial protocol (CIP, Common Industrial Protocol) as an application layer for applications in automation. This makes Ethernet ideal for use in industrial control technology.

3. EtherNet/IP settings

The ifm object recognition sensor as an EtherNet/IP adapter device supports the communication with a device configured as EtherNet/IP scanner. This is usually the processor (e.g. a PLC).

Communication is carried out using 2 EtherNet/IP assemblies; one for data transport from the controller to the sensor ("Output Assembly Instance”**,** ID address 100 / 0x64) and one for data transport from the sensor to the controller ("Input Assembly Instance”**,** ID address 101 / 0x65).

The same lengths of the assemblies must be set in the sensor and the controller.

- "Length of the consuming assembly" defines the length of the "Output Assembly Instance" (ID 100)
- "Apply segmentation" activates the definition of the different "Input Assembly Instance" (ID 101) parameters.

3.1 Structure of the assembly in case of deactivated segmentation

The "Input Assembly Instance" (101) is 450 bytes long and divided into 3 segments:

	Offset	Length	Content
Segment 1	0	215 bytes	Reply to the incoming messages
Segment 2	215	215 bytes	Byte result of the code or image evaluations
Segment 3	430	20 bytes	**Fixed device and result information**

The last byte of each segment serves as control byte which is incremented during processing by the object recognition sensor. With identical code content these control bytes are used for distinguishing the input data.

Test	Offset	Length	Check box designation
Control byte segment 1	214	1 byte	Reply
Control byte segment 2	429	1 byte	Evaluation
Control byte segment 3	449	1 byte	Device information

3.2 Segmentation

In the "Output Assembly Instance" messages are always written as from address 0; only the length can be determined. It must correspond to at least the length of the longest possible message (max. 450 bytes).

The "Input Assembly Instance" can be segmented in order to save memory space. This way, only the data which is actually required for the application is transmitted. For each segment, an "Offset" and the required "Length" can be selected from the segment selection list.

The predefined "Input Assembly Instance" segmentation can be reconfigured using the segmentation table. The segmentation table provides a new order of the bytes for the "Input Assembly Instance". A segment is defined by its index, a number of bytes (segment length) and a byte address from the predefined "Input Assembly Instance" (segment offset).

The "Input Assembly Instance" is restructured on this basis. The segment index defines the order of assignment. The number of assigned bytes is defined by the segment length and the segment offset points to the address from the predefined "Input Assembly Instance" from which the bytes are extracted.

The last byte of each segment can be activated as control byte. This is then incremented by the object recognition sensor during processing. With identical code content these control bytes are used for distinguishing the input data. The control byte can be activated or deactivated by clicking onto the respective field.

Caution! For a correct segmentation, all bytes from the "Input Assembly Instance" must be unambiguously represented.

3.2.1 Segmentation example 1

Segmentation table

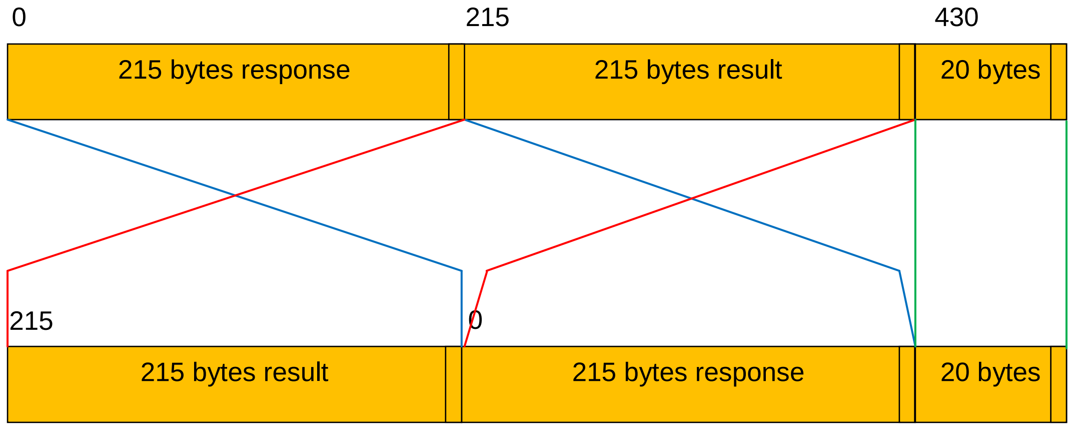
Index	Offset	Length
1	0	450 bytes

Explanation:
Default segmentation table. Takes 450 bytes (all!) from the predefined "Input Assembly Instance" and positions these on the address 0. Therefore, this segmentation table has no actual influence on the "Input Assembly Instance".

3.2.2 Segmentation example 2

Segmentation table

Index	Offset	Length
1	215	215 bytes
2	0	215 bytes
3	430	20 bytes



Explanation:
Segments 1 and 2 of the predefined "Input Assembly Instance" are swapped: First, 215 bytes starting with byte address 215 of the predefined "Input Assembly Instance" are repositioned. Then, 215 bytes starting with byte address 0 and then 20 bytes starting with byte address 430 from the predefined "Input Assembly Instance".

4. Data exchange via EtherNet/IP

The data exchange between a sensor with EtherNet/IP capacity and a PLC is carried out cyclically. This means that the data stored in the sensor in the output assembly segment (ID 100) is retrieved from the connected PLC in each cycle and stored in the data area defined in the PLC.

If the data in the sensor changes, it will be adopted in the defined data area in the next cycle of the PLC and will be available until the sensor overwrites its output assembly area.