OMRON

V460-H

Industrial Handheld DPM Reader

Communication Manual

Z462-E1-01

NOTE

- All rights reserved.
- No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.
- No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions.
  Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- Microsoft, Windows, Windows Vista, Excel, and Visual Basic are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- ODVA, CIP, CompoNet, DeviceNet, and EtherNet/IP are trademarks of ODVA.
- QR Code is a registered trademark of DENSO WAVE INCORPORATED.

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

Copyrights

Microsoft product screen shots reprinted with permission from Microsoft Corporation.

Introduction

Thank you for purchasing the V460-H Industrial Handheld DPM Reader.
This manual contains information that is necessary for using V460-H Industrial Handheld DPM Reader.
Please read this manual and make sure you understand the functions and capabilities before you attempt to use it in a control system.
Function Blocks Library and Sample Program for Omron Controllers are available for download.
Please, visit Omron website for Function Blocks Library and Sample Program for additional PLC / Controllers.
Keep this manual in a safe place where it will be available for reference during operation.

Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of introducing barcoding systems.
- Personnel in charge of designing barcoding systems.
- Personnel in charge of installing and maintaining barcoding systems.
- Personnel in charge of managing barcoding systems and facilities.

Applicable Products

This manual covers the following product:

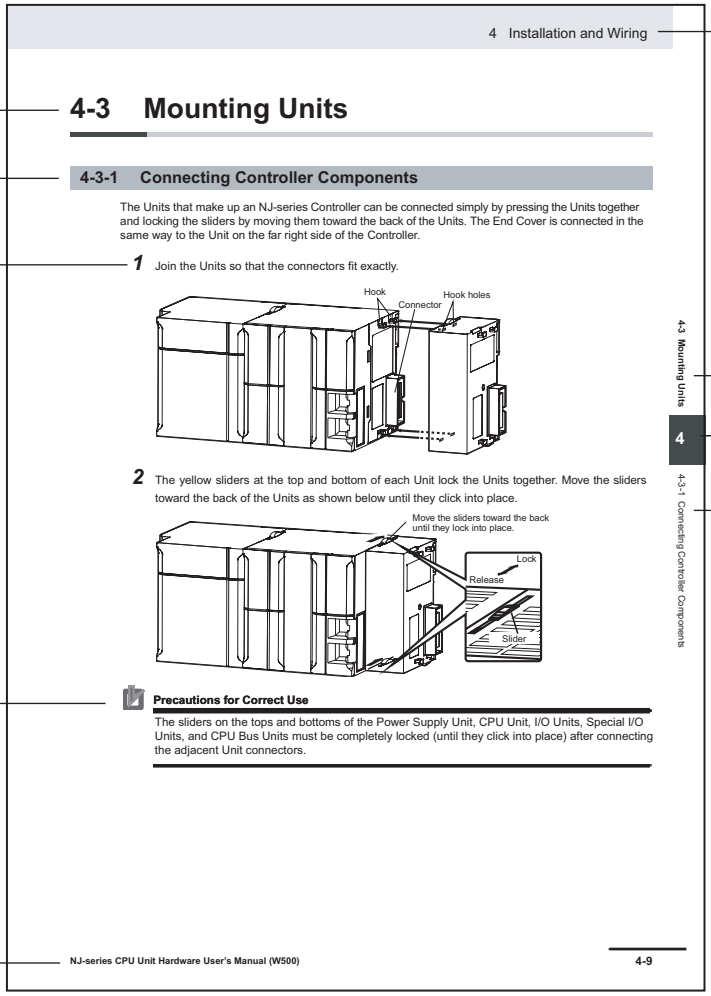
- V460-H Industrial Handheld DPM Reader

Parts of the specifications and restrictions for each product may be listed in other manuals. Please refer to *Related Manuals* on page 15.

Manual Structure

Page Structure

The following page structure is used in this manual.



Level 2 heading — **4-3 Mounting Units**

Level 3 heading — **4-3-1 Connecting Controller Components**

A step in a procedure — ***1*** Join the Units so that the connectors fit exactly.

Indicates a procedure.

2 The yellow sliders at the top and bottom of each Unit lock the Units together. Move the sliders toward the back of the Units as shown below until they click into place.

Special information — **Precautions for Correct Use**

The sliders on the tops and bottoms of the Power Supply Unit, CPU Unit, I/O Units, Special I/O Units, and CPU Bus Units must be completely locked (until they click into place) after connecting the adjacent Unit connectors.

Icons indicate precautions, additional information, or reference information.

Manual name — NJ-series CPU Unit Hardware User's Manual (W500)

Level 1 heading — 4 Installation and Wiring

Level 2 heading — 4-3 Mounting Units

Level 3 heading — 4-3-1 Connecting Controller Components

Gives the current headings.

Page tab — 4

Gives the number of the main section.

Note : This page is a sample for the purpose of describing the page structure. It differs in its actual content.

Icons

The icons used in this manual have the following meanings.

Precautions for Safe Use

Precautions on what to do and what to avoid doing to ensure the safe use of the product.

Precautions for Correct Use

Precautions on what to do and what to avoid doing to ensure proper operation and performance.

Additional Information

Additional information to read as required.
This information is provided to increase understanding or make operation easier.

Sections in this Manual

CONTENTS

Section 3 Controlling Operation and Data Output with PROFINET

Appendices

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

- **Exclusive Warranty**

  Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

- **Limitations**

  OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

  Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

- **Buyer Remedy**

  Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

  See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY

WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may

be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Safety Precautions

For details on Safety Precautions, please refer to *Safety Precautions* in *V460-H Industrial Handheld DPM Reader User Manual (Z461 (84-9000460-02))*.

Precautions for Safe Use

For details on Precautions for Safe Use, please refer to *Precautions for Safe Use* in *V460-H Industrial Handheld DPM Reader User Manual (Z461 (84-9000460-02))*.

Precautions for Correct Use

For detailed precautions on the correct use of the product, please refer to *Precautions for Correct Use* in *V460-H Industrial Handheld DPM Reader User Manual (Z461 (84-9000460-02))*.

Regulations and Standards

For details on Regulations and Standards, please refer to *Regulations and Standards* in *V460-H Industrial Handheld DPM Reader User Manual (Z461 (84-9000460-02))*.

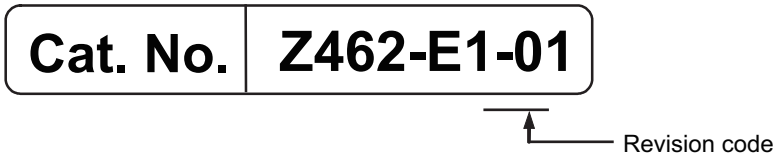
Related Manuals

The followings are the manuals related to this manual. Use these manuals for reference.

Name of Manual	Cat. No.	Model	Usage	Description
V460-H Industrial Handheld DPM Reader User Manual	Z461	V460-H Industrial Handheld DPM Reader	When you want to know the product specifications and basic settings for using the V460-H Industrial Handheld DPM Reader	V460-H Industrial Handheld DPM Reader specifications, getting started, explanation of settings, command parameters.
V460-H Industrial Handheld DPM Reader Communication Manual	Z462		When you want to operate the V460-H Industrial Handheld DPM Reader from an external device	It describes the system configuration, control methods, I/O specifications, supported network types and communication setting for using the V460-H Industrial Handheld DPM Reader.

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.



Revision Code	Date	Revised Content
01	September 2023	First Publication.

1

Communication Specifications Overview

This section provides a basic overview of the communications specifications and methods for controlling the code readers. This information is required before performing communications between the V460-H Industrial Handheld DPM Reader and an external device.

1-1 Confirming the System Configuration

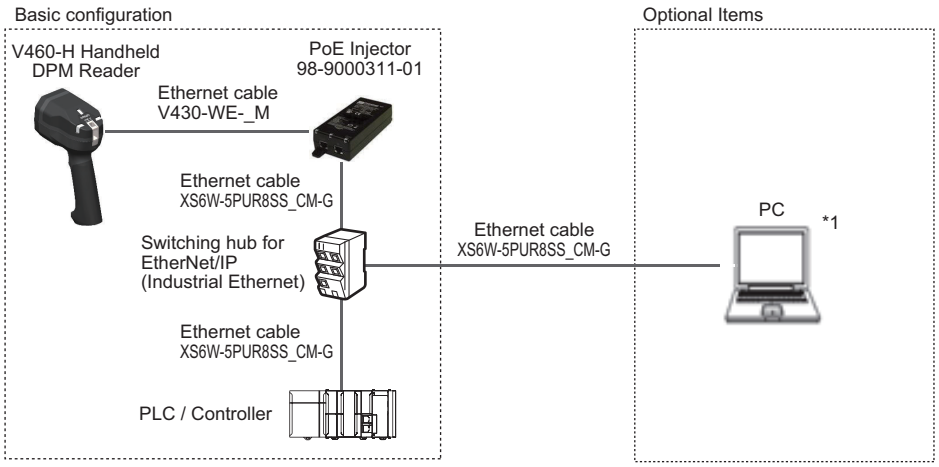
This product is a multi-code reader that captures images of 1D symbols (barcodes) and 2D Symbols and reads and processes their embedded data.
In a system configuration in which it is connected to a PLC, PC, or other external device, serial commands can be received from, and code reading results can be output to the external device.

1-1-1 V460-H Series System Configuration

The V460-H can be used in the following types of system configurations.

Connecting over Ethernet (EtherNet/IP, Serial (TCP), PROFINET)

Establish network connections via an Ethernet cable to input triggers and communication commands and to output reading results (Judgment results and decoded content). Using the data link function for each network (excluding Serial), data transfer can be done periodically between the code reader and the external device.



*1 If monitor display is not required, it is not necessary to connect with a PC during operation.

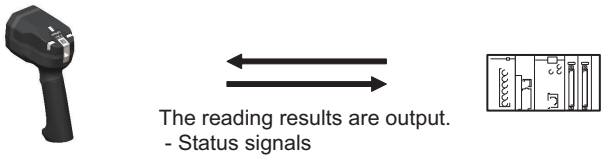
NOTE*:* Once the reader is connected, If the Ethernet cable (PoE injector to PLC/Controller) will be unplugged (on either side), the reader will start beeping to notify the user.

1-2 Communicating with an External Device

This section gives the communications specifications, describes the control methods that you can use for communications, and describes the settings that are required before starting communications with an external device.

1-2-1 Basic Control Operations of the Code Reader

The following figure shows basic communications between an external device and the code reader and the flow of signals and data.



The following methods can be used to exchange data between an external device and the code reader.

Commands that can be input to the code reader from an external device

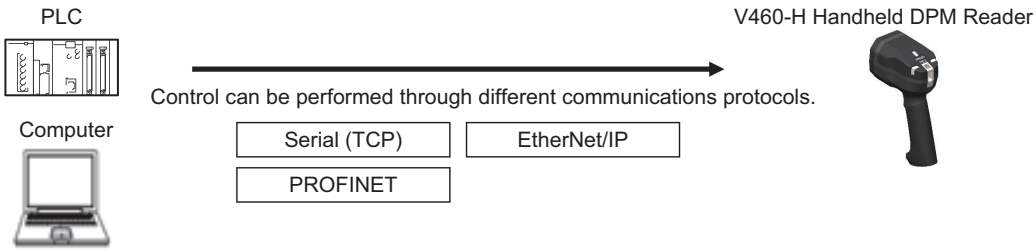
Type		Description
Control Commands	Communication Command Input	Various commands can be executed, such as a Read commands (trigger), enable matchcode, clear counters. The communication commands differ depending on the communications protocol that you use.

Data output from the code reader to an external device

Type	Description
Status Signals	When the code reader confirms the input of a control signal or communication command and starts the reading process, it notifies the external device of its status (by signals such as InReadCycle, etc.) and its judgement with the OK/NG Judgment signal.
Read Character String Output	You can output the character string read from barcodes, or 2D Codes
Additional Information	Additional data such as read time and code position coordinates can be output. For items appended to the output, they must be setup in advance in WebLink$_{HH}$'s advanced settings menu.

1-2-2 Applicable Communications Protocols for the V460-H Industrial Handheld DPM Reader

The V460-H Industrial Handheld DPM Reader can be controlled from a PLC, computer, or other external device using various communication protocols.
The following types of communication protocols can be used for controlling the V460-H Industrial Handheld DPM Reader from an external device.



Applicable Communications Protocols

○: Supported -: Not supported

Communication Method	Communication Protocol	Description	Communication Cable Type
			Ethernet
Data Sharing	EtherNet/IP	This is an open communications protocol. Tag Data Links are used for communication with the code reader. On the PLC, structured variables are created that correspond to the control signals, Command/Response data, and Read data. These variables are then used as I/O Tag Data Links to exchange data between the PLC and the code reader.	○
	PROFINET	This is an open communications protocol. Software-based RT (Real-time) communications, (SRT) is used for communication with the code reader. The control signals, Command Area/ Response Area, and area to store Read result data are assigned in the I/O memory of the PLC, and data is exchanged cyclically between the PLC and the code reader.	○
Frame Transmission	Serial (TCP)	Command frames are sent to the code reader and Response frames are received from the code reader without the use of any specific protocol. Data can be exchanged between the PLC, computer, or other external device and the code reader in ASCII or binary format.	○

Simultaneous Use of Communication Methods and Connections

○: Supported -: Not supported

Code reader Connection Method	Simultaneous Connection Method		
	EtherNet/IP	PROFINET	Serial (TCP)
EtherNet/IP	N/A	-	○
PROFINET	-	N/A	○
Serial (TCP)	○	○	N/A

Additional Information

About connections over network routers
$WebLink_{HH}$ can connect to code reader on different networks across routers.

- To connect to the code reader, enter code reader's IP address from the browser.
- Set a fixed IP address for the code reader you wish to connect to.

2

Controlling Operation and Data Output with Ethernet

2-1 Controlling Operation and Data Output with EtherNet/IP

2-1-1 EtherNet/IP Overview

EtherNet/IP is an industrial multi-vendor network that uses Ethernet. The EtherNet/IP specifications are open standards managed by the ODVA (OpenDeviceNet Vendor Association). EtherNet/IP is used by a wide range of industrial devices.
Because EtherNet/IP uses standard Ethernet technology, various general-purpose Ethernet devices can be used in the network.
EtherNet/IP has mainly the following features.

- **High-speed, High-capacity Data Exchange through Tag Data Links (Cyclic Communications)**
  The EtherNet/IP protocol supports implicit communications, which allows cyclic communications (called Tag Data Links) with EtherNet/IP devices.

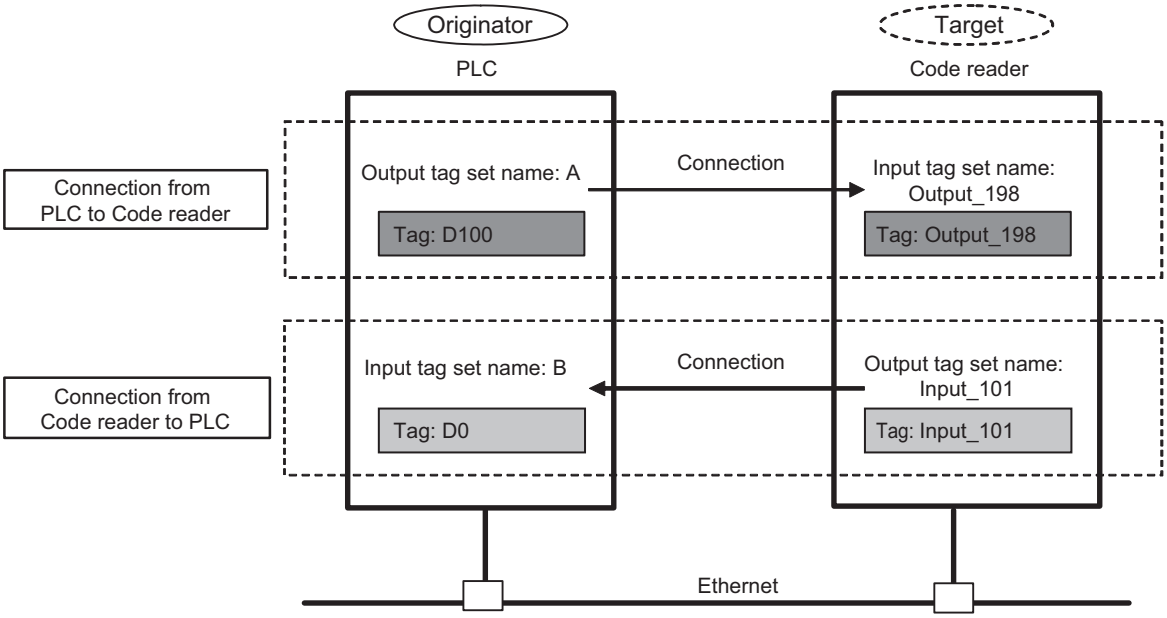
- **Tag Data Links are set at the specified communication cycle for each application regardless of the number of nodes**
  Because the data is exchanged over the network at the refresh cycle that is set for each connection regardless of the number of nodes, that refresh cycle will not increase even if the number of nodes increases. (Data exchange in the connection is kept in synch)
  Because the refresh cycle can be set for each connection, each application can communicate at its ideal refresh cycle. (For example, interprocess interlocks can be transferred at high speed, while the production commands and the status monitor information are transferred at low speed.)

Precautions for Correct Use

On a network to which many devices are connected, performance may drop (e.g., responses may be delayed or packets lost) or communications errors may occur when there is temporarily high traffic on the network.
Test the operation under actual conditions before you start actual operation of the system.

Data Exchange with EtherNet/IP (Implicit Communications)

Data is exchanged cyclically between Ethernet devices on the EtherNet/IP network using Tag Data Links as shown below.



- **Data Exchange Method**
  To exchange data, a connection is opened between two EtherNet/IP devices.
  One of the nodes requests the connection to open a connection with a remote node.
  The node that requests the connection is called the *Originator* and the node that receives the request is called the *Target*.

- **Data Exchange Memory Locations**
  The memory locations that are used to exchange data across a connection are specified as tags.
  You can specify memory addresses or variables for tags.
  A group of tags consists of an output tag set and an input tag set.

Additional Information

Message communications are used when communicating over EtherNet/IP with a PLC that does not support Tag Data Link communications (*2-1-10 Communicating with the Code Reader with EtherNet/IP Message* on page 2-28).

2-1-2 Communication with the Code Reader over EtherNet/IP Connection

You can use an EtherNet/IP Tag Data Link to communicate between the PLC and the code reader. The PLC can control the code reader with Command/Response communications and the code reader can output data after executing a Read.

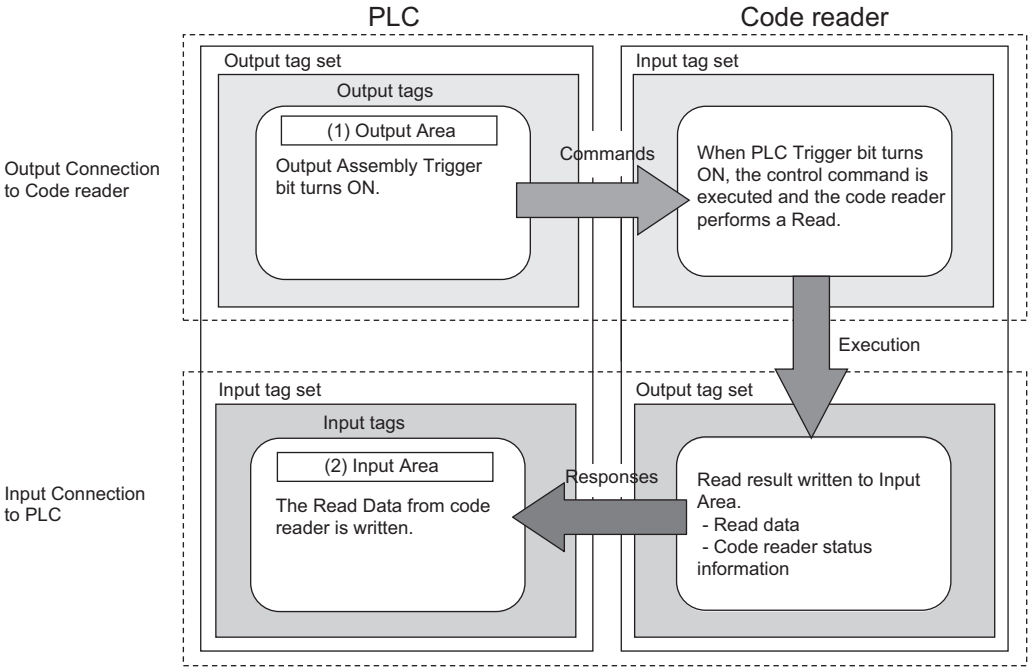
To connect to OMRON Controllers and communicate through EtherNet/IP, you can use Sysmac Studio, or Network Configurator to set up the Tag Data Links (tags, tag sets, and connection settings). For more detailed information on Tag Data Link settings, please refer to the following manuals.

- *NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual (Cat. No. W506)*
- *CS/CJ-series EtherNet/IP Units Operation Manual (Cat. No. W465)*
- *CJ-series EtherNet/IP Units Operation Manual for NJ-series CPU Unit (Cat. No. W495)*

Function Blocks Library and Sample Program for Omron Controllers are available for download. Please, visit Omron website for Function Blocks Library and Sample Program for additional PLC / Controllers.

Types of Communication Areas

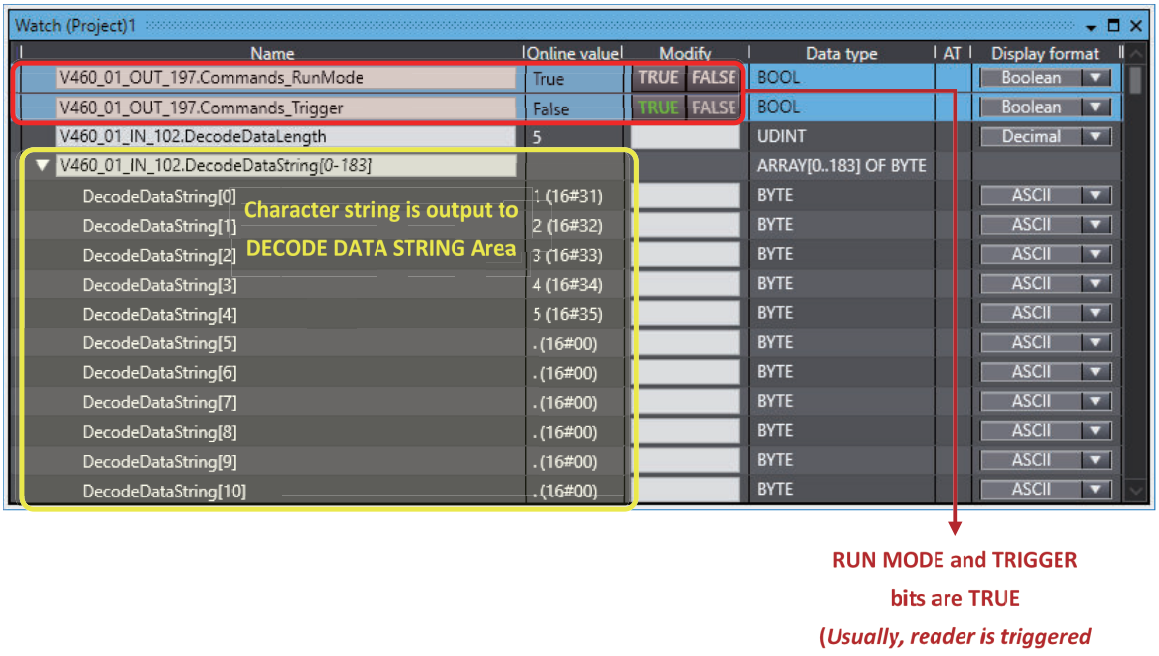
For EtherNet/IP, communication with a PLC, the communication is performed using two communication areas on the PLC, the Input Field and the Output Field. This code reader has 6 types of Input Field Assemblies and 2 types of Output Field Assemblies, and one can be selected for each.



2-1-3 Communication Flow Between PLC and Code Reader

1. The PLC (User) changes the Trigger bit assigned to the memory area (Output Field) of the PLC in advance from OFF to ON.
2. When the Trigger bit from the PLC is ON, the code reader executes a Read process.
3. After the code reader's Read process is complete, it then stores its Read data in the specified memory area (Input Field) on the PLC.

[Output Data Example]



2-1-4 Communication Settings (EtherNet/IP)

Using WebLink$_{HH}$ to Set the Code Reader Network Settings

Use WebLink$_{HH}$ to set the IP address on the code reader to match the network settings of the PLC or other external device.

- WebLink$_{HH}$ - **Setup** - **Gear Icon** - **Advanced Settings** - **Communications** - **Ethernet**

1 Set the **IP Address** and **Subnet mask** according to the network settings of the PLC or other external device.

Setting Item	Setting Value	Description
IP Address	a.b.c.d a: 0 to 255 b: 0 to 255 c: 0 to 255 d: 0 to 255 (Default: 192.168.188.2)	Enter the IP address of the Code Reader
Subnet	a.b.c.d a: 0 to 255 b: 0 to 255 c: 0 to 255 d: 0 to 255 (Default: 255.255.0.0)	Input the subnet mask address.
Gateway	a.b.c.d a: 0 to 255 b: 0 to 255 c: 0 to 255 d: 0 to 255 (Default: 0.0.0.0)	If a Gateway is used, enter the gateway address. If a Gateway is not used, use the default value 0.0.0.0.
IP Address Mode	• Fixed (Default) • DHCP	In Fixed mode, the code reader uses a user-defined IP address. In DHCP mode, the code reader acquires its IP address, subnet, and gateway from the DHCP server. For PLC communication, **Fixed IP Address Mode** is mandatory. DO NOT enable DHCP in this case.

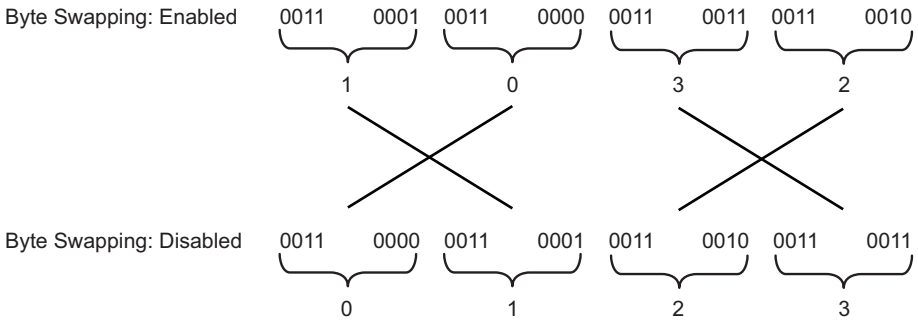
Using WebLink$_{HH}$ to Set Up EtherNet/IP Communication

- WebLink$_{HH}$ - **Setup** - **Gear Icon** - **Advanced Settings** - **Communications** - **Ethernet**

Setting Item	Setting Value	Description
EtherNet/IP	• Enabled • Disabled	• Enabled: EtherNet/IP connectivity is enabled on the code reader. • Disabled: EtherNet/IP connectivity is disabled on the code reader.
Ethernet/IP Byte Swapping	• Enabled • Disabled	• Enabled: Byte Swapping is enabled for the Read data. The Read data is stored in Decode Data in Little endian format. This is used when the Endian of the CPU architecture is difer-ent from that of the Read data. • Disabled: Byte Swapping is disabled for the Read data. The Read data is stored in Decode Data in Big endian format.

When Byte Swapping is used, the output changes as follows.

Example: Where the character string of the read code is *0123*



Note: Byte Swapping only applies to the Decode Data String memory region of the EtherNet/IP Input Assemblies.

2-1-5 Tag Data Link Setting Methods

This section describes how to set data links for EtherNet/IP.
The communications areas in the PLC for which data links to the code reader are created are specified as tags and tag sets, and the connections are set for tag data link communications.

Precautions for Correct Use

When connecting to an NJ/NX-series or CJ-series CPU Unit, install the EDS file that defines the connection information for the code reader in to Sysmac Studio.
Download the EDS file from OMRON's website.

Tags, Tag Sets, and Connection Settings

The code reader has 6 types of Input Assemblies and 2 types of Output Assemblies, and one type can be selected for each. The Data Structure changes based on the selected Assembly.
For more detailed information about Memory Allocation and the Data Structure of each Assembly, please refer to *A-2 EtherNet/IP Specifications* on page A-3.

Assemblies

Assembly Name	Connection I/O Type	Input / Output	Assembly ID	Size (bytes)	Used with Output	Assembly Description	Data Structure
Small Input	IO small	Input	100	84	198	It is a compact, lightweight input assembly. Holds 64 bytes of Read data.	*1
Large Input	IO large	Input	101	176	198	Allows for more Device Status Information to be stored for verification than what can be stored with the Small Input Assembly. Holds 128 bytes of Read data.	
MXL/SLC Input	Input MXLSLC	Input	102	258	197	Allows advanced Device Status Information too large to be stored in Large Input Assembly to be stored for verification. Holds 184 bytes of Read data.	
1 Decode Input	Input 1 Decode	Input	103	500	197	Holds 436 bytes of Read data.	
4 Decode Input	Input 4 Decode	Input	104	500	197	Holds Read result information for 4 symbols. The first Read data is stored in a 160 byte Area and the 2nd to 4th Read data are stored in the 72 byte Area.	
N Decode Input	Input N Decode	Input	105	500	197	Holds Symbol information and Read result information for any number of symbols. Holds 456 bytes of Read data.	
Output	-	Output	197	4	-	For commands to be sent to the code reader.	
Output (Legacy)	-	Output	198	12	-	Commands and Command Echo for fixed data can be sent to the code reader.	

*1. refer to *A-2 EtherNet/IP Specifications* on page A-3.

Tag Set Settings

Setting Item	Setting
Input	
Tag Set Name	Tag Set Name on PLC
Size	Input Assembly Dependency • 84, 176, 248, 500 byte
Output	
Tag Set Name	Tag Set Name on PLC
Size	Output Assembly Dependency • 4 and 12 byte

Connection Settings

Setting Item	Setting
Input	
Assembly ID	Input Assembly Dependency • 100, 101, 102, 103, 104, 105
Size	Input Assembly Dependency • 84, 176, 248, 500 byte
Originator Variable	Variable defined on the PLC
Size	Input Assembly Dependency • 4 and 12 byte
Connection type	Point to Point connection
RPI	4.0 to 65.0ms (Default: 10.0ms)
Timeout	RPI × (4 to 512) (Default: RPI × 512)
Output	
Assembly ID	Output Assembly Dependency • 197, 198
Size	Output Assembly Dependency • 4 and 12 byte
Originator Variable	Variable defined on the PLC
Size	Output Assembly Dependency • 4 and 12 byte
Connection type	Point to Point connection

Precautions for Correct Use

- If I/O memory addresses are specified for the communications areas, the information in the communications areas will be cleared when the operating mode of the PLC changes unless addresses in the CIO Area, which holds memory, are specified.
- The following Assembly objects are required to specify instances when the EDS file is not used.

Setting the Assembly Object

Setting Item	Setting Value	Note
Instance ID	100	Small Input
	101	Large Input
	102	MXL/SLC Input
	103	1 Decode Input
	104	4 Decode Input
	105	N Decode Input
	197	Output
	198	Output (Legacy)

2-1-6 Status and Control Signals for Each Input and Output Assembly

This code reader has the following types of Input Assemblies.

1. Small Input
2. Large Input
3. MXL/SLC Input
4. 1 Decode Input
5. 4 Decode Input
6. N Decode Input

The Status signals are as follows.
These signals are controlled automatically based on the status of the code reader.

○: Verifiable ×: Not Verifiable

Status Signal	Description	1	2	3	4	5	6
InReadCycle	While in Read Cycle, this bit is set to *1*.	×	○	×	×	×	×
Trigger Acknowledged	This bit becomes *1* when the Trigger bit from the Output Assembly is received. When the Trigger bit is OFF, Trigger Acknowledged also becomes *0*.	×	×	○	○	○	○
Exposure Done	When Exposure is done, this bit becomes *1*. During exposure, this bit is set to *0*.	×	×	○	○	○	○
Decoding	When reader is decoding image, this bit is set to *1*. When the decode is completed, this bit becomes *0*.	×	×	○	○	○	○
Data is Ready	When the data from Read Cycle Report and Data Cycle Report is confirmed, this bit becomes *1*. When the next Read starts, this bit becomes *0*.	×	×	○	○	○	○
Read Cycle Pass	On Good Read (or Match if Matchcode enabled), bit becomes *1*. When the next Read starts, this bit becomes *0*.	×	×	○	○	○	○
Read Cycle Fail	On No Read (or Mismatch if Matchcode enabled), bit becomes *1*. When the next Read starts, this bit becomes *0*.	×	×	○	○	○	○
Decode Data	This field stores the Read string. When additional information such as a Print Quality Grading Standard is set, it is stored following the Read string.	○	○	○	○	○	○

This code reader has the following types of Output Assemblies.

1. Output Assembly
2. Output Assembly (Legacy)

The Control Signals are as follows.
They can be controlled by the user at an arbitrary timing.

○: Verifiable ×: Not Verifiable

Control Signal	Description	1	2
Trigger	Executes Read. The code reader recognizes this bit changing from 0 to 1 as the rising edge of the trigger and its change from 1 to 0 as the falling edge of the trigger.	○	○
New Master	When this bit is ON, the next Read result is registered as the Master Symbol.	○	○

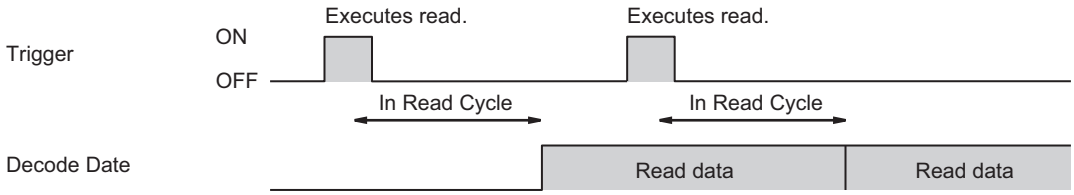
2-1-7 Timing Charts by Assembly Type

Read is executed by the Read (TRIG) Signal

The timing signal at completion of storing the Read data to PLC data memory differs by the Input Assembly type.

● Small Input (100)

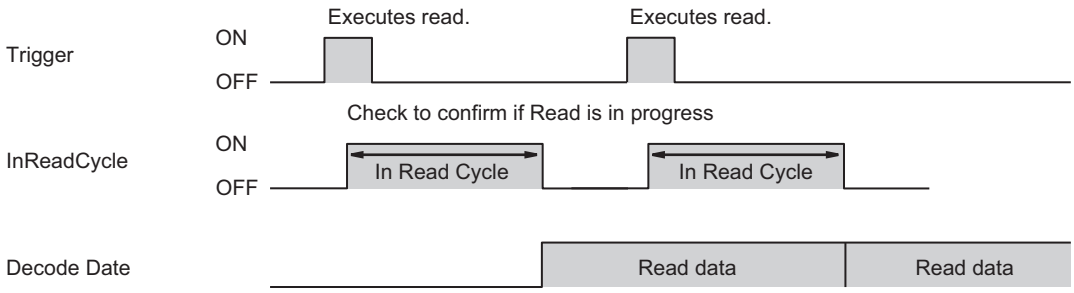
It does not correspond to the Timing Signal for storing Read data.



1. Reading starts at the rising edge of the Trigger.
2. At the end of reading, the read data is stored in Decode Data.

● Large Input (101)

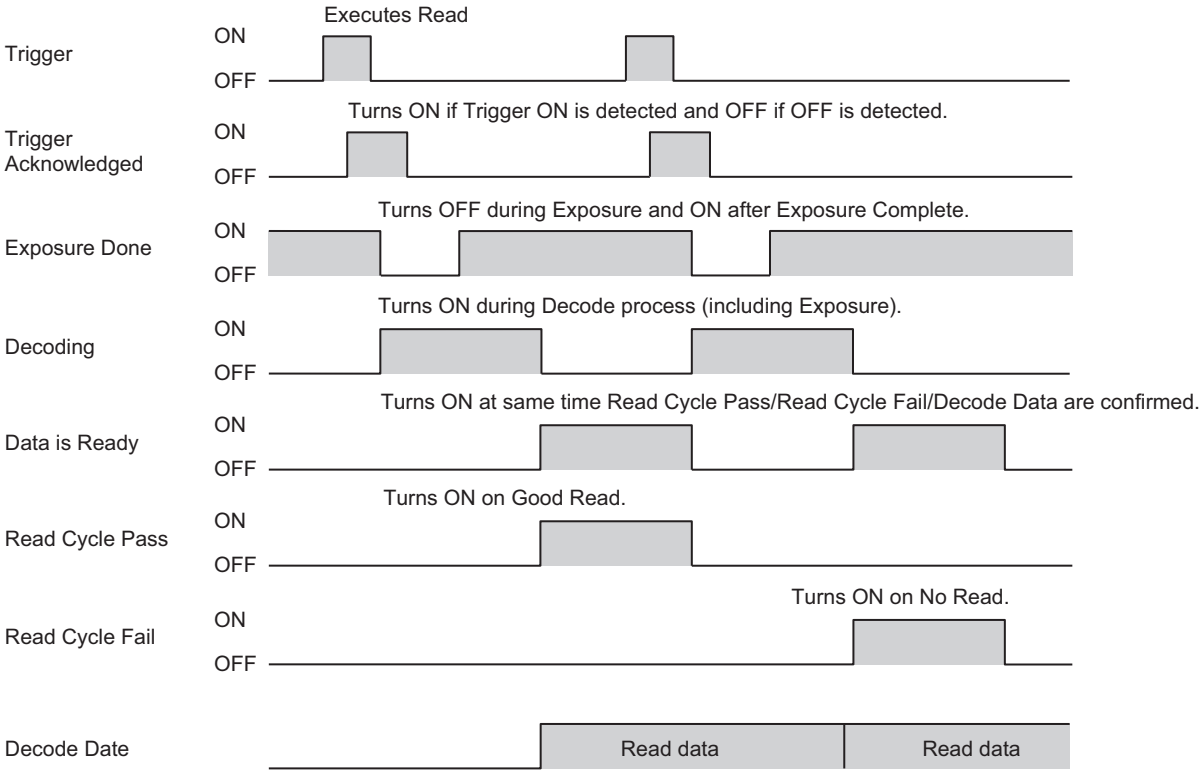
It is output at the timing of the **Device Status** - **InReadCycle** bit turning from ON → OFF.



1. Reading starts at the rising edge of the Trigger.
2. At start of Read, InReadCycle turns ON and Trigger turns OFF.
3. At end of Read, the Read data is stored in Decode Data and InReadCycle turns OFF.

● MXL/SLC Input (102) through N Decode Input (105)

It is output at the timing of the **Device Status** - **Decoding** bit turning from ON → OFF.



1. Reading starts at the rising edge of the Trigger.
2. Trigger Acknowledged turns ON when Trigger ON is detected and turns OFF when Trigger OFF is detected.
3. ExposureDone turns OFF when exposure starts and turns ON when exposure completes.
4. Decoding is ON during decoding processing. The Decoding process overlaps the Exposure process.
5. Data is Ready turns ON at the same time Decode Data / Read Cycle Pass or Read Cycle Fail is confirmed.
6. Read Cycle Pass turns ON when there is a Good Read and Read Cycle Fail turns ON when there is a No Read. The Read data is stored in Decode Data.

Additional Information

There can be up to a 10ms delay in the Output timing of the Symbol data.

7. When the next Trigger is detected, Data is Ready turns OFF.

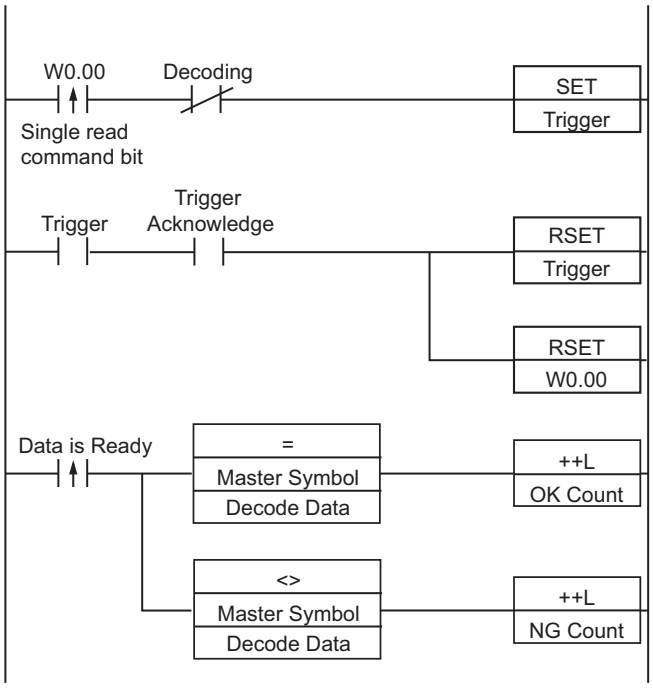
2-1-8 Sample Ladder Program

A sample ladder program to execute the following operation.

- Input the Trigger Signal to execute Triggered Read.
- The read character string (Decode Data) is compared with the Verification string (Master Symbol) stored in the PLC.
- If they match, it is added to the OK/Match count, and if they do not match, it is added to the Mismatch/NG count.

The following Input and Output Assemblies are used.

- Input Assembly: MXL/SLC Input (102)
- Output Assembly: Output (197)



1. When the flag for Triggered is ON, The Trigger Bit turns ON.
2. The Trigger Acknowledged Bit (for detecting trigger input) is ON.
3. When the Trigger Acknowledged Bit ON is detected, the Trigger Bit turns OFF.
4. When Read is completed, the Data is Ready Bit turns ON.
5. The Read string (Decode Data) is compared with the Verification string (Master Symbol).
6. If the two strings match, the Match/OK Count is incremented by 1.
7. If the two strings do not match, the Mismatch/NG Count is incremented by 1.

2-1-9 Accessing the NJ/NX-series Controller Communication Areas using Variables

With an NJ/NX-series, accessing the I/O memory allocated to each communication area can be done with the user program with the use of variables.
Here is an example of using the MXL/SLC Input (102) and Output (197) for that purpose.
For more detailed information about the data structure of each Assembly, please refer to *A-2 EtherNet/IP Specifications* on page A-3.

Access Using Network Variables

Create user-defined variables that match the structures of the communications areas of the Sensor.
Use the Sysmac Studio to define the variables.
For how to use Sysmac Studio, please refer to *Sysmac Studio Version1 Operation Manual (W504)*.

1 Defining the Data Types of the Variables

Define data types for variables that match the structures of the communications areas.

1) Defining a Data Type for Control Signal Access

First, define a BOOL array data type to access the control signals and status signals.
Here, we define the Data types, *COMMAND* and *Device_Status*.

Control Signal

Data Name		Data Type
COMMAND		ARRAY[0..31] OF BOOL
	Run_Mode	BOOL
	Trigger	BOOL
	Enable_Matchcode	BOOL
	…	
	Reserved	ARRAY[0..21] OF BOOL

Status Signals

Data Name		Data Type
Device_Status		ARRAY[0..31] OF BOOL
	Run Mode	BOOL
	Trigger_Acknowledged	BOOL
	Exposure_Done	BOOL
	…	
	Buffer_Overflow	BOOL
	Reserved	ARRAY[0..9] OF BOOL

2) Defining Data Types for Communications Area Access

Data types are defined according to the communication area to access, with one data type for Output Area and another data type for Input Area.
Here, there are two Data types defined, *S_EIPOutput197* and *S_EIPInput102*.

- Data Type to access Output Area
  Data type name: S_EIPOutput197
  Type of derivative data type: Structure

Data Name	Data Type
S_EIPOutput197	STRUCT
COMMANDS	COMMAND

Example assignments of Variable Data Type for Output Area:

	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+0	Reserved			*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	TRIG	*1
+1	Reserved															

*1. For Bits other than TRIG, please refer to *Output (Instance ID: 197)* on page A-24.

- Data Type to access Input Area

  Data type name: S_EIPInput102

  Type of derivative data type: Structure

Data Name	Data Type
S_EIPInput102	STRUCT
RESERVED	BYTE
RESERVED	BYTE
RESERVED	BYTE
RESERVED	BYTE
DEVICE_STATUS	Device Status
RESERVED	DINT
COUNTERS	ARRAY[0..5] OF DINT
READ_CYCLE_REPORT	ARRAY[0..3] OF INT
DECODE_CYCLE_REPORT	ARRAY[0..3] OF DINT
DECODE_LENGTH	DINT
DECODE_DATA	ARRAY[0..183] OF BYTE

Example assignments of Variable Data Type for Input Area:

	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+0	RESERVED								INFO_BITS							
+1	RESERVED								RESERVED							
+2	Code reader Signal Status Information (Device_Status)															
+3																
+4	RESERVED															
+5																
+6	Read Counter Information (COUNTERS)															
…																
...																
+11																
+12	Read Cycle Information (READ CYCLE REPORT)															
+13																
+14																
+15																
+16	Number of characters in Read data (DECODE LENGTH)															
+17																

	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+18	The content of the Read data (DECODE DATA)															
…																
...																
+89																

2 Defining the Variables

Define variables for the data links for the communications area data that is used in EtherNet/IP communications.

These variables use the data types that were defined above in procedure 1.

Variable	Variable Type	Network Publish Attribute	Data Type	Application
EIPOutput	Global variable	Output	S_EIPOutput197	For data links to the Output Area
EIPInput	Global variable	Input	S_EIPInput102	For data links to the Input Area

3 Accessing the Communications Areas from the User Program

The defined variables are used to access the communications areas for the Sensor using the following

Output Area

Signal Name	Variable Name
Trigger	EIPOutput.COMMANDS.Trigger

Input Area

Signal Name	Variable Name
Run Mode	EIPInput.DEVICE_STATUS.Run Mode
Trigger_Acknowledged	EIPInput.DEVICE_STATUS.Trigger_Acknowledged
Decoding	EIPInput.DEVICE_STATUS.Decoding
DataIsReady	EIPInput.DEVICE_STATUS.DataIsReady
Decode_Data	EIPInput.DECODE_DATA

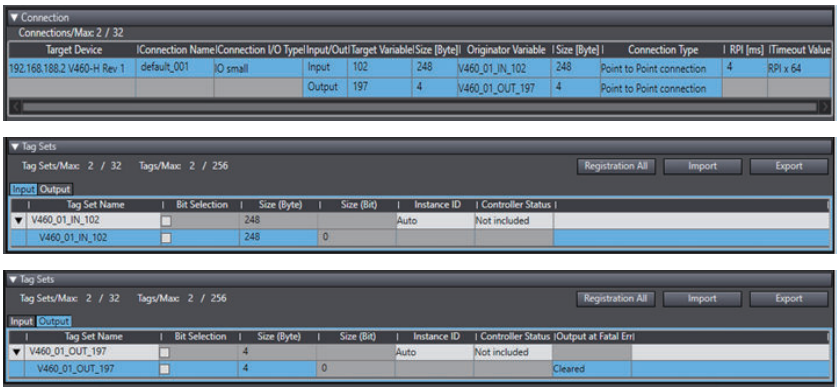
Command Control Example

Here is an example of how Command Control is executed in EtherNet/IP communications between a PLC and the code reader.

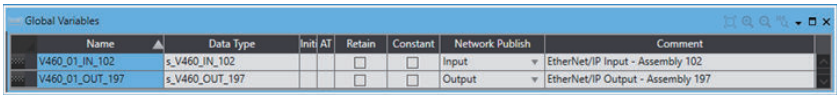
● Read a Code and Store the Read String Output on the PLC

<Example Tag Sets and Connection Settings>

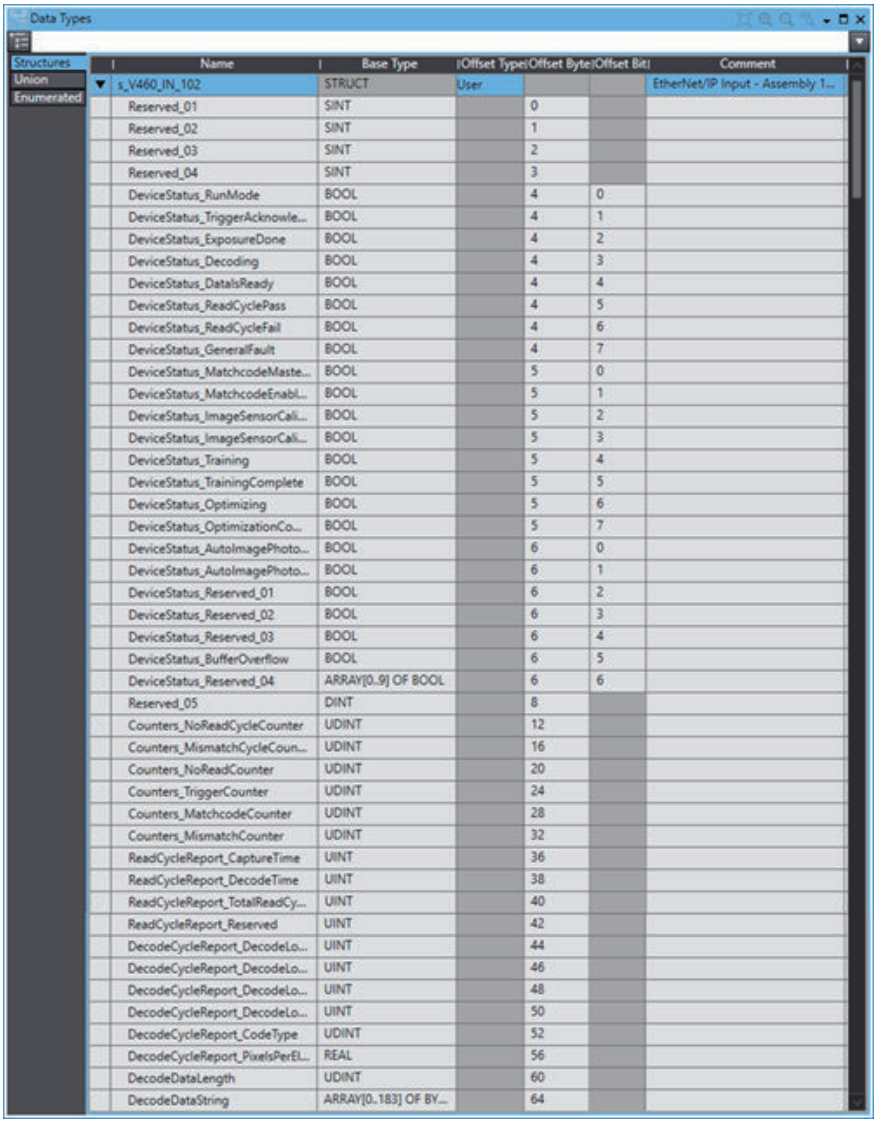
- Input Assembly: MXL/SXL Input (102)
- Output Assembly: Output (197)



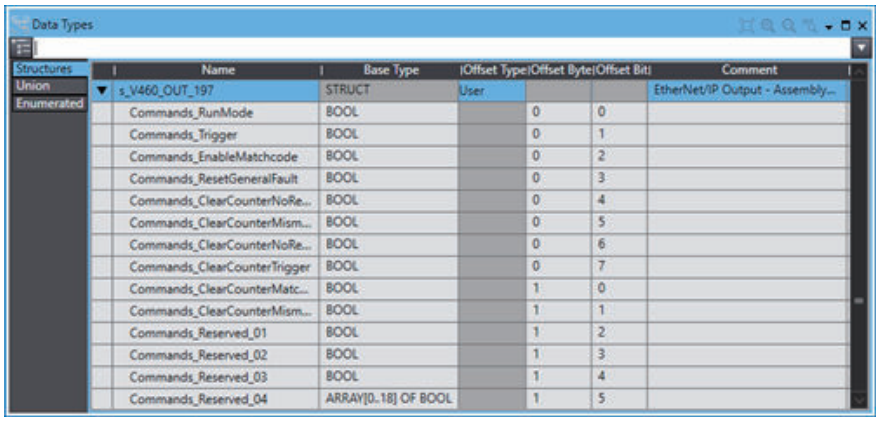
<Example Variables>



<Input Assembly structure (102)>



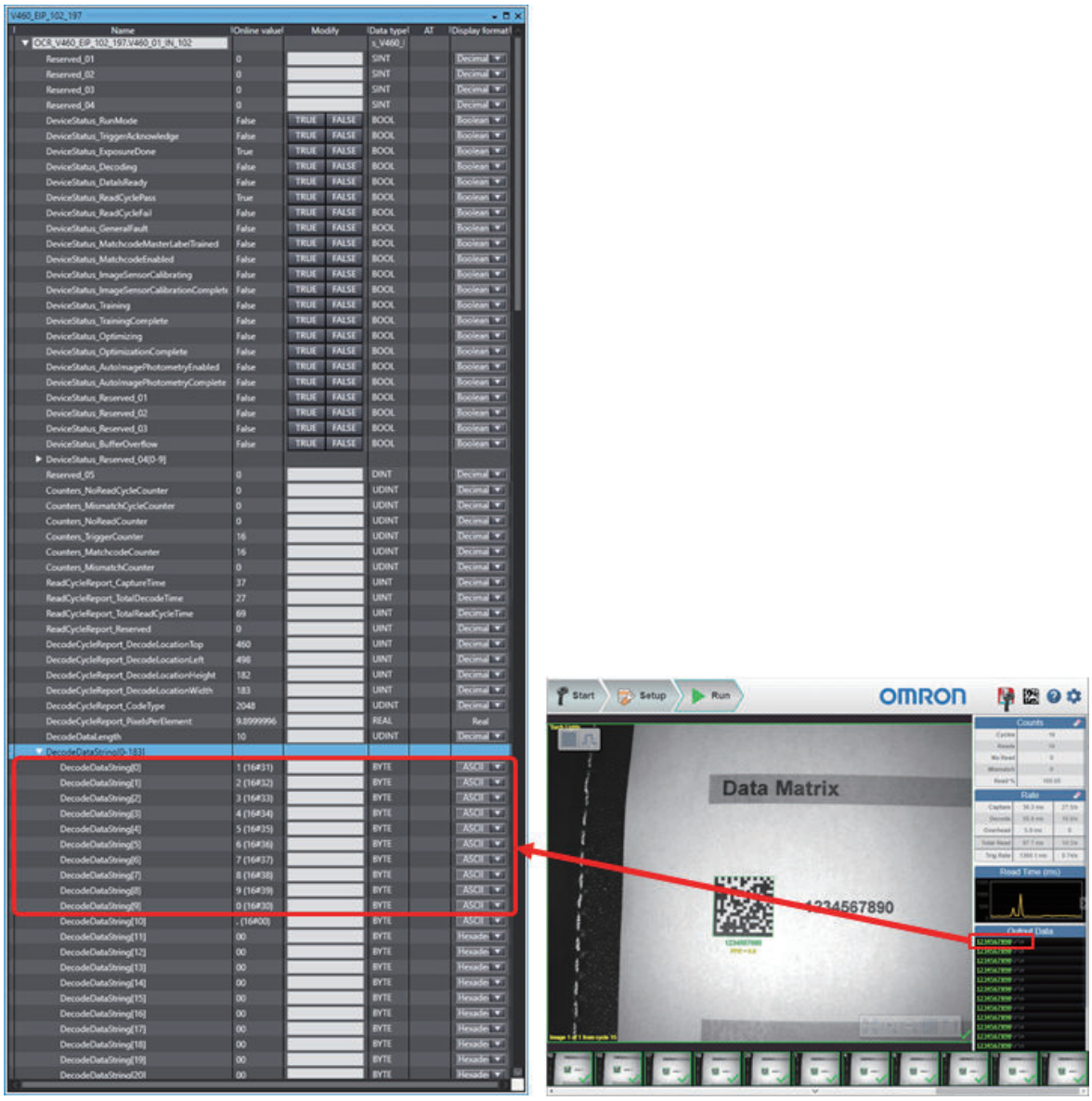
<Output Assembly structure (197)>



Name	Base Type	Offset Type	Offset Byte	Offset Bit	Comment
s_V460_OUT_197	STRUCT	User			EtherNet/IP Output - Assembly...
Commands_RunMode	BOOL		0	0	
Commands_Trigger	BOOL		0	1	
Commands_EnableMatchcode	BOOL		0	2	
Commands_ResetGeneralFault	BOOL		0	3	
Commands_ClearCounterNoRe...	BOOL		0	4	
Commands_ClearCounterMism...	BOOL		0	5	
Commands_ClearCounterNoRe...	BOOL		0	6	
Commands_ClearCounterTrigger	BOOL		0	7	
Commands_ClearCounterMatc...	BOOL		1	0	
Commands_ClearCounterMism...	BOOL		1	1	
Commands_Reserved_01	BOOL		1	2	
Commands_Reserved_02	BOOL		1	3	
Commands_Reserved_03	BOOL		1	4	
Commands_Reserved_04	ARRAY[0..18] OF BOOL		1	5	

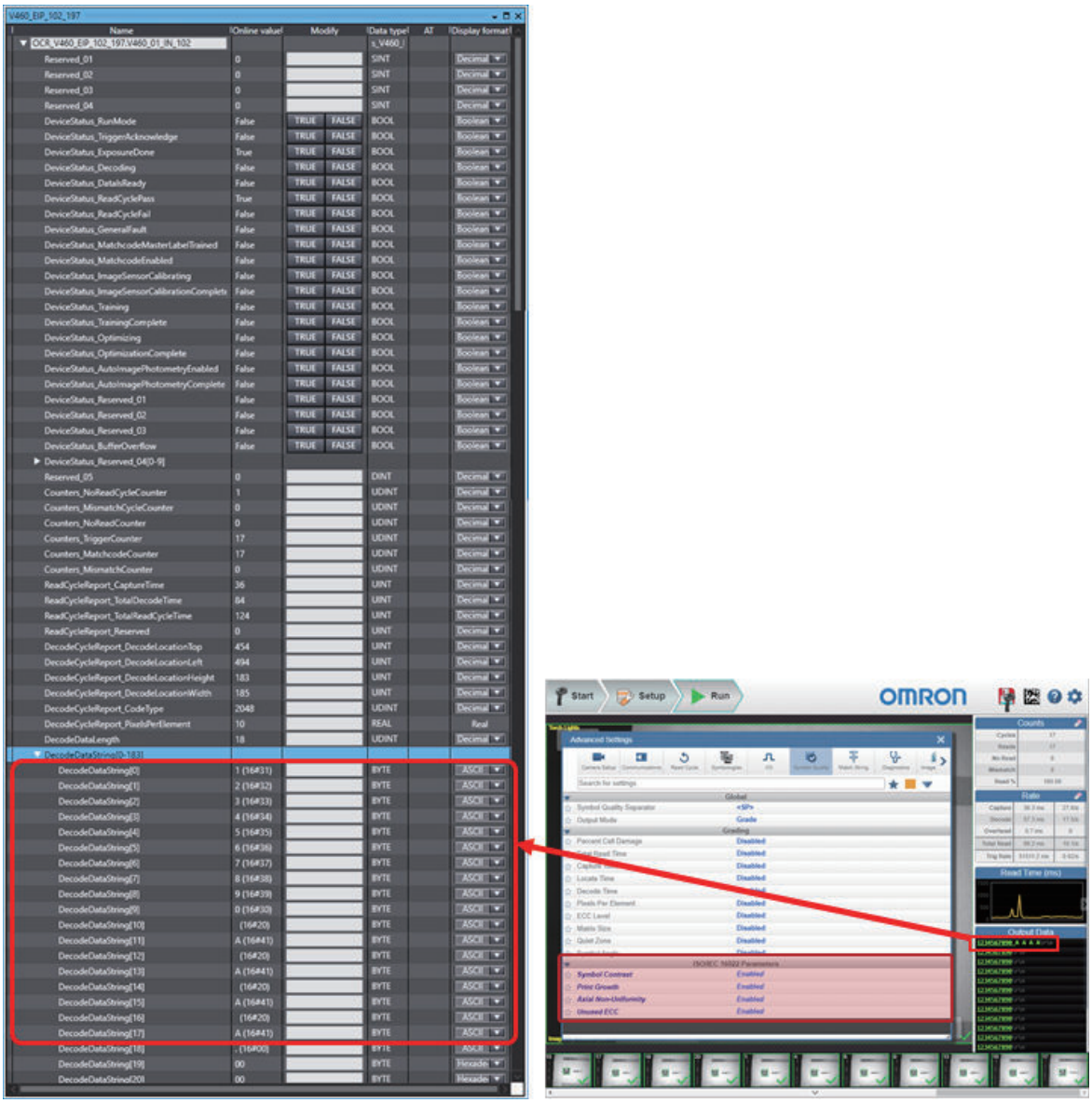
<Example of Data Storage>

- Decode Data String: *1234567890*



<Example of Data Storage if ISO/IEC 16022 Parameters are enabled>

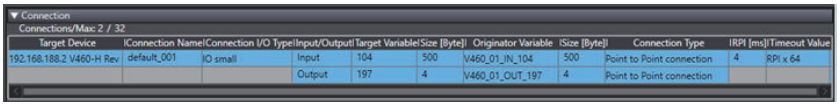
- Decode Data String: *1234567890 A A A A*

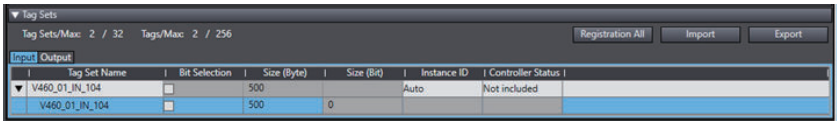


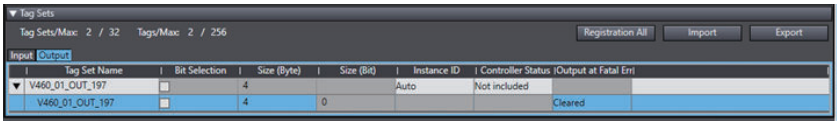
● Read 4 Codes and Store the Read String Output on the PLC

<Example of Tag Sets and Connection Settings>

- Input Assembly: 4 Decode Input (104)
- Output Assembly: Output (197)







<Example of Setting Variables>

Global Variables

Name	Data Type	Init	AT	Retain	Constant	Network Publish	Comment
V460_01_IN_104	s_V460_IN_104			☐	☐	Input	EtherNet/IP Input - Assembly 104
V460_01_OUT_197	s_V460_OUT_197			☐	☐	Output	EtherNet/IP Output - Assembly 197

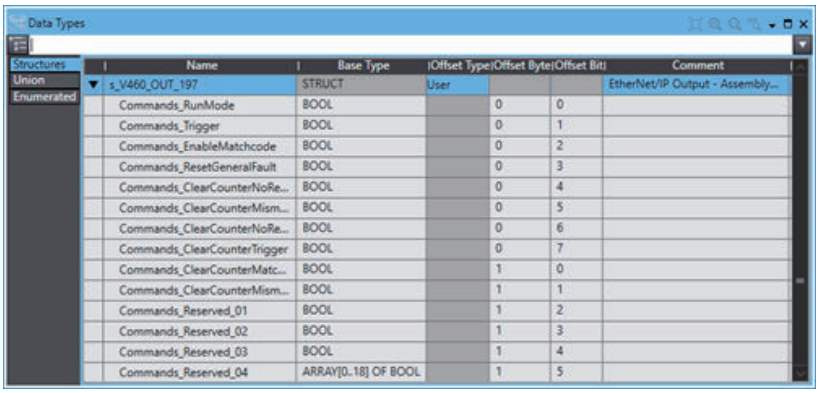
<Input Assembly structure (104)>

Data Types

Structures | Union | Enumerated

Name	Base Type	Offset Type	Offset Byte	Offset Bit	Comment
s_V460_IN_104	STRUCT	User			EtherNet/IP Input - Assembly 104
Reserved_01	SINT		0		
Reserved_02	SINT		1		
Reserved_03	SINT		2		
Reserved_04	SINT		3		
DeviceStatus_RunMode	BOOL		4	0	
DeviceStatus_TriggerAcknowledge	BOOL		4	1	
DeviceStatus_ExposureDone	BOOL		4	2	
DeviceStatus_Decoding	BOOL		4	3	
DeviceStatus_DataIsReady	BOOL		4	4	
DeviceStatus_ReadCyclePass	BOOL		4	5	
DeviceStatus_ReadCycleFail	BOOL		4	6	
DeviceStatus_GeneralFault	BOOL		4	7	
DeviceStatus_MatchcodeMasterLabelTrained	BOOL		5	0	
DeviceStatus_MatchcodeEnabled	BOOL		5	1	
DeviceStatus_ImageSensorCalibrating	BOOL		5	2	
DeviceStatus_ImageSensorCalibrationComplete	BOOL		5	3	
DeviceStatus_Training	BOOL		5	4	
DeviceStatus_TrainingComplete	BOOL		5	5	
DeviceStatus_Optimizing	BOOL		5	6	
DeviceStatus_OptimizationComplete	BOOL		5	7	
DeviceStatus_AutoImagePhotometryEnabled	BOOL		6	0	
DeviceStatus_AutoImagePhotometryComplete	BOOL		6	1	
DeviceStatus_Reserved_01	BOOL		6	2	
DeviceStatus_Reserved_02	BOOL		6	3	
DeviceStatus_Reserved_03	BOOL		6	4	
DeviceStatus_BufferOverflow	BOOL		6	5	
DeviceStatus_Reserved_04	ARRAY[0..9] OF BOOL		6	6	
Reserved_05	DINT		8		
Counters_NoReadCycleCounter	UDINT		12		
Counters_MismatchCycleCounter	UDINT		16		
Counters_NoReadCounter	UDINT		20		
Counters_TriggerCounter	UDINT		24		
Counters_MatchcodeCounter	UDINT		28		
Counters_MismatchCounter	UDINT		32		
ReadCycleReport_CaptureTime	UINT		36		
ReadCycleReport_DecodeTime	UINT		38		
ReadCycleReport_TotalReadCycleTime	UINT		40		
ReadCycleReport_Reserved	UINT		42		
Decode01CycleReport_DecodeLocationTop	UINT		44		
Decode01CycleReport_DecodeLocationLeft	UINT		46		
Decode01CycleReport_DecodeLocationHeight	UINT		48		
Decode01CycleReport_DecodeLocationWidth	UINT		50		
Decode01CycleReport_CodeType	UDINT		52		
Decode01CycleReport_PixelsPerElement	REAL		56		
Decode01DataLength	UDINT		60		
Decode01DataString	ARRAY[0..159] OF BY...		64		
Decode02CycleReport_DecodeLocationTop	UINT		224		
Decode02CycleReport_DecodeLocationLeft	UINT		226		
Decode02CycleReport_DecodeLocationHeight	UINT		228		
Decode02CycleReport_DecodeLocationWidth	UINT		230		
Decode02CycleReport_CodeType	UDINT		232		
Decode02CycleReport_PixelsPerElement	REAL		236		
Decode02DataLength	UDINT		240		
Decode02DataString	ARRAY[0..71] OF BYTE		244		
Decode03CycleReport_DecodeLocationTop	UINT		316		
Decode03CycleReport_DecodeLocationLeft	UINT		318		
Decode03CycleReport_DecodeLocationHeight	UINT		320		
Decode03CycleReport_DecodeLocationWidth	UINT		322		
Decode03CycleReport_CodeType	UDINT		324		
Decode03CycleReport_PixelsPerElement	REAL		328		
Decode03DataLength	UDINT		332		
Decode03DataString	ARRAY[0..71] OF BYTE		336		
Decode04CycleReport_DecodeLocationTop	UINT		408		
Decode04CycleReport_DecodeLocationLeft	UINT		410		
Decode04CycleReport_DecodeLocationHeight	UINT		412		
Decode04CycleReport_DecodeLocationWidth	UINT		414		
Decode04CycleReport_CodeType	UDINT		416		
Decode04CycleReport_PixelsPerElement	REAL		420		
Decode04DataLength	UDINT		424		
Decode04DataString	ARRAY[0..71] OF BYTE		428		

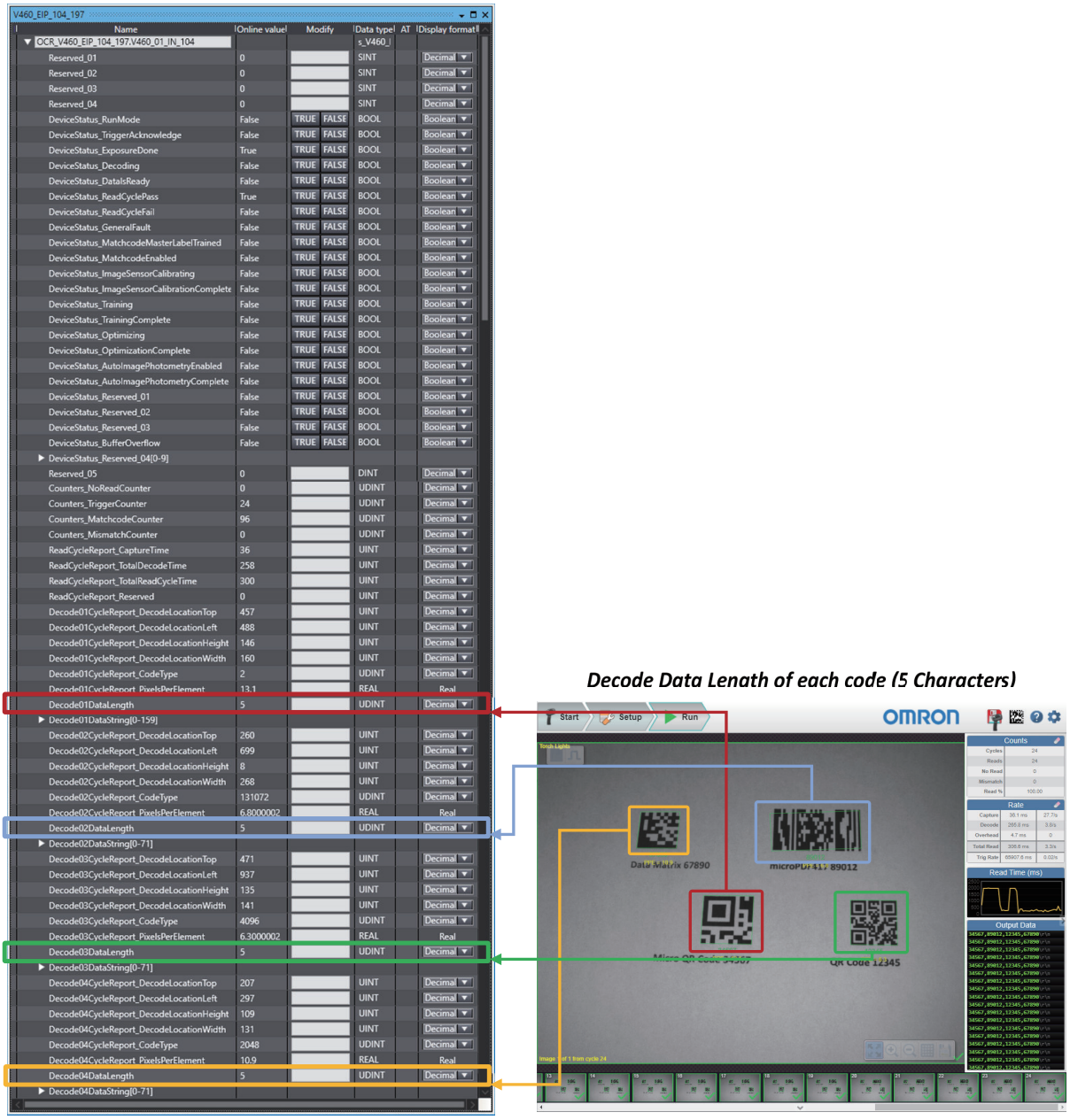
<Output Assembly structure (197)>



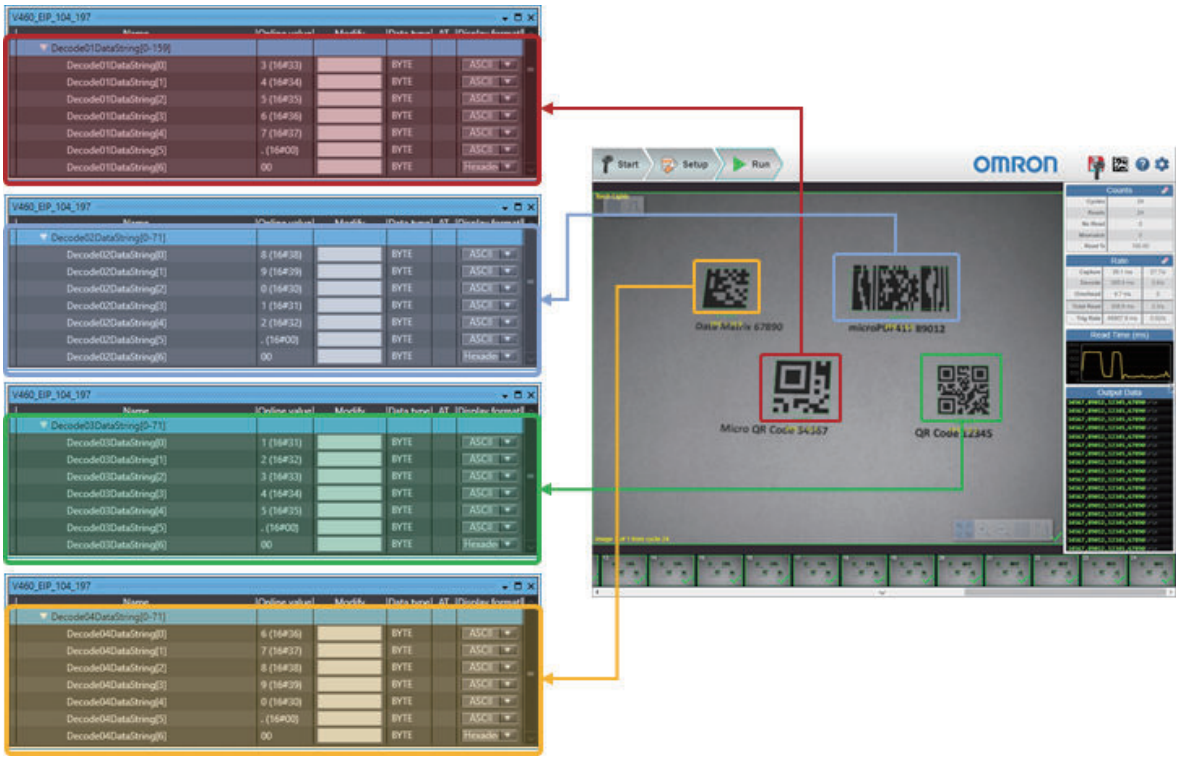
Name	Base Type	Offset Type	Offset Byte	Offset Bit	Comment
s_V460_OUT_197	STRUCT	User			EtherNet/IP Output - Assembly...
Commands_RunMode	BOOL		0	0	
Commands_Trigger	BOOL		0	1	
Commands_EnableMatchcode	BOOL		0	2	
Commands_ResetGeneralFault	BOOL		0	3	
Commands_ClearCounterNoRe...	BOOL		0	4	
Commands_ClearCounterMism...	BOOL		0	5	
Commands_ClearCounterNoRe...	BOOL		0	6	
Commands_ClearCounterTrigger	BOOL		0	7	
Commands_ClearCounterMatc...	BOOL		1	0	
Commands_ClearCounterMism...	BOOL		1	1	
Commands_Reserved_01	BOOL		1	2	
Commands_Reserved_02	BOOL		1	3	
Commands_Reserved_03	BOOL		1	4	
Commands_Reserved_04	ARRAY[0..18] OF BOOL		1	5	

<Example of Data Storage for 4 different codes in 1 image capture>

- Decode Data String 01: *34567* (5 characters, Micro QR Code)
- Decode Data String 02: *89012* (5 characters, MicroPDF417)
- Decode Data String 03: *12345* (5 characters, QR Code)
- Decode Data String 04: *67890* (5 characters, Data Matrix)



- Decode Data String 01: *34567* (5 characters, Micro QR Code)
- Decode Data String 02: *89012* (5 characters, MicroPDF417)
- Decode Data String 03: *12345* (5 characters, QR Code)
- Decode Data String 04: *67890* (5 characters, Data Matrix)



ATTENTION – For **Input Assembly 104 (4 Decode Input)** and **Input Assembly 105 (N Decode Input)**:

If **Format Output is NOT enabled on the reader**, the **DECODE 'X' DATA**, **DECODE 'X' LENGTH** and **DECODE 'X' CYCLE REPORT** will reflect the data of each code, grouped all together, i.e., **DECODE 1 DATA**, **DECODE 1 LENGTH** and **DECODE 1 CYCLE REPORT** related to the same code.

If **Format Output is ENABLED on the reader**, the **DECODE 'X' DATA** and **DECODE 'X' LENGTH** will reflect the format defined by user. However, the **DECODE 'X' CYCLE REPORT** cannot be affected by the format defined by user, keeping its information in the same way as if the **Format Output is NOT enabled on the reader**.

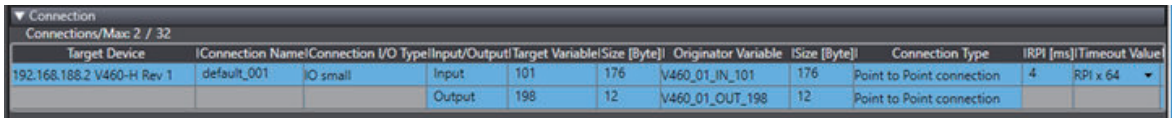
Summarizing:

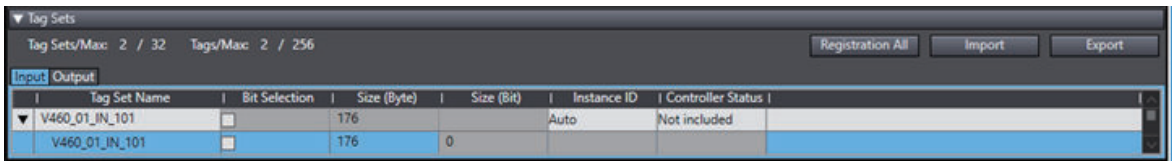
- If **Format Output is ENABLED on the reader**, the **DECODE 'X' CYCLE REPORT** cannot be related to such code.
- If you need them, to guarantee the correct information, enable "Output Coordinates" and "Pixel Per Element" and "Include Symbology Identifier" options (I/O tab) to be added to the **Format Output**, and do consider only the information presented by **DECODE 'X' DATA** and **DECODE 'X' LENGTH**, ignoring the information presented by **DECODE 'X' CYCLE REPORT**.
- This behavior is valid for both **Input Assembly 104 (4 Decode Input)** and **Input Assembly 105 (N Decode Input)**.

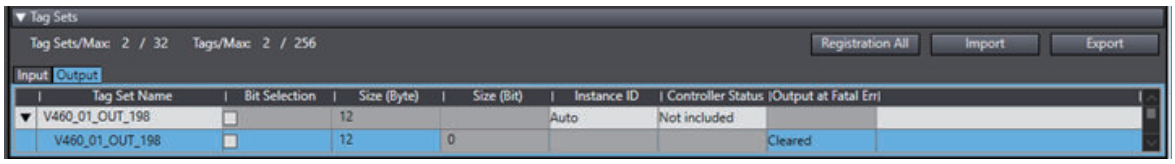
● Using the NEW MASTER Bit in the Output (Legacy) to Register Master Symbol Data

<Example Tag Sets and Connection Settings>

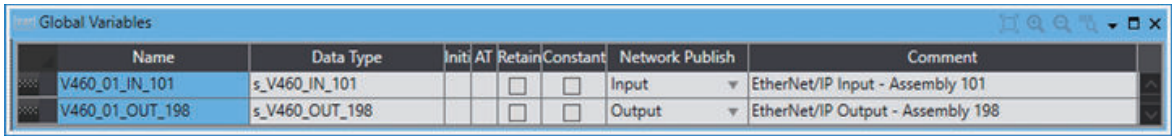
- Input Assembly: Large Input (101)
- Output Assembly: Output (Legacy) (198)







<Example Variables>



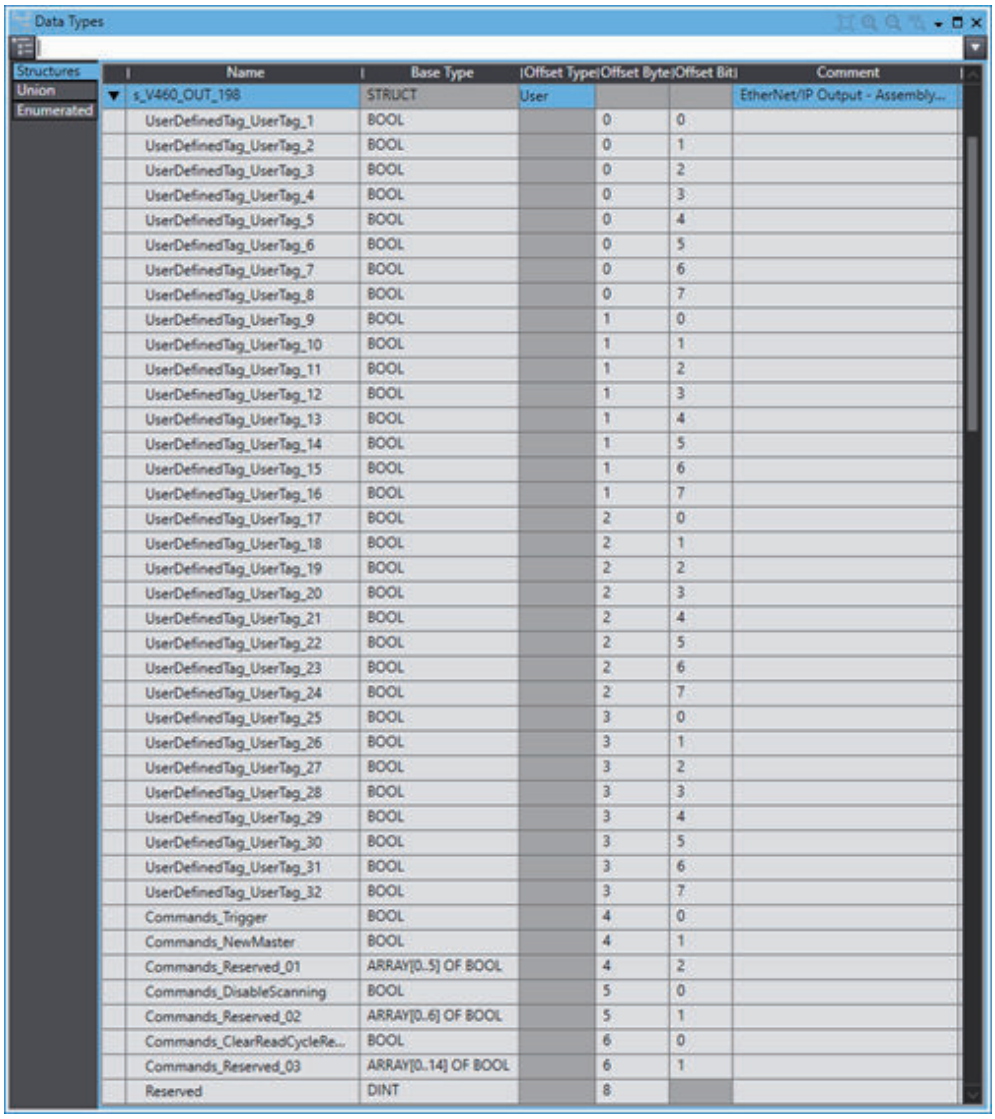
<Input Assembly structure (101)>

Data Types

Structures
Union
Enumerated

Name	Base Type	Offset Type	Offset Byte	Offset Bit	Comment
s_V460_IN_101	STRUCT	User			EtherNet/IP Input - Assembly 101
UserDefinedTagEcho_UserTag_1	BOOL		0	0	
UserDefinedTagEcho_UserTag_2	BOOL		0	1	
UserDefinedTagEcho_UserTag_3	BOOL		0	2	
UserDefinedTagEcho_UserTag_4	BOOL		0	3	
UserDefinedTagEcho_UserTag_5	BOOL		0	4	
UserDefinedTagEcho_UserTag_6	BOOL		0	5	
UserDefinedTagEcho_UserTag_7	BOOL		0	6	
UserDefinedTagEcho_UserTag_8	BOOL		0	7	
UserDefinedTagEcho_UserTag_9	BOOL		1	0	
UserDefinedTagEcho_UserTag_10	BOOL		1	1	
UserDefinedTagEcho_UserTag_11	BOOL		1	2	
UserDefinedTagEcho_UserTag_12	BOOL		1	3	
UserDefinedTagEcho_UserTag_13	BOOL		1	4	
UserDefinedTagEcho_UserTag_14	BOOL		1	5	
UserDefinedTagEcho_UserTag_15	BOOL		1	6	
UserDefinedTagEcho_UserTag_16	BOOL		1	7	
UserDefinedTagEcho_UserTag_17	BOOL		2	0	
UserDefinedTagEcho_UserTag_18	BOOL		2	1	
UserDefinedTagEcho_UserTag_19	BOOL		2	2	
UserDefinedTagEcho_UserTag_20	BOOL		2	3	
UserDefinedTagEcho_UserTag_21	BOOL		2	4	
UserDefinedTagEcho_UserTag_22	BOOL		2	5	
UserDefinedTagEcho_UserTag_23	BOOL		2	6	
UserDefinedTagEcho_UserTag_24	BOOL		2	7	
UserDefinedTagEcho_UserTag_25	BOOL		3	0	
UserDefinedTagEcho_UserTag_26	BOOL		3	1	
UserDefinedTagEcho_UserTag_27	BOOL		3	2	
UserDefinedTagEcho_UserTag_28	BOOL		3	3	
UserDefinedTagEcho_UserTag_29	BOOL		3	4	
UserDefinedTagEcho_UserTag_30	BOOL		3	5	
UserDefinedTagEcho_UserTag_31	BOOL		3	6	
UserDefinedTagEcho_UserTag_32	BOOL		3	7	
CommandEcho_TriggerEcho	BOOL		4	0	
CommandEcho_NewMasterEcho	BOOL		4	1	
CommandEcho_Reserved_01	ARRAY[0..5] OF BOOL		4	2	
CommandEcho_DisableScanni...	BOOL		5	0	
CommandEcho_Reserved_02	ARRAY[0..6] OF BOOL		5	1	
CommandEcho_ClearReadCycl...	BOOL		6	0	
CommandEcho_Reserved_03	ARRAY[0..14] OF BOOL		6	1	
Reserved_01	DINT		8		
Reserved_02	DINT		12		
Reserved_03	DINT		16		
DeviceStatus_Reserved_01	BOOL		20	0	
DeviceStatus_NewMasterRequ...	BOOL		20	1	
DeviceStatus_Reserved_02	ARRAY[0..5] OF BOOL		20	2	
DeviceStatus_ScanningDisabled	BOOL		20	8	
DeviceStatus_Reserved_03	ARRAY[0..6] OF BOOL		20	9	
DeviceStatus_InReadCycle	BOOL		20	16	
DeviceStatus_ActivelyScanning	BOOL		20	17	
DeviceStatus_Reserved_04	ARRAY[0..13] OF BOOL		20	18	
ReadCycleSequenceCounter	UDINT		24		
TriggerCounter	UDINT		28		
DecodeMatchcodeCounter	UDINT		32		
MismatchCounter	UDINT		36		
NoReadCounter	UDINT		40		
DecodeDataLength	UDINT		44		
DecodeDataString	ARRAY[0..127] OF BYTE		48		

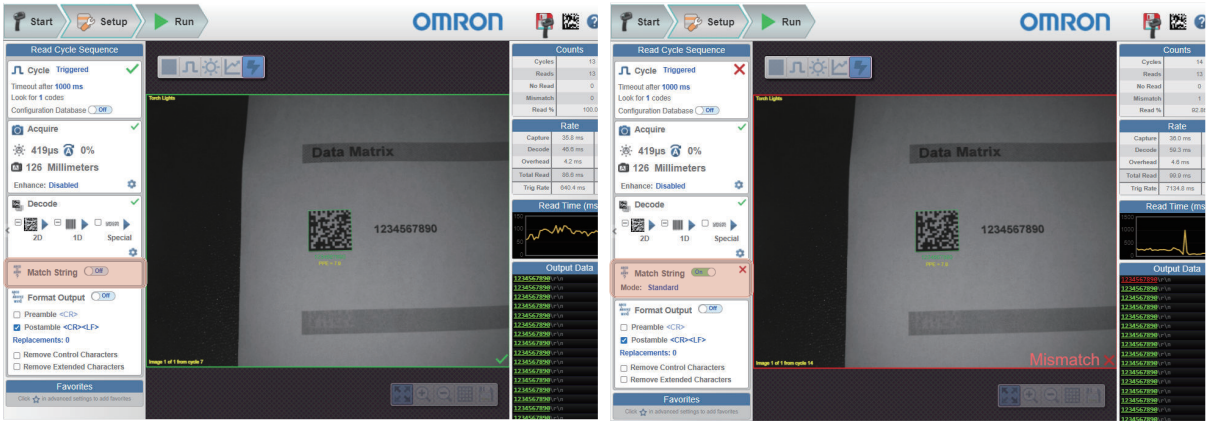
<Output Assembly structure (198)>



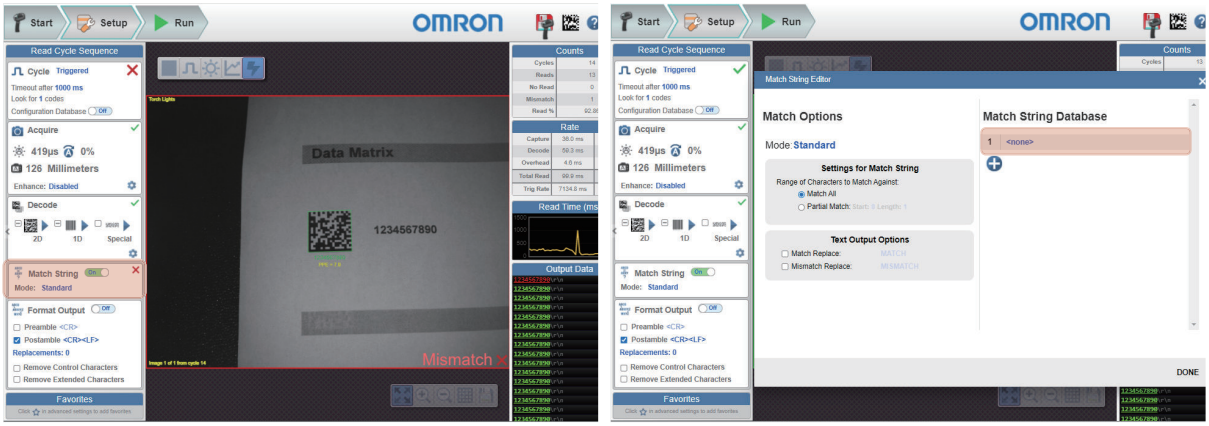
Name	Base Type	Offset Type	Offset Byte	Offset Bit	Comment
s_V460_OUT_198	STRUCT	User			EtherNet/IP Output - Assembly...
UserDefinedTag_UserTag_1	BOOL		0	0	
UserDefinedTag_UserTag_2	BOOL		0	1	
UserDefinedTag_UserTag_3	BOOL		0	2	
UserDefinedTag_UserTag_4	BOOL		0	3	
UserDefinedTag_UserTag_5	BOOL		0	4	
UserDefinedTag_UserTag_6	BOOL		0	5	
UserDefinedTag_UserTag_7	BOOL		0	6	
UserDefinedTag_UserTag_8	BOOL		0	7	
UserDefinedTag_UserTag_9	BOOL		1	0	
UserDefinedTag_UserTag_10	BOOL		1	1	
UserDefinedTag_UserTag_11	BOOL		1	2	
UserDefinedTag_UserTag_12	BOOL		1	3	
UserDefinedTag_UserTag_13	BOOL		1	4	
UserDefinedTag_UserTag_14	BOOL		1	5	
UserDefinedTag_UserTag_15	BOOL		1	6	
UserDefinedTag_UserTag_16	BOOL		1	7	
UserDefinedTag_UserTag_17	BOOL		2	0	
UserDefinedTag_UserTag_18	BOOL		2	1	
UserDefinedTag_UserTag_19	BOOL		2	2	
UserDefinedTag_UserTag_20	BOOL		2	3	
UserDefinedTag_UserTag_21	BOOL		2	4	
UserDefinedTag_UserTag_22	BOOL		2	5	
UserDefinedTag_UserTag_23	BOOL		2	6	
UserDefinedTag_UserTag_24	BOOL		2	7	
UserDefinedTag_UserTag_25	BOOL		3	0	
UserDefinedTag_UserTag_26	BOOL		3	1	
UserDefinedTag_UserTag_27	BOOL		3	2	
UserDefinedTag_UserTag_28	BOOL		3	3	
UserDefinedTag_UserTag_29	BOOL		3	4	
UserDefinedTag_UserTag_30	BOOL		3	5	
UserDefinedTag_UserTag_31	BOOL		3	6	
UserDefinedTag_UserTag_32	BOOL		3	7	
Commands_Trigger	BOOL		4	0	
Commands_NewMaster	BOOL		4	1	
Commands_Reserved_01	ARRAY[0..5] OF BOOL		4	2	
Commands_DisableScanning	BOOL		5	0	
Commands_Reserved_02	ARRAY[0..6] OF BOOL		5	1	
Commands_ClearReadCycleRe...	BOOL		6	0	
Commands_Reserved_03	ARRAY[0..14] OF BOOL		6	1	
Reserved	DINT		8		

Set Matchcode to ON.

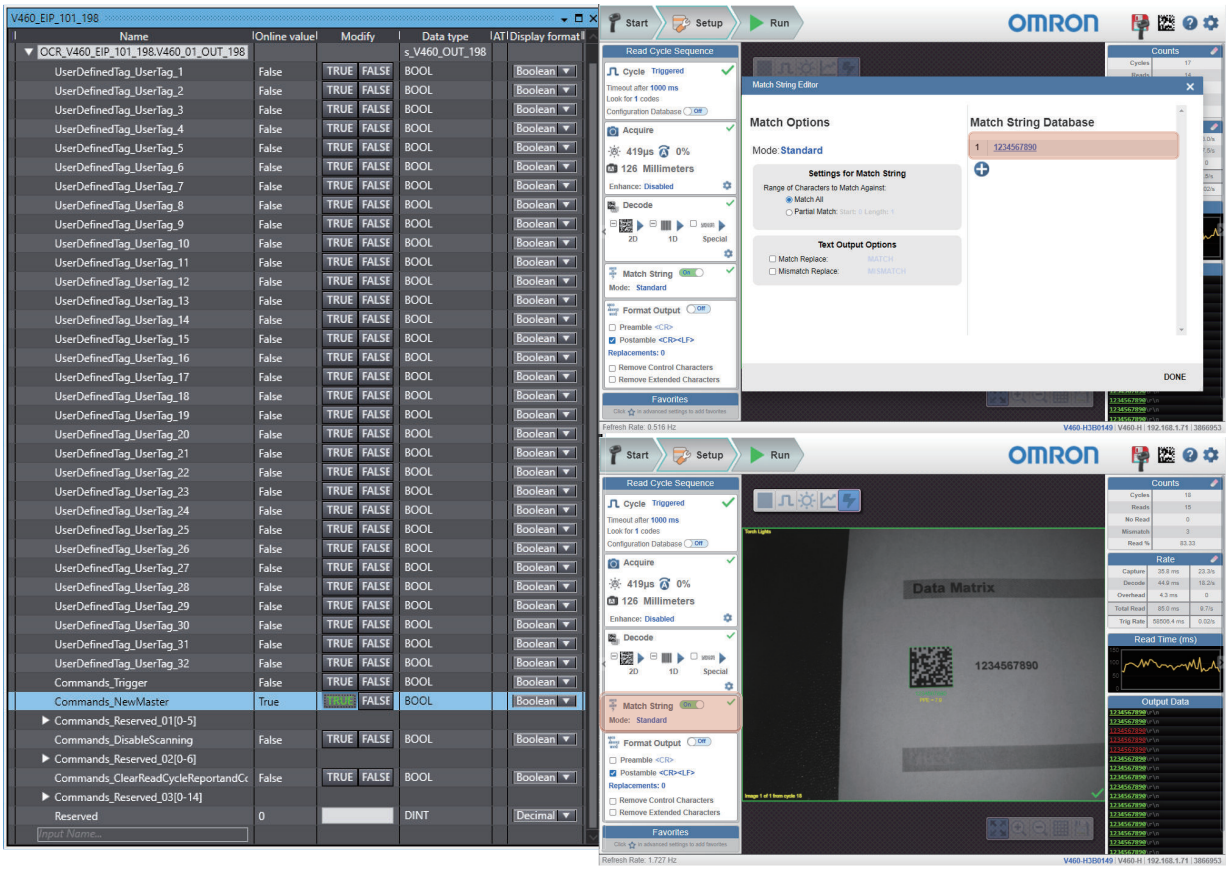
- WebLink$_{HH}$ - **Setup** Screen



With nothing currently registered as the Master Symbol, when a Read is executed with Matchcode ON, the result is *Mismatch*.



By setting the NewMaster bit in the Output Assembly to *1, the next code that is read will be registered as the Master Symbol data.*



2-1-10 Communicating with the Code Reader with EtherNet/IP Message

Serial commands can be executed using EtherNet/IP Message (Explicit) communication.
For more information on Serial commands, please refer to the following.

2-2-5 Controlling Operation from an External Device on page 2-39

Note Any explicit message that causes an action that takes longer than 3 seconds will time out. It is recommended to use implicit messaging in these cases.

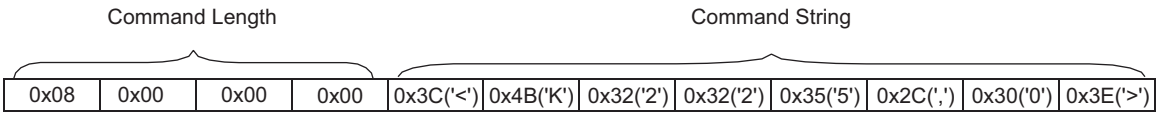
Message Communication Objects have the following structure.

Item	Setting Value
Class ID	104 (0x68 Hex)
Instance ID	1
Attribute ID	1
Service code	69 (0x45 Hex)

EtherNet/IP Message (Explicit) Format

EtherNet/IP messages, both transmitted and received, are comprised of two parts, the Command Length and Command String.

- **Command Length (4 bytes)**
  The total number of characters in the Command String.
- **Command String (256 byte maximum)**
  The ASCII character array of the command sent from the PLC to the code reader.



Command Setting Example

This example shows how to set Message communication command strings.

- For the data that is sent from the PLC to the code reader, set a serial command character string.
- When using K Commands some commands do not provide a Response.
  In other words, there is no data to receive after sending the command.

However, in the case of a Verify setting command like <K225?>, there is a Response so data will be received after this command type is sent.

Please note that multiple commands can be sent in a single transmission, so if the command sent typically doesn't produce a response a verify setting command can be sent in addition as shown in Example 3.

Example 1: The received data string when the data was sent using the *<K225,0>* command.
(Transmitted data) in 12 bytes 0x08 0x00 0x00 0x00 0x3C('<') 0x4B('K') 0x32('2') 0x32('2') 0x35('5') 0x2C(',') 0x30('0') 0x3E('>')
(Received data) None

Example 2: The received data string when the data was sent using the *<K225?>* command.
(Transmitted data) in 11 bytes 0x07 0x00 0x00 0x00 0x3C('<') 0x4B('K') 0x32('2') 0x32('2') 0x35('5') 0x3F('?') 0x3E('>')
(Received data) in 12 bytes 0x08 0x00 0x00 0x00 0x3C('<') 0x4B('K') 0x32('2') 0x32('2') 0x35('5') 0x2C(',') 0x30('0') 0x3E('>')

Example 3: The received data string when the data was sent using the *<K225,0><K225?>* commands.
(Transmitted data) in 19 bytes 0x0F 0x00 0x00 0x00 0x3C('<') 0x4B('K') 0x32('2') 0x32('2') 0x35('5') 0x2C(',') 0x30('0') 0x3E('>') 0x3C('<') 0x4B('K') 0x32('2') 0x32('2') 0x35('5') 0x3F('?') 0x3E('>')
(Received data) in 12 bytes 0x08 0x00 0x00 0x00 0x3C('<') 0x4B('K') 0x32('2') 0x32('2') 0x35('5') 0x2C(',') 0x30('0') 0x3E('>')

2-2 Controlling Operation and Data Output with Serial (TCP)

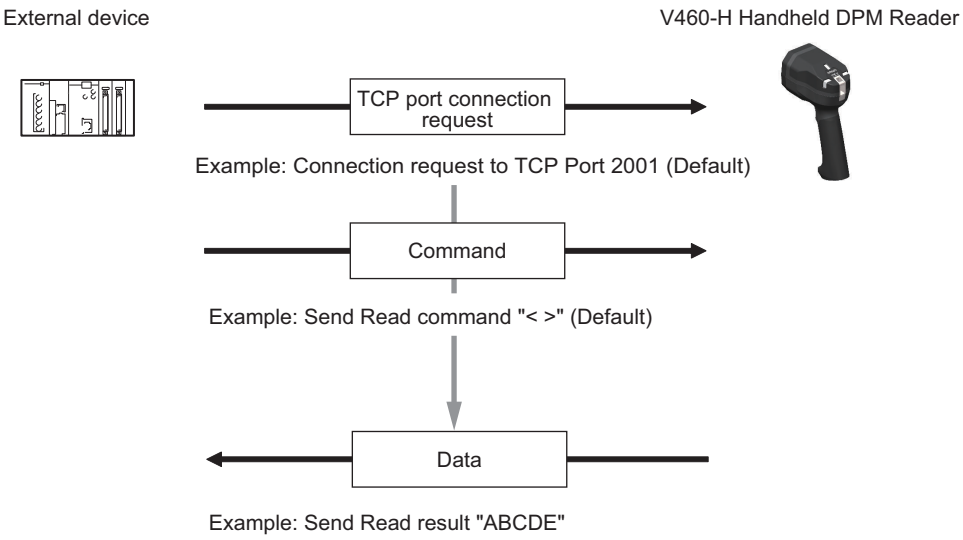
This section explains the communications settings required for using Serial (TCP) communications between the code reader and an external device.

2-2-1 Serial (TCP) Overview

Serial (TCP) conforms to the TCP/IP communication protocols. It can be used with any Ethernet communication equipment compatible with TCP/IP communication protocol. Since this code reader communicates as a TCP server, the external device to be connected must be connected to this code reader as a TCP client. If you intend to use with an Omron PLC, please verify that it supports Socket Services (TCP Client).

2-2-2 Communications Processing Flow

In a system configuration in which the code reader is connected by Serial (TCP) communications to an external device (such as PLC), serial commands can be received and reading results can be output to the external device.
Below is the basic flow for establishing the Serial (TCP) communications, executing a Read command and outputting the Read result.



2-2-3 Communication Settings (Serial (TCP))

Network Settings on the Code Reader

Set the IP address on the code reader to match the network settings of the PLC or other external device.

- WebLink$_{HH}$ - **Setup** - **Gear Icon** - **Advanced Settings** - **Communications** - **Ethernet**

1 Set the **IP Address** and **Subnet mask** according to the network settings of the PLC or other external device.

Setting Item	Setting Value	Description
IP Address	a.b.c.d a: 0 to 255 b: 0 to 255 c: 0 to 255 d: 0 to 255 (Default: 192.168.188.2)	Enter the IP address of the Code Reader
Subnet	a.b.c.d a: 0 to 255 b: 0 to 255 c: 0 to 255 d: 0 to 255 (Default: 255.255.0.0)	Input the subnet mask address.
Gateway	a.b.c.d a: 0 to 255 b: 0 to 255 c: 0 to 255 d: 0 to 255 (Default: 0.0.0.0)	If a Gateway is used, enter the gateway address. If a Gateway is not used, use the default value 0.0.0.0.
IP Address Mode	• Fixed (Default) • DHCP	In Fixed mode, the code reader uses a user-defined IP address. In DHCP mode, the code reader acquires its IP address, subnet, and gateway from the DHCP server. For PLC communication, **Fixed IP Address Mode** is mandatory. DO NOT enable DHCP in this case.
TCP Port 1	1024 to 65536 (Default: 2001)	Enter one of the two TCP port numbers for communication with the code reader over Serial (TCP).
TCP Port 2	1024 to 65536 (Default: 2003)	Enter one of the two TCP port numbers for communication with the code reader over Serial (TCP).

Additional Information

Through the use of two TCP ports at the same time, it is possible for this code reader to communicate over Serial (TCP) with two different external devices.

Change the Command that Executes Read

It is possible to change the command used in Serial communications to execute Read.
There are two types of Read execution commands. One includes a Separator Character (delimiter) <> and the other has no delimiter.

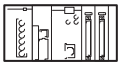
- WebLink$_{HH}$ - **Setup** - **Gear Icon** - **Advanced Settings** - **Read Cycle** - **Serial Trigger (Non-Delimited)**

Setting Item	Setting Value	Description
Serial Trigger Character (Delimited)	ASCII code for 1 character (Default: Space (Hex: 20))	Specifies the command character string used to start a Read. To execute the command, the trigger character must be delimited in brackets <>. This command can only be executed when the Trigger Mode is set in **Read Cycle** - **Trigger** - **Mode** to either *Serial Data* or *Serial Data or External Edge*.
Start Character (Non-Delimited)	The ASCII codes for maximum of 2 characters (Default: NULL (Hex:00))	Specifies the command character string used to start a Read and the command character used to end a Read. The Start command character and the End command character must be different characters. When set it to NULL (Hex:00) it is disabled. The behavior will differ according to the selection made for **Read Cycle** - **Trigger** - **Mode**. • If *External Edge* is selected, the code reader executes Read with the Start command character. An End command character is not necessary. • If *External Level* or *Serial Data and Edge* is selected, the Start trigger character starts a Read cycle and the End command character ends the Read cycle. Even for a Good Read, the Read Cycle does not end until the End command is sent.
Stop Character (Non-Delimited)	The ASCII codes for maximum of 2 characters (Default: NULL (Hex:00))	

● Example Use of Character (Delimited) Command

- Read string: 12345, Character (Delimited): Space, Preamble: None, Postamble: CRLF

External device



	Serial Trigger command		
Character notatiion	<		>
Hex notatiion	3C	20	3E

Code Reader

In Read Cycle	Read result						
Character notatiion	1	2	3	4	5	CR	LF
Hex notatiion	31	32	33	34	35	0D	0A

2-2-4 Setting the Data to Output after a Read

The code reader can be configured so that after a Read is executed, its read results are automatically output to the TCP port it is connected to. Additional information such as print quality grade and code position coordinates can be appended to the Read result output and the format of that output can be modified.

Change the Read Result Output Condition

You can change the conditions by which you will output your Read results.

- WebLink$_{HH}$ - **Setup** - **Gear Icon** - **Advanced Settings** - **I/O** - **Symbol Data Output**

Setting Item	Setting Value	Description
Symbol Data Output	• Disabled • Match • Mismatch • Any Good Read (Default) • Only If All Are Good Reads	• Disabled: Read result is not output. • Match: The Read result is only output when it matches the Master Symbol set in the Matchcode function. • Mismatch: The Read result is only output when it does not match the Master Symbol set in the Matchcode function. • Any Good Read: Read results are output for even just one Good Read. • Only If All Are Good Reads: The Read result is output only when all the symbols specified in the Read Multiple Symbols function are successfully read.
Output Timing	• As Soon As Possible (Default) • End of Read Cycle	• As Soon As Possible: Outputs the Read result immediately on Good Read and ends the Read Cycle. • End of Read Cycle: The Read result is not output until the End of Read Cycle condition is met. The End of Read Cycle condition is set in **Advanced Settings** - **Read Cycle** - **End of Read Cycle**.

Set Output Data for No Read Condition

You can change the data to output when there is a No Read result.

- WebLink$_{HH}$ - **Setup** - **Gear Icon** - **Advanced Settings** - **I/O** - **No Read Message**

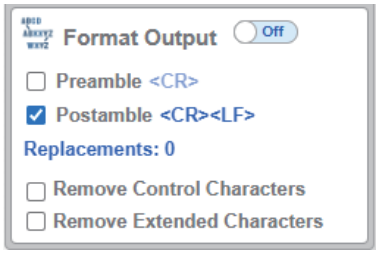
Setting Item	Setting Value	Description
No Read Message	• Enabled (Default) • Disabled	• Enabled: A message is output when there is a No Read. However, if the Trigger Mode set in *Read Cycle is Continuous Read,* no message is output regardless of this setting. • Disabled: No message is output for a No Read.
Message	NOREAD (Default)	Set the message to output when there is a No Read. You can set up to 64 ASCII characters.

Header and Footer Settings

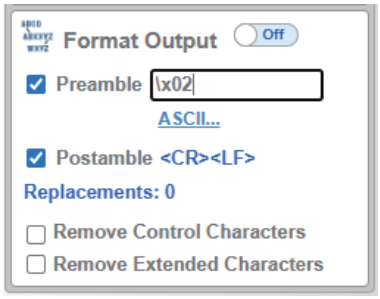
You can change the Header (Preamble) and Footer (Postamble) that precedes and follows the Read string.

- WebLink$_{HH}$ - **Setup** - **Format Output**

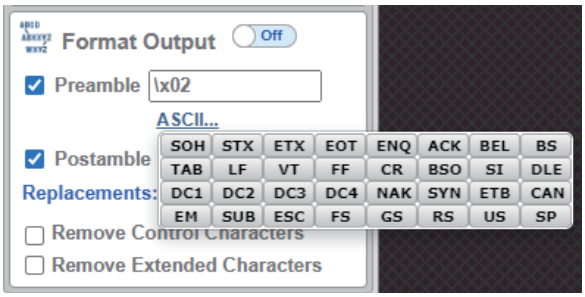
1 Set the Header (Preamble) as needed.
In **Format Output**, check the box for **Preamble** to add a header to the Read result.



To edit the characters used in the header, click on the blue text to the right of **Preamble**. Characters can be entered from the keyboard in to the Text Input Box.



If you want to use a Control Character as the input, select **ASCII...** below the text input box. Control characters will be displayed and can be selected from here.



2 Set the Footer (Postamble) as needed.
The procedure for setting it is the same as that for the Header (Preamble).

Additional Information

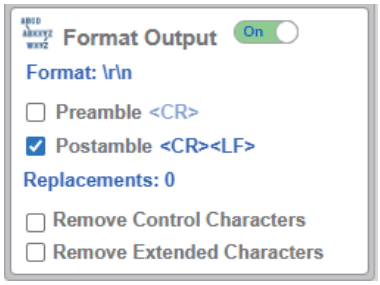
The Header and Footer can also be set in **Advanced Settings** - **Communications** - **Preamble** / **Postamble**.

Setting the Format of Data Output

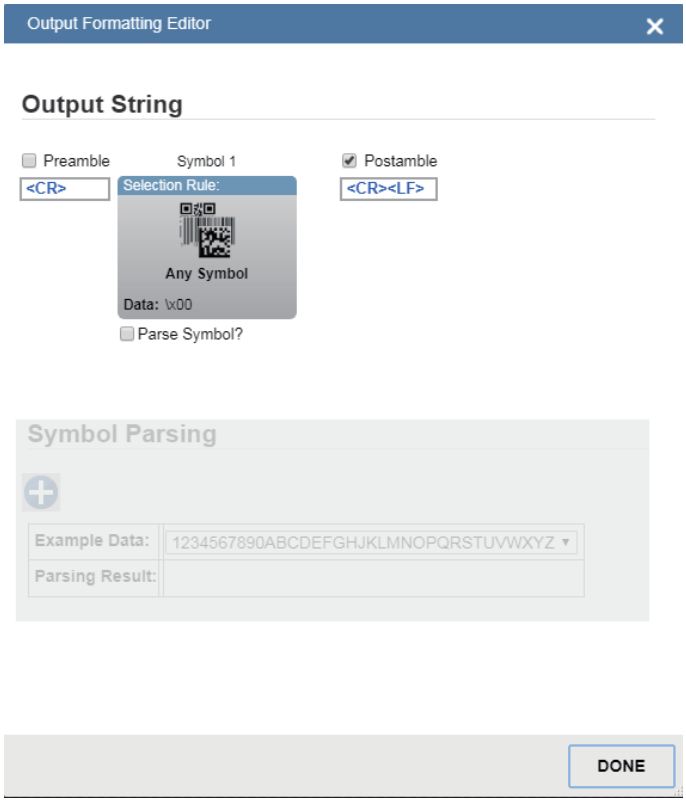
You can format the Read data you wish to output, for example, by specifying the number of characters read from a code symbol to output and appending a fixed character string to the output.

- WebLink$_{HH}$ - **Setup** - **Format Output**

1 Change the format of the output as needed.
Turn **Format Output** ON.

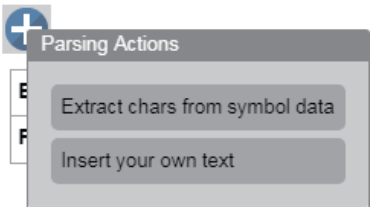


2 The Output Formatting Editor screen opens.
Click on the **Format:** text in blue below **Format Output** to open the Output Formatting Editor.



3 **Parse Symbol?**Check this box.

Press the appropriate button to select either **Extract chars from symbol data**, or **Insert your own text**.



4 If you want to specify the range (number) of characters from the read character string to output, select **Extract chars from symbol data**.
Enter the range for the number of characters to output. In the example setting below, a string length of 1 to 4 characters is output.

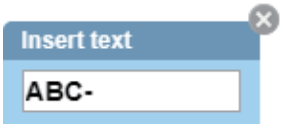
To apply this setting, click the  button.

5 If you want to insert a fixed character string in to the read (decoded) character string to output, select **Insert your own text**.
The default text in the field is */r*. If you click on it, a Text input box will appear so that you can input text from your keyboard. In the example below, *ABC-* is set for the 4 characters.

If you want to use a Control Character as the input, select **ASCII...** below the text input box. Control characters will be displayed and can be selected from here.

When Input is complete, press Enter on the keyboard.

6 To delete the formatting you set for the output, place the cursor on it and click the X button displayed on the upper right.



7 Click the **Done** button. It will close the Output Formatting Editor screen.

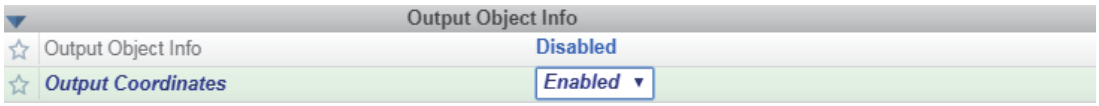
How to Append Additional Symbol Information

Additional information such as print quality grade and code position coordinates can be appended to the Read result output.

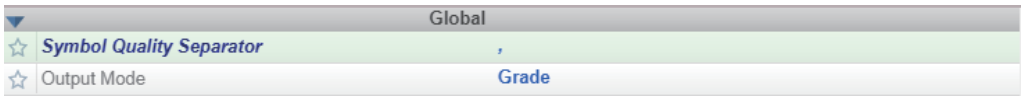
● Outputting a Code Symbol's Position Information

- WebLink$_{HH}$ - **Setup** - **Gear Icon** - **Advanced Settings** - **I/O** - **Output Object Info**

1 Enable **Output Coordinates**.



2 If necessary, change the Separator Character.
In WebLink$_{HH}$ - **Setup** - **Gear Icon** - **Advanced Settings** - **Symbol Quality** - **Global**, enter a character for **Symbol Quality Separator**. Below is an example where , is used as the Separator Character.



3 The position coordinates of the code symbol is output appended to the Read result.
The following example shows the code symbol's position coordinates appended to the Read string *ABCDE*.
ABCDE,(0867,0708)(0867,0708)(1741,0673)(1741,0673)

● Additional Symbol Information That Can Be Appended

Additional information	Setting to adjust (WebLink$_{HH}$ - Advanced Settings Menu)	Description	Example Output (For Read string *ABCDE*.) The delimiter character is a , <comma>.)	Output Order
Symbol Identifier	**I/O - Symbol Data Output**	A (3 character) Symbol Identifier indicating the type of the read symbol is put in front of its Read string.	]dlABCDE	Put in front of the Read string
Decodes per Trigger	**I/O - Decodes per Trigger Output**	Outputs the number of Good Read in Read Cycle.	ABCDE,00002	1
Configuration Data Identifier	**I/O - Database Identifier Output**	Outputs the Index Number of the Configuration Database used to get a Good Read.	ABCDE,DB01	2
Frame Number	**I/O - Output Object Info**	Outputs the Frames number (number of images) that were needed to get a Good Read result. The output is a 3 digit number.	ABCDE,F010	3

Additional information	Setting to adjust (WebLink$_{HH}$ - Advanced Settings Menu)	Description	Example Output (For Read string *ABCDE*.) The delimiter character is a , <comma>.)	Output Order
Code Position Coordinates	**I/O** - **Output Object Info**	Outputs the coordinates of the four vertices of the read symbol in pixels.	ABCDE, (0032,0040) (0287,0056) (0287,0279) (0048,0271)	4
Print Quality (ISO/IEC 16022)	**Symbol Quality** - **ISO/IEC 16022 Parameters**	Outputs the DataMatrix Symbol Quality Grade defined by ISO/IEC 16022.*1	ABCDE,B,A,A,A	5
Print Quality (Omron Microscan)	**Symbol Quality** - **Omron Microscan Parameters**	Outputs the Omron Microscan Proprietary Symbol Quality Grade.*1	ABCDE,000,092,143,091,001,14.3,200,16X16,PASS,349	6
Read Time	**I/O** - **Read Duration Output**	Outputs the Read Duration time in milliseconds.	ABCDE,100	10
Read Cycle ID	**I/O** - **Output Cycle ID**	The Output Cycle ID number (number of Reads executed) is output in hexadecimal format.	ABCDE,CycleID=0x8	11

*1. For more information on Symbol Quality Grade, please refer to *V460-H Industrial Handheld DPM Reader User Manual (Z461 (84-9000460-02)) – Symbol Quality Grade*.

Additional Information

When Multiple Symbol Reading is enabled, the Output Order is *read character string of symbol 1, additional information for symbol 1, read character string for symbol 2, additional information for symbol 2, and so on*.

2-2-5 Controlling Operation from an External Device

The code reader can be controlled, have its settings viewed and changed from an external device with the use of serial commands.

The serial commands of this code reader are divided broadly in to two different types.

- **Serial Configuration Commands (K Commands)**
  Commands to change settings on this code reader.
- **Serial Utility Commands**
  Commands used to test Read Rate, get code reader status and control automatic adjustments.

Serial Command Format

Explanation of how commands are formatted in Serial communication.

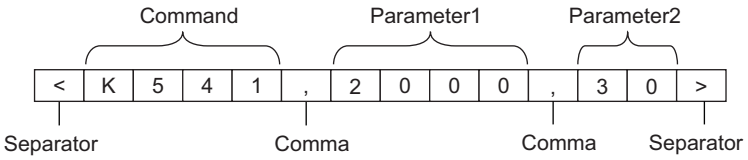
● Common Command Format for Serial Configuration Commands and Serial Utility Commands

- Enclose the commands in brackets "< >".
- Characters used in commands and data are case-sensitive. Use either upper-case, or lower-case characters as required.

- Serial commands can be linked together in a chain. For example, the following command sets Trigger Mode to External Trigger/Edge, sets the End of Read Cycle condition to New Trigger, and saves the setting.
  <K200,3> <K220,1> <Z>

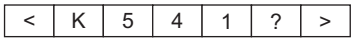
● Serial Configuration Command (K Commands) Format

- The K Commands consist of the letter K, followed by a three digit number and comma-separated parameters as shown below.

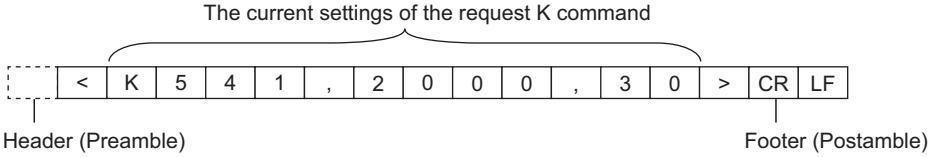


- Some K Commands can change multiple parameters. For those, if the final parameter does not need to be changed, it can be omitted.
  For example, when using the K Command <K541> which is used for changing both Exposure time and Gain, if you only need to change the Exposure time, it can be entered as follows.
  <K541,1000>
- If the parameter that does not need to be changed is not the last in sequence, only the comma delimiter for it is necessary.
  For example, when using the K Command <K541> which is used for changing both Exposure time and Gain, if you only need to change the Gain, it can be entered as follows.
  <K541,,30>
- If any characters other than numeric values, such as Control characters, need to be used in the command, they must be entered in hexidecimal format. If you need to include the characters <,>, comma (,) as parameters, enter them as their hexadecimal value. To enter a hexadecimal value as a parameter, add lowercase h immediately after the K command.
  For example, to set CR (hexadecimal value: 0D) to the footer (postamble), you can enter it as follows.
  <K142h,,0D>
- By default, there is no Response when a K Command is used. To query a current state on the code reader, use a <Knnn?> Command.
  For example, the following is the command to query the current Exposure time and Gain settings and its Response.

- Status Request command



- Response



Additional Information

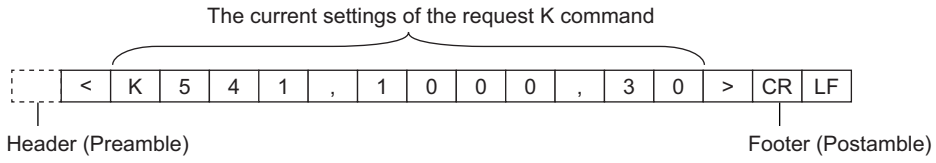
The Response includes a Header (Preamble) and Footer (Postamble). The defaults for these are Header: *None* and Footer: *CRLF* (hexadecimal: 0D0A).

- When the Serial Verification function is enabled (by default: disabled), the current setting status is returned as the response to the K command. If you want to confirm that the K command was applied correctly, please enable the Serial Verification function.

- K command (when the Serial Verification function is enabled)

<	K	5	4	1	,	1	0	0	0	>

- Response



Additional Information

The Response includes a Header (Preamble) and Footer (Postamble).

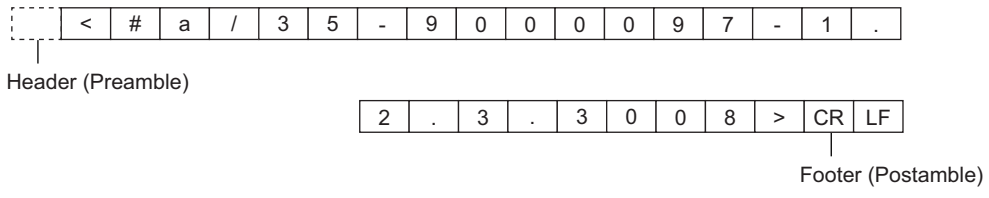
● Command Format for the Serial Utility

- For the Serial Utility commands, there are commands for which there is a response and commands for which there is no response. The format of the response differs with each command.

- Application version Request command

<	#	a	>

- Response



- Clear Trigger counters

<	U	>

- Response

None

2-2-6 Serial Command List

A list of the supported Serial commands.

Category	Command	Description	Response Data Example (For Read string *ABCDE*.)
Setting change (K Command)	<Knnn, Parameter> nnn: Three-digit number of each K command	Commands to change settings on the code reader. Refer to the *V460-H Industrial Handheld DPM Reader User Manual (Z461 (84-9000460-02))* for additional information on K Commands.	None (If the Serial Verification function is enabled, the Response data will be the same as that for the <Knnn?> command.)
Execute a Read	User Defined (Default: < >)	Command (Delimited) to execute a Read. (Reference: *Change the Command that Executes Read* on page 2-33)	ABCDE
	User Defined (Default: Disabled)	The Start Character (Non-Delimited) (Reference: *Change the Command that Executes Read* on page 2-33)	ABCDE
	User Defined (Default: Disabled)	The Stop Character (Non-Delimited) (Reference: *Change the Command that Executes Read* on page 2-33)	ABCDE
Read Test	<C>	Tests the number of Decodes per second. The Response data output is the number of Decodes per second and the Read character string.	5 Decodes / Sec ABCDE *1 (By this you can see 5 Good Read in 1 second)
	<Cp>	Tests the Read Rate (%). The Response data output is the percentage of Good Read per 100 Reads and the Read character string.	95% ABCDE *1 (By this you can see 95 Good Reads out of 100 Reads.)
	<J>	Ends the Read test.	None
Auto-adjust	<@CAL>	Automatically adjusts the settings for Exposure, Focus Position and Symbol Type. Calibration PASSED is output as the Response data from halfway through the progress and when calibration completes successfully. If calibration fails, the message, Calibration FAILED will be output.	Prog | Exposure Gain Brightness 2 | 5764 33 24 100 | 6011 33 37 Calibration PASSED. *1
Train	<TRAIN>	Start the Train operation. Trains with the next Symbol read. When reading the same code symbols, using Train can make Reading results more stable.	None
	<UNTRAIN>	Release the Train operation.	None
	<TRAIN?>	Verifies the Train status. Depending on the status of the Train, the Response data will be one of the following. • <TRAIN,0>: Default, Train not done • <TRAIN,1>: Train in progress • <TRAIN,2>: Training of Symbol complete	<TRAIN,2>
Optimization	<OPT>	Starts Optimization. Optimization using the next Symbol read. When reading the same code symbols, using Optimization can make the Reading speed faster.	None
	<UNOPT>	Releases Optimization.	None

Category	Command	Description	Response Data Example (For Read string *ABCDE*.)
	<OPT?>	Confirm the Optimization status. Depending on the Optimization status, the Response data will be one of the following. • <OPT,0>: Default, No Optimization • <OPT,1>: Optimization in Progress • <OPT,2>: Optimization of Symbol is complete	<OPT,0>
	<?>	Gets the hexidecimal number showing the code reader status. Information for error conditions on the code reader and Read Cycle status can be obtained. For more detailed information, please refer to V460-H Industrial Handheld DPM Reader User Manual (Z461 (84-9000460-02)) - *Appendices A-16 Utilities*.	<?/02> (You can see there is currently no error condition on the code reader.)
	<K?>	Queries the settings status of all K Commands.	Settings status of all K Commands.
	<K??>	Gets the description of all K Command parameters.	Description of all K Command parameters.
	<K?#>	Gets the parameter ranges of all K Commands.	Parameter ranges of all K Commands.
	<Knnn?> (nnn: Three-digit number of each K command)	Queries the setting status of the specified K Command.	<K541,2000,30> (Example response to <K541??> Exposure: 2000us, Gain: 30)
	<Knnn??> (nnn: Three-digit number of each K command)	Queries the parameter description of the specified K Command.	<K541??,Exposure,Gain> (Example Response data for <K541??> You can see that Parameter 1 is Exposure and Parameter 2 is Gain.)
	<Knnn?#> (nnn: Three-digit number of each K command)	Gets the parameter range of the specified K Command.	<K541?#,Value 25-100000:Def=2500,Value 0-100:Def=33> (Example Response data for <K541?#> You can see that the setting range of Parameter 1 is 25 to 100000 with a default value of 2500, while the setting range of Parameter 2 is 0 to 100 with a default value of 33.)
	<Knnn?*> (nnn: Three-digit number of each K command)	Gets the same Response data as when the <Knnn?> command, <Knnn??> command, or <Knnn?#> command is executed.	<K541,2000,30> <K541??,Exposure,Gain> <K541?#,Value 25-100000:Def=2500,Value 0-100:Def=33> (Example Response data for <K541?*>.)
Device Control	<l1>	Turn ON the target pattern (blue LEDs).	None
	<l0>	Turn OFF the target pattern (blue LEDs).	None

Category	Command	Description	Response Data Example (For Read string *ABCDE*.)
	<I>	Disables Read Cycle. While Read Cycle is Disabled, it cannot accept a trigger.	None
	<H>	Enables Read Cycle.	None
Counters and Counter Resets	<q>	Gets the number of No Reads in the Read Cycle. The Response data output is q/ followed by a 9 digit value for the number of No Read.	<q/000000005>
	<q0>	Clears the number of No Reads in the Read Cycle.	None
	<$>	Gets the Mismatch Count. The Response data output is $/ followed by a 9 digit value for the number of Mismatch.	<$/000000002>
	<$0>	Clears the Mismatch Count.	None
	<N>	Gets the number of No Reads. The Response data output is N/ followed by a 9 digit value for the number of No Reads	<N/000000005>
	<O>	Clears the Mismatch Count.	None
	<T>	Gets the Trigger Input Count. The Response data output is T/ followed by a 9 digit value for the number of Triggers.	<T/000000010>
	<U>	Clears the Trigger Input Count.	None
	<V>	Gets the Match Count when the Matchcode function is used. The Response data output is V/ followed by a 9 digit value for the number of Matched strings.	<V/000000010>
	<W>	Clears the Match Count.	None
	<X>	Gets the Mismatch Count when the Matchcode function is used. The Response data output is X/ followed by a 9 digit value for the number of Mismatched strings.	
	<Y>	Clears the Mismatch Count.	None
Confirm Firmware Version	<#>	Queries all the firmware version information.	<#b/35-9000033-122.3021><#a/35-9000097-1.2.3.3008><#w/30-9000079-1.2.3.3006><#p/N/A><#d/35-xxxxxx-x.x.x.xxxx>
	<#a>	Queries the version information of application software.	<#a/35-9000097-1.2.3.3008>
	<#b>	Queries the Boot Software Version information.	<#b/35-9000033-122.3021>
	<#w>	Queries the $WebLink_{HH}$ version.	<#w/30-9000079-1.2.3.3006>
	<!>	Queries the Application software checksum and Boot Software checksum.	<!b/38B7><!a/9555>
	<!a>	Queries the Application software checksum.	<!a/9555>
	<!b>	Queries the Boot Software checksum.	<!b/38B7>

Category	Command	Description	Response Data Example (For Read string *ABCDE*.)
Save for Power-on, Re-initialize and Restart	<Z>	Saves current settings to the code reader and restarts it.	<A?/0>
	<Zc>	Saves the current setting as the Customer default setting on the code reader and restarts.	<A?/0>
	<Zrc>	Restores the code reader settings to the customer default and restarts the code reader.	<A?/0>
	<Zrd>	Resets the code reader to its factory default settings (excluding communication settings and user-defined names) and restarts.	<A?/0>
	<Zrdall>	Resets the code reader to its factory default settings and restarts.	<A?/0>1>
	<A>	Restarts the code reader with its current settings.	<A?/0>
	<Ard>	Resets the code reader to its factory default settings (excluding communication settings and user-defined names) and restarts.	<A?/0>
	<Arp>	Restores the code reader settings to the previously saved state and restarts.	<A?/0>
	<Arc>	Restores the code reader settings to the customer default and restarts.	<A?/0>
Master Database	<G>	Sets the database number to be registered in the Master database to 1.	None
	<Gn> n: Master Database Index Number	Sets the database number to be registered in the Master database to n.	<NEWM/01> (The data for the next Good Read is registered in Master database 1.)
	<NEWM>	Queries the database number to register. <NEWM/00> is returned if there is no database yet specified to be registered.	None
Barcode Configuration	<BCCFG>	Transitions to the Barcode Configuration Mode in which Read can be performed on a Data-Matrix converted to data with a K Command. For more information please refer to *V460-H Industrial Handheld DPM Reader User Manual (Z461 (84-9000460-02)) - Appendices A-16 Utilities*.	None

*1. The Headers and Footers are not dependent on code reader settings. They are Header: None and Footer: CRLF.

3

Controlling Operation and Data Output with PROFINET

This section describes the procedures for connecting the V460-H Series Reader to the NJ/NX Series Machine Automation Controller (hereinafter referred to as Controller) via Profinet IO), and for verifying the device connections. After following the configurations in this section, the user will be able to view PROFINET input and output module data, make changes to the output module, and verify those changes at the input module. The examples in this section do not contain any PLC programming, custom data structures, or setup, beyond connecting the input and output modules. It is the user's responsibility to program the controller once data access to the V460-H has been established.

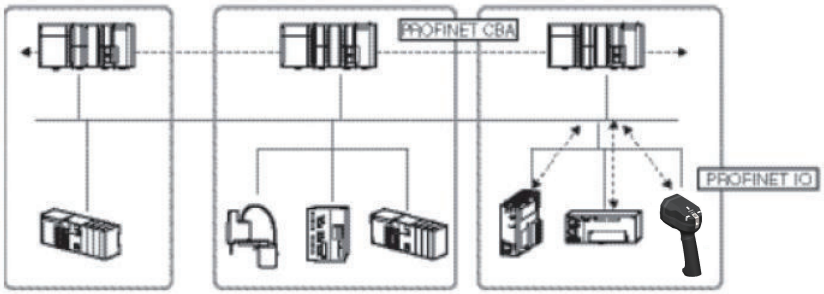
3-1 Overview of PROFINET

PROFINET is a network for industrial use that applies industrial Ethernet (100 Mbps, full duplex) to PROFIBUS DP.
PROFINET is an open standard that is managed by PI (PROFIBUS and PROFINET International) and is used in a variety of types of industrial equipment. Because PROFINET uses standard Ethernet technology, a variety of general-purpose Ethernet devices can be included in the network.
This section provides an overview sufficient to use this code reader with PROFINET.
Refer to the standards IEC61158, IEC61784, and PI for detailed PROFINET specifications.
Function Blocks Library and Sample Program for Omron Controllers are available for download.
Please, visit Omron website for Function Blocks Library and Sample Program for additional PLC / Controllers.

3-1-1 Types of PROFINET

There are two PROFINET standards: PROFINET CBA and PROFINET IO.

- PROFINET CBA
  Inter-device communication using components. Mainly used between controllers.
- PROFINET IO
  Control by I/O data between a controller and devices.



This code reader supports PROFINET IO. PROFINET IO uses the same device model as PROFINET DP.
The information of each device is described in a GSD (General Station Description) file based on XML (Extensible Markup Language).

Communication Specifications of PROFINET IO

The communication specifications of PROFINET IO are described below.

Communication Specifications	Type	Details	Support
Periodic data communication method	RT (real-time) communication	Uses standard Ethernet hardware and achieves the same level of performance as the existing Fieldbus.	Supported
	IRT (Isochronous real-time) communication	This method provides a higher level of assurance than RT that communication will be executed within a specific time. Intended for use in systems such as motion control that require strict real-time.	Not supported

PROFINET IO specifies the supported functions by conformance class, with consideration given to the application.

Class	Overview	Support
Class A	Supports the basic functions of RT communication.	Supported
Class B	This class adds network diagnosis and redundancy functions used in process automation and other applications.	Not Supported
Class C	Supports IRT communication that realizes reliable synchronization.	Not Supported

The functions below are defined in Class A.

Function	Overview
Cyclic Data Exchange	Real-time data communication between the IO controller and IO devices at determined cycles. Set by IO data CR.
Acyclic Parameter Data / Device Identification	Used for parameter settings, IO device configuration, and reading of device information. Set by record data CR.
Device / Network Diagnosis	Communication for the purpose of sending alarms and statuses from IO devices to the IO controller. Set by Alarm CR.

The functions below are defined in Class B, which expands upon Class A.

Function	Overview
SNMP (Simple Network Management Protocol)	Allows additional Network Diagnostics via Management Information Base 2 (MIB2) and Lower Link Layer Discovery Protocol-MIB(LLDP-EXT-MIB).
PDEV (Physical Device Object)	Can also gather diagnostic information using acyclic PROFINET services.

Device Types Used in PROFINET IO

The devices below are defined in PROFINET IO.

Type	Details
IO Controller	Controller for external and other devices.
IO Device	Reader device connected to the IO controller. This code reader is an IO device.
IO Supervisor	PC or other device used for maintenance and diagnosis.

IO Devices

IO devices consist of DAPs and IO modules.
The functions and properties of these devices are described in a GSD file.

- **DAP (Device Access Point)**: This is an Ethernet access point and is used by means of a communication program.
- **IO Module**: Consists of the Slot, Subslot, and Index below. An IO module has one or multiple slots.
- **Slot**: Indicates the location of the IO module in the IO device.
- **Subslot**: IO interface inside the slot. This defines data types such as bit data and byte data, and the meanings of the data types.
- **Index**: Data in a Subslot.

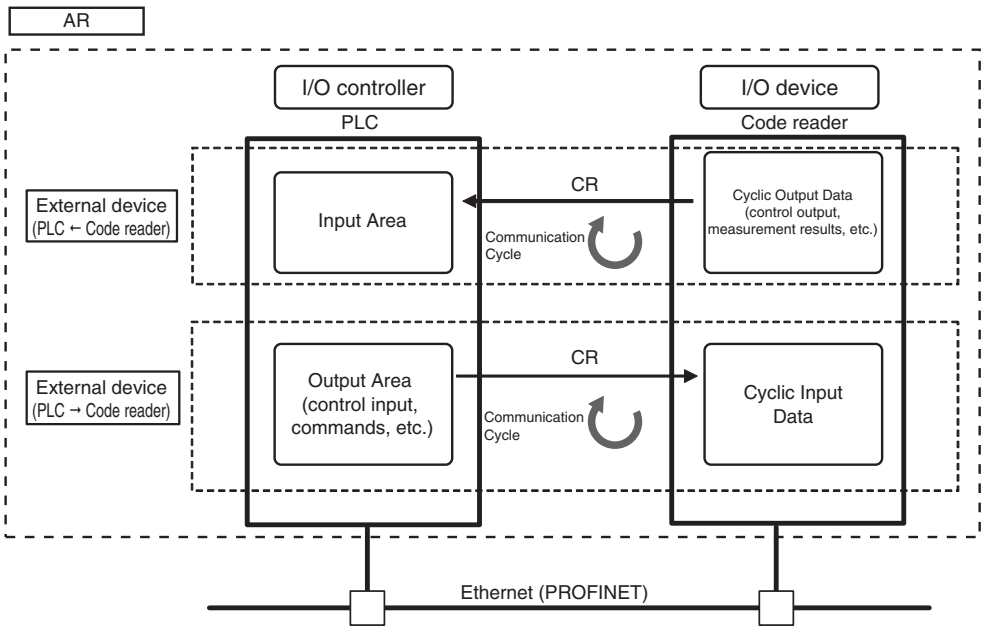
The above information is described in the GSD file of this code reader, and the IO controller uses the GSD file of this code reader to build the system.

Additional Information

When an IO device is used in PROFINET, the GSD file that describes the device functions and properties is used to configure the network configuration settings.
When this code reader is used in PROFINET as an IO device, the GSD file of this code reader must be installed in the engineering tool.

Data Communication in PROFINET IO

For an IO controller and IO device to communicate, a connection called an AR (Application Relation) must first be established between the two devices. When the AR connection is established, data communication between the IO controller and IO device takes place by means of a CR (Communication Relation) that defines the content of the data communication. An IO device can establish AR relations with multiple communication devices. In addition, multiple CR relations can be defined inside one AR. By establishing multiple CR relations inside one AR, communication that requires multiple profiles or differing Subslots can be performed. It is also possible to set a cycle time for each CR or IO.



CR is classified into IO data CR, record data CR, and alarm CR. Within the IO data CR, data communication is performed for each refreshing task period. Within CRs other than the IO data CR, communication takes place between the periodic data communications. Within the record data CR, the I/O controller will send commands to the IO device(s) at any time. IO device(s) will send back responses to the IO controller.

3-2 Code Reader Communications for PROFINET Connections

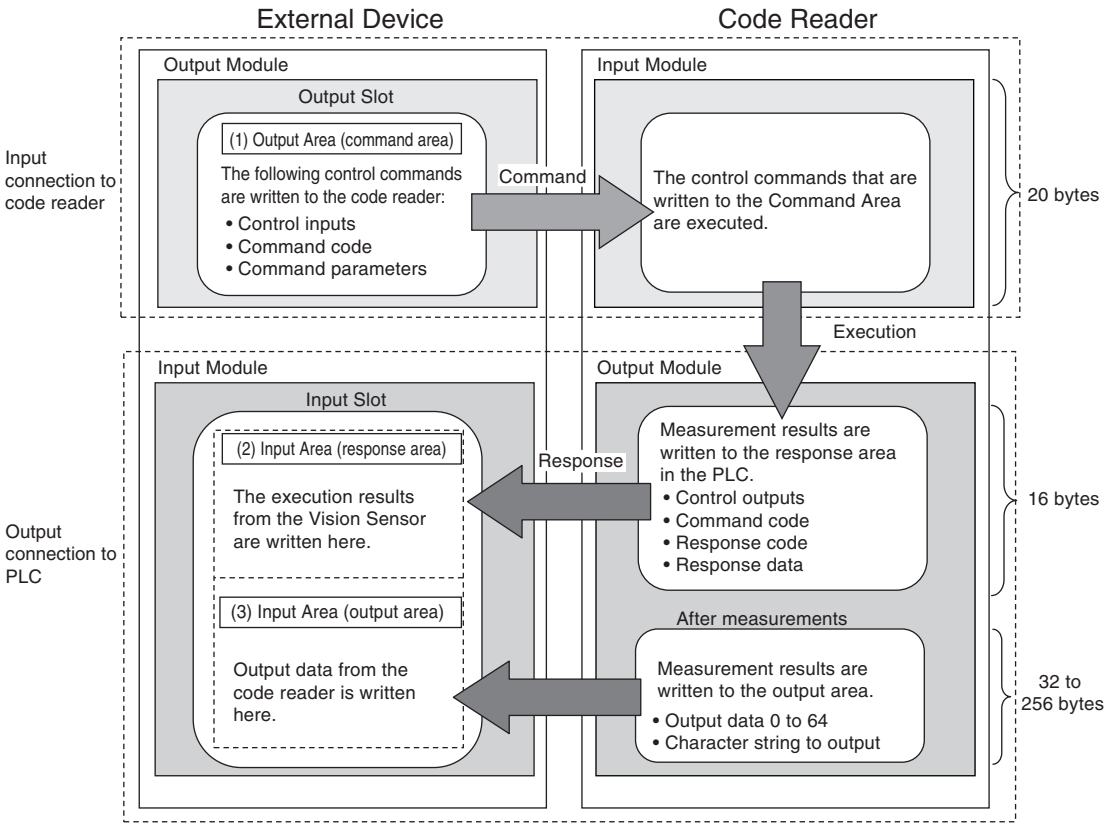
You can use PROFINET IO data CR to communicate between the PLC and the code reader to perform control via command/response communications or to output data after measurements.
This code reader complies with PROFINET conformance class A.
To connect to external devices and communicate using PROFINET, configure the PROFINET IO data CR settings with the engineering tool.
For details on the IO data CR settings in the engineering tool, refer to the manual for each engineering tool.

3-2-1 Types of Communications Areas

For PROFINET communications, the following three communications areas are used in the PLC to perform communications.

Command / Response Communications	(1) Output Area (Command Area)	This is the area to which you write control commands for this code reader to execute.
	(2) Input Area (Response Area)	This is the area to which this code reader writes the results of control commands executed from the command area.
Data Output after Measurements	(3) Input Area (Output Area)	This is the area to which this code reader writes output data for measurements after an inspection is performed.

The Input Area (Response Area) (2) and Input Area (Output Area) (3) are assigned to continuous memory addresses or to a variable.

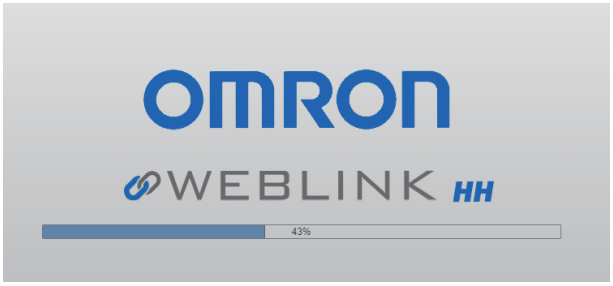

External Device
Code Reader
Output Module
Output Slot
(1) Output Area (command area)
The following control commands are written to the code reader:
• Control inputs
• Command code
• Command parameters
Command
Input Module
The control commands that are written to the Command Area are executed.
20 bytes
Input connection to code reader
Execution
Input Module
Input Slot
(2) Input Area (response area)
The execution results from the Vision Sensor are written here.
(3) Input Area (output area)
Output data from the code reader is written here.
Response
Output Module
Measurement results are written to the response area in the PLC.
• Control outputs
• Command code
• Response code
• Response data
16 bytes
After measurements
Measurement results are written to the output area.
• Output data 0 to 64
• Character string to output
32 to 256 bytes
Output connection to PLC

3-3 Setting Up PROFINET Communications

3-3-1 Configuring Network Settings in the Code Reader

1 Launch a browser and enter http://192.168.188.2.
Google Chrome is the recommended browser.

2 The WebLink$_{HH}$ startup screen will be displayed.

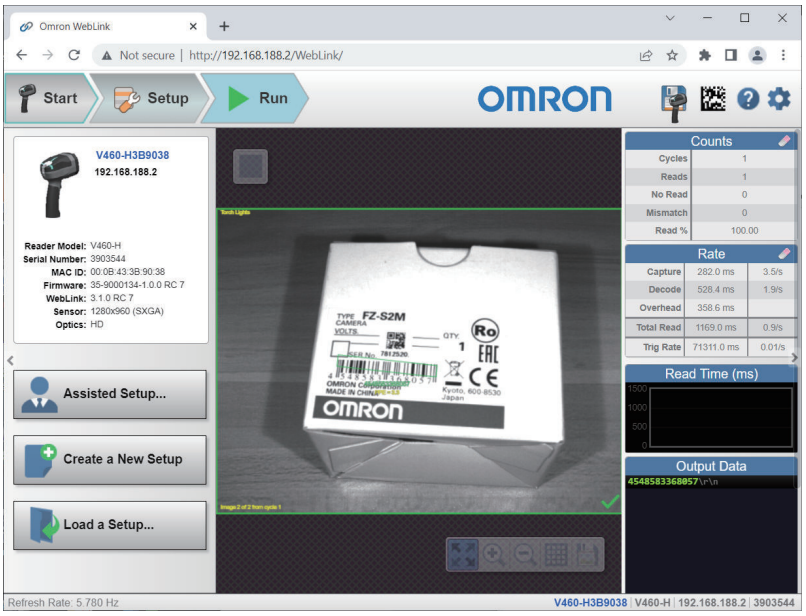


3 If the WebLink$_{HH}$ startup screen does not appear, it means that communication between the code reader and the PC has not been established. Check the following:

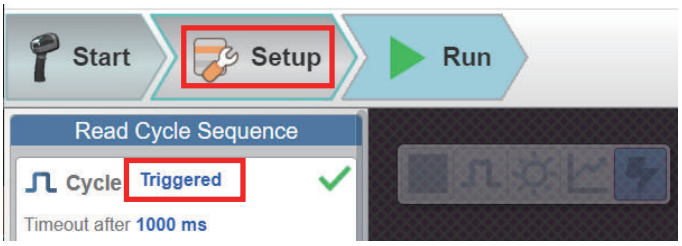
- Does the code reader and the PC have a proper physical (cable) connection?
- Are the respective IP Addresses on the PC and on the code reader set correctly?

Set the IP Address of the PC and perform a hardware reset of the code reader.
For other measures that can be taken, refer to the *V460-H Industrial Handheld DPM Reader User Manual (Z461 (84-9000460-02))*, *Appendices, Q&A, How to react when unable to connect to WebLink$_{HH}$*.

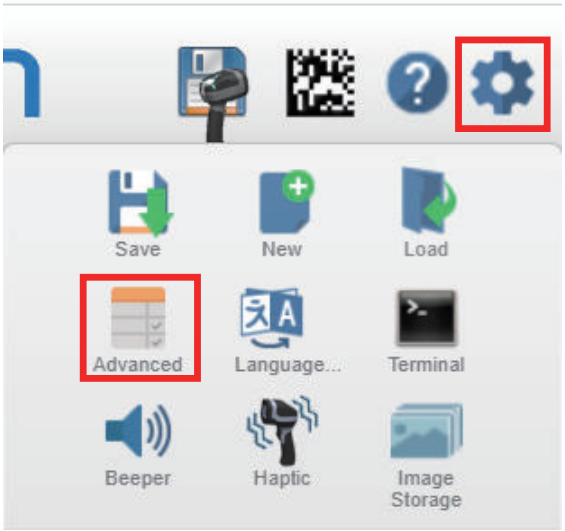
4 The WebLink$_{HH}$ screen shown below will appear.



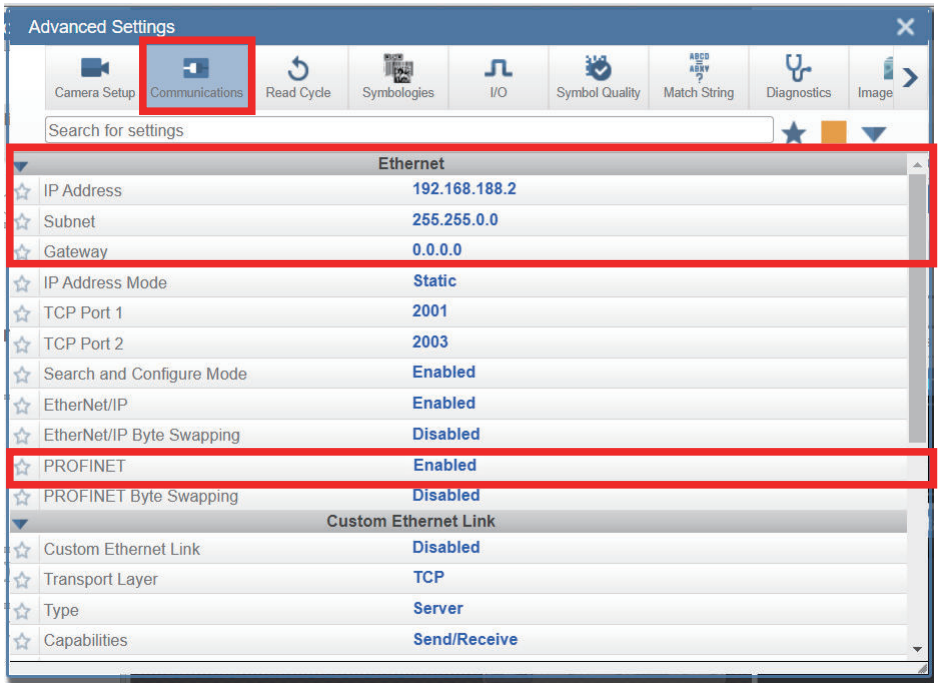
5 Click on the **Setup** tab and set the *Cycle* to *Triggered*.



6 Click on the gear icon on the upper right of the screen to select **Advanced** settings.



7 The *Advanced* Settings will appear. Check the settings indicated by the red boxes.
EtherNet/IP connection is **Enabled** by default. Disable EtherNet/IP and set **PROFINET** to **Enabled**.
If the IP address needs to be changed (when connecting multiple code readers, for example), configure the **IP Address** as needed for your application.

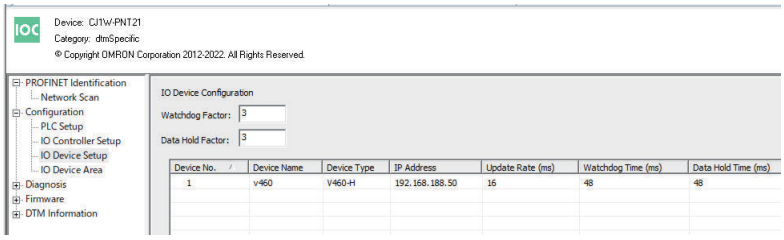

Advanced Settings
Camera Setup
Communications
Read Cycle
Symbologies
I/O
Symbol Quality
Match String
Diagnostics
Image
Search for settings
Ethernet
IP Address 192.168.188.2
Subnet 255.255.0.0
Gateway 0.0.0.0
IP Address Mode Static
TCP Port 1 2001
TCP Port 2 2003
Search and Configure Mode Enabled
EtherNet/IP Enabled
EtherNet/IP Byte Swapping Disabled
PROFINET Enabled
PROFINET Byte Swapping Disabled
Custom Ethernet Link
Custom Ethernet Link Disabled
Transport Layer TCP
Type Server
Capabilities Send/Receive

Communications Settings Procedure

To use PROFINET communication, the settings below must be configured.

Configure Network Settings of code reader	Set in WebLink$_{HH}$.
↓	
Configure Default PROFINET Settings	Set in WebLink$_{HH}$.
↓	
Configure IO Supervisor (Computer) Settings • IP Address Settings	
↓	
Configure IO Controller Settings • IP Address Settings • PROFINET IO System Settings • IO Device Settings and Assignments*1 • Compile and Save Settings	Configure the settings with the engineering tool. If you are setting the code reader as an IO device, install a GSD file that defines the V460-H IO data CR connection information in the engineering tool. The GSD file can be downloaded from our website. For PROFINET IO system settings, refer to applicable engineering tool manual(s).
↓	
Transfer Settings and Check Connection • Check IO Controller Connection • Transfer Parameters • Check Connection Status • Check Data Assignments	Connect the computer (engineering tool), code reader, and external devices, and use the engineering tool to transfer settings and check communication.

*1. If V460-H IP address is assigned in the PLC project and the PLC is not connected to the network when the V460-H is powered, it will revert to the default IP address "192.168.188.2", until the PLC is reconnected to the network.
The picture below shows the PROFINET configuration page for the Omron PLC, but this equally applies to other PLC vendors.



Memory Assignments

Refer to *A-3 PROFINET - V460-H Input and Output Modules* on page A-29 for the definition of input and output modules.

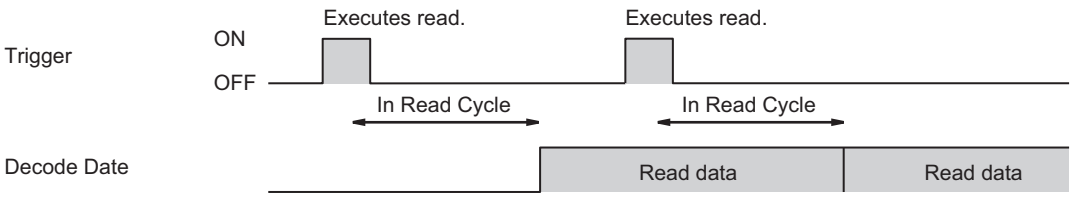
3-4 Timing Charts by Module Type

3-4-1 Read is Executed by the Read (TRIG) Signal

The timing signal at completion of storing the Read data to PLC data memory differs by the Input Module type.

Small Input Module (100)

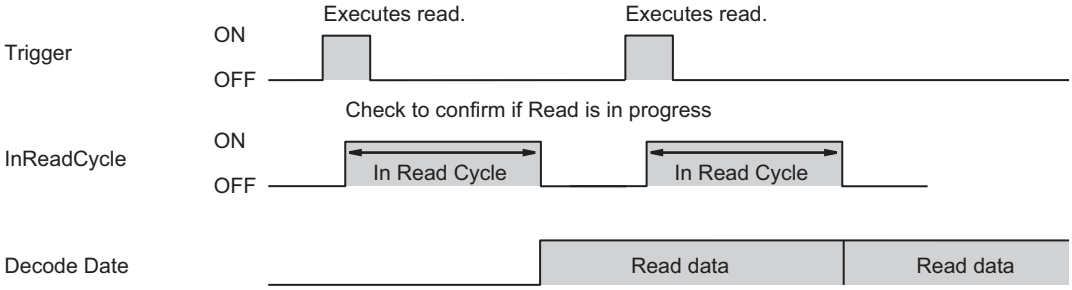
Small Input Module does not correspond to the Timing Signal for storing Read data.



1. Reading starts at the rising edge of the *Trigger*.
2. At the end of a Read, the read data is stored in *Decode Data*.

Large Input Module (101)

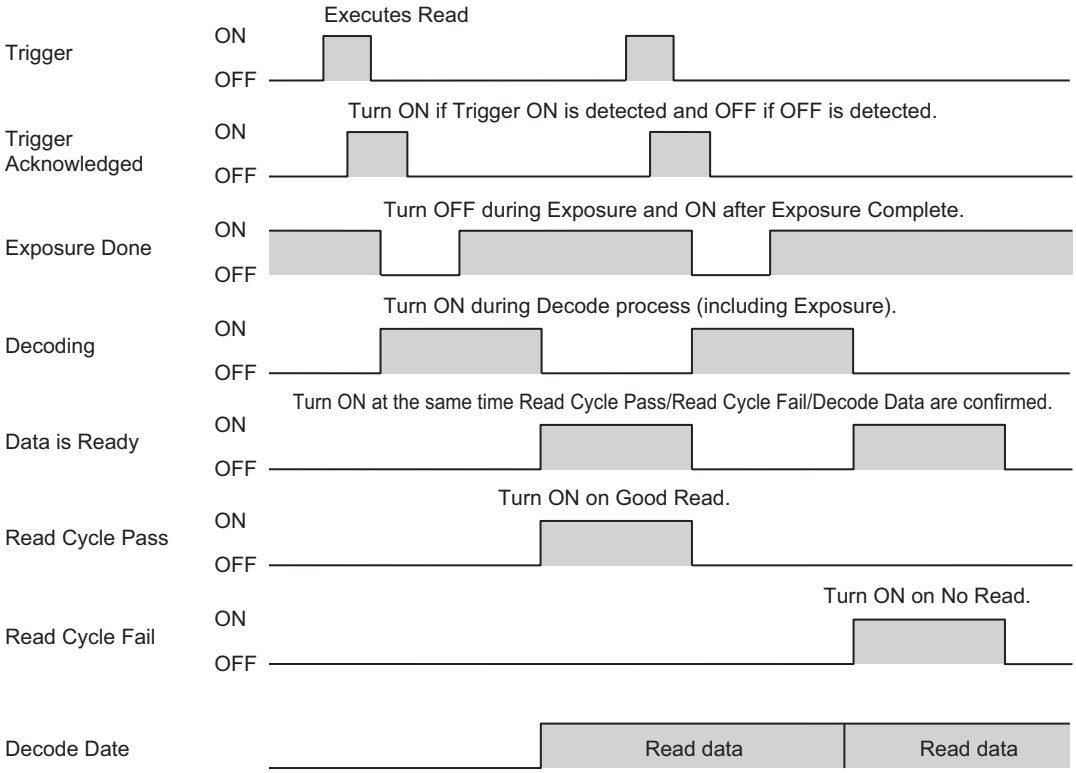
Large Input Module is output at the timing of the *Device Status* - *InReadCycle* bit turning from ON to OFF.



1. Reading starts at the rising edge of the *Trigger*.
2. At the start of a Read, *InReadCycle* turns ON and *Trigger* turns OFF.
3. At the end of a Read, the Read data is stored in *Decode Data* and *InReadCycle* turns OFF.

MXL Input Module (102)

MXL/Input Module (102) is output at the timing of the **Device Status** - **Decoding** bit turning from ON to OFF.



1. Reading starts at the rising edge of the **Trigger**.
2. **Trigger Acknowledged** turns ON when Trigger ON is detected and turns OFF when Trigger OFF is detected.
3. **Exposure Done** turns OFF when exposure starts and turns ON when exposure completes.
4. **Decoding** is ON during decoding processing. The Decoding process overlaps the Exposure process.
5. **Data is Ready** turns ON at the same time Decode Data / Read Cycle Pass or Read Cycle Fail is confirmed.
6. **Read Cycle Pass** turns ON when there is a Good Read and **Read Cycle Fail** turns ON when there is a No Read. The Read data is stored in **Decode Data**.

Additional Information

There can be up to a 10 ms delay in the Output timing of the Symbol data.

7. When the next **Trigger** is detected, **Data is Ready** turns OFF.

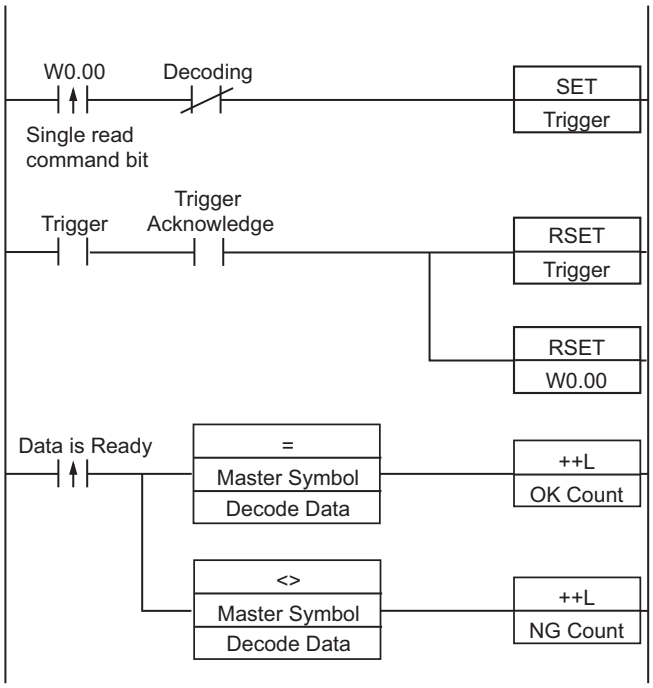
3-5 Sample Ladder Program

A sample ladder program is shown below.

- Input the Trigger Signal to execute Triggered Read.
- The read character string (Decode Data) is compared with the Verification string (Master Symbol) stored in the PLC.
- If they match, it is added to the Match/OK Count, and if they do not match, it is added to the Mismatch/NG Count.

The following Input and Output Modules are used.

- Input Module: MXL/SLC Input Module (102)
- Output Module: Output Module (197)



1. When the flag for Triggered is ON, The Trigger Bit turns ON.
2. The Trigger Acknowledged Bit (for detecting trigger input) is ON.
3. When the Trigger Acknowledged Bit ON is detected, the Trigger Bit turns OFF.
4. When Read is completed, the Data is Ready Bit turns ON.
5. The Read string (Decode Data) is compared with the Verification string (Master Symbol).
6. If the two strings match, the Match/OK Count is incremented by 1.
7. If the two strings do not match, the Mismatch/NG Count is incremented by 1.

A

Appendices

This section describes the industrial communication protocols that you can use with your code reader.

A-1 Command List

A-1-1 Command List

This section lists the commands that you can use with this code reader and the communications protocols for which each command is supported.

○: Supported Command, -: Non-Supported Command

Function	Serial (TCP)	EtherNet/IP
Change the settings	○	-*1
Performs Read	○	○
Starts Read Counts Test	○	-
Starts Read Rate Test	○	-
Ends Reads Count Test / Read Rate Test	○	-
Performs Calibration	○	-
Performs Training	○	-
Performs Optimization	○	-
Gets Error information from code reader	○	○
Gets settings	○	-*1
Turns Target Pattern light (Blue LED) ON/OFF	○	○
Enables / Disables Read Cycle	○	○
Gets Counters	○	○
Resets Counters	○	○
Gets Version information	○	-
Saves settings to Code reader	○	-
Restores code reader factory default settings	○	-
Restarts Code reader	○	-
Writes Read results to the Master Database	○	○
Gets Code quality grade report	○	-

*1. It can be used for sending serial command over EtherNet/IP message communications.

A-2 EtherNet/IP Specifications

A-2-1 EDS Files by Firmware Version

Product	Code Version	EDS File	Version	Product Code	Device Major Rev	Device Minor Rev
V460-H	1.0.0.xxxx	V460-H_1_0_0_20211119.eds	1.0	3414	1	1

Function Blocks Library and Sample Program for Omron Controllers are available for download. Please, visit Omron website for Function Blocks Library and Sample Program for additional PLC / Controllers.

A-2-2 Memory Allocation

An explanation of the memory allocation of each Input Assembly (code reader → PLC) and each Output Assembly (PLC → code reader).

Small Input (Instance ID: 100)

It is a compact, lightweight input assembly. It is designed to hold 64 bytes of information in the Read result. When reading multiple symbols, the Read strings are output delimited by Separator Characters. The following table lists the Member Structure of the Small Input Assembly

Small Input Member Structure

Member Name	Size (Bytes)
USER-DEFINED TAG ECHO	4
COMMAND ECHO	4
RESERVED	4
READ CYCLE SEQUENCE COUNTER	4
DECODE DATA LENGTH	4
DECODE DATA STRING	64

Total Size: 84 Bytes

Member Description

- **User-Defined Tag Echo**
  Returns the value set in the User-Defined Tag field of the Output (Legacy).
- **Command Echo**
  Returns the value set in the Command field of the Output (Legacy).
- **Read Cycle Sequence Counter**
  Stores the current Read Cycle Count.
- **Decode Data Length**
  Stores the number of characters in the Read string.
- **Decode Data String**
  Stores the Read string. When additional information such as Print Quality Grading Standard is set, it is stored following the Read string.

Memory Allocation

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
32 bit	**User Defined Tag Echo**	**DINT**		**4 Byte**	**0**
	UserTag_1		0	1 bit	
	UserTag_2		1	1 bit	
	UserTag_3		2	1 bit	
	UserTag_4		3	1 bit	
	UserTag_5		4	1 bit	
	UserTag_6		5	1 bit	
	UserTag_7		6	1 bit	
	UserTag_8		7	1 bit	
	UserTag_9		8	1 bit	
	UserTag_10		9	1 bit	
	UserTag_11		10	1 bit	
	UserTag_12		11	1 bit	
	UserTag_13		12	1 bit	
	UserTag_14		13	1 bit	
	UserTag_15		14	1 bit	
	UserTag_16		15	1 bit	
	UserTag_17		16	1 bit	
	UserTag_18		17	1 bit	
	UserTag_19		18	1 bit	
	UserTag_20		19	1 bit	
	UserTag_21		20	1 bit	
	UserTag_22		21	1 bit	
	UserTag_23		22	1 bit	
	UserTag_24		23	1 bit	
	UserTag_25		24	1 bit	
	UserTag_26		25	1 bit	
	UserTag_27		26	1 bit	
	UserTag_28		27	1 bit	
	UserTag_29		28	1 bit	
	UserTag_30		29	1 bit	
	UserTag_31		30	1 bit	
	UserTag_32		31	1 bit	
32 bit	**Command Echo**	**DINT**		**4 Byte**	**4**
	Trigger_Echo		0	1 bit	
	New Master Echo		1	1 bit	
	Reserved		2 - 7	6 bit	
	Disable Scanning Echo		8	1 bit	
	Reserved		9 - 15	7 bit	
	Clear Read Cycle Report and Counters Echo		16	1 bit	
	Reserved		17 - 31	15 bit	
32 bit	**Reserved**	**DINT**		**4 Byte**	**8**
32 bit	**Read Cycle Sequence Counter**	**UDINT**	0 - 31	**4 byte**	**12**
32 bit	**Decode Data Length**	**UDINT**	0 - 31	**4 byte**	**16**
	Decode Data String	**SINT[64]**	0 - 512	**64 byte**	**20**

Large Input (Instance ID: 101)

Compared to the Small Input, the Large Input holds more Device Status information and Read result character strings of 128 bytes. When reading multiple symbols, the Read strings are output delimited by Separator Characters.

Large Input Member Structure

Member Name	Size (Bytes)
USER-DEFINED TAG ECHO	4
COMMAND ECHO	4
RESERVED	4
RESERVED	4
RESERVED	4
DEVICE STATUS (Legacy)	4
READ CYCLE SEQUENCE COUNTER	4
TRIGGER COUNT	4
DECODE/MATCH COUNT	4
MISMATCH COUNT	4
NOREAD COUNT	4
DECODE DATA LENGTH	4
DECODE DATA STRING	128

Total Size: 176 Bytes

Member Description

- **User-Defined Tag Echo**
  Returns the value set in the User-Defined Tag field of the Output (Legacy).
- **Command Echo**
  Returns the value set in the Command field of the Output (Legacy).
- **Device Status (Legacy)**
  Displays code reader Status

Bit	State	Description
0	Reserved	-
1	New Master Requested	When the bit is ON, the next read result is registered as the Master Symbol.
2 - 7	Reserved	-
8	Scanning Disabled	When the bit is ON, the Read Cycle is Disabled.
9 - 15	Reserved	-
16	In Read Cycle	Bit is ON when In Read Cycle.
17	Actively Scanning	Bit is ON when In Read Cycle.

- **Read Cycle Sequence Counter**
  Stores the current Read Cycle Count.
- **Trigger Counter**
  Stores the current total number of triggers input.
- **Decode/Matchcode Counter**
  Stores one of the following.
  1. Total number of Good Reads (When Matchcode: Disabled)

2. Total number of matches to the Master Symbol (When Matchcode: Enabled)

- **Mismatch Counter**
  Stores the total number of Mismatches (not matching Master Symbol).
- **No Read Counter**
  Stores the total number of No Reads.
- **Decode Data Length**
  Stores the number of characters in the Read string.
- **Decode Data String**
  Stores the Read string. When additional information such as Print Quality Grading Standard is set, it is stored following the Read string.

Memory Allocation

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
32 bit	**User Defined Tag Echo**	**DINT**		**4 Byte**	**0**
	UserTag_1		0	1 bit	
	UserTag_2		1	1 bit	
	UserTag_3		2	1 bit	
	UserTag_4		3	1 bit	
	UserTag_5		4	1 bit	
	UserTag_6		5	1 bit	
	UserTag_7		6	1 bit	
	UserTag_8		7	1 bit	
	UserTag_9		8	1 bit	
	UserTag_10		9	1 bit	
	UserTag_11		10	1 bit	
	UserTag_12		11	1 bit	
	UserTag_13		12	1 bit	
	UserTag_14		13	1 bit	
	UserTag_15		14	1 bit	
	UserTag_16		15	1 bit	
	UserTag_17		16	1 bit	
	UserTag_18		17	1 bit	
	UserTag_19		18	1 bit	
	UserTag_20		19	1 bit	
	UserTag_21		20	1 bit	
	UserTag_22		21	1 bit	
	UserTag_23		22	1 bit	
	UserTag_24		23	1 bit	
	UserTag_25		24	1 bit	
	UserTag_26		25	1 bit	
	UserTag_27		26	1 bit	
	UserTag_28		27	1 bit	
	UserTag_29		28	1 bit	
	UserTag_30		29	1 bit	
	UserTag_31		30	1 bit	
	UserTag_32		31	1 bit	

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
32 bit	**Command Echo**	**DINT**		**4 Byte**	**4**
	Trigger Echo		0	1 bit	
	New Master Echo		1	1 bit	
	Reserved		2 - 7	6 bit	
	Disable Scanning Echo		8	1 bit	
	Reserved		9 - 15	7 bit	
	Clear Read Cycle Report and Counters Echo		16	1 bit	
	Reserved		17 - 31	15 bit	
32 bit	**Reserved**	**DINT**		**4 Byte**	**8**
32 bit	**Reserved**	**DINT**		**4 Byte**	**12**
32 bit	**Reserved**	**DINT**		**4 Byte**	**16**
32 bit	**Device Status**	**DINT**		**4 Byte**	**20**
	Reserved		0	1 bit	
	New Master Requedted		1	1 bit	
	Reserved		2 - 7	6 bit	
	Scanning Disabled		8	1 bit	
	Reserved		9 - 15	7 bit	
	In Read Cycle		16	1 bit	
	Actively Scanning		17	1 bit	
	Reserved		18 - 31	14 bit	
32 bit	**Read Cycle Sequence Counter**	**UDINT**	0 - 31	**4 byte**	**24**
32 bit	**Trigger Count**	**UDINT**	0 - 31	**4 byte**	**28**
32 bit	**Decode/Matchcode Count**	**UDINT**	0 - 31	**4 byte**	**32**
32 bit	**Mismatch Count**	**UDINT**	0 - 31	**4 byte**	**36**
32 bit	**No Read Count**	**UDINT**	0 - 31	**4 byte**	**40**
32 bit	**Decode Data Length**	**UDINT**	0 - 31	**4 byte**	**44**
	Decode Data String	**SINT[128]**	0 - 1024	**128 byte**	**48**

MXL/SLC Input (Instance ID: 102)

Compared to the Large Input, the MXL/SLC Input holds the more detailed Device Status information and Read result character strings of up to 184 bytes. When reading multiple symbols, the Read strings are output delimited by Separator Characters.

MXL/SLC Input Member Structure

Member Name	Size (Bytes)
RESERVED	1
RESERVED	1
RESERVED	1
RESERVED	1
DEVICE STATUS	4
RESERVED	4
COUNTERS	24
READ CYCLE REPORT	8
DECODE CYCLE REPORT	16
DECODE DATA LENGTH	4

Member Name	Size (Bytes)
DECODE DATA STRING	184

Total Size: 248 Bytes

Member Description

- **Device Status**

  Shows the current status of the code reader.

Bit	State
0	Run Mode
1	Trigger Acknowledged
2	Exposure Done
3	Decoding
4	Data Is Ready
5	Read Cycle Pass
6	Read Cycle Fail
7	General Fault
8	Matchcode Master Label Trained
9	Matchcode Enabled
10	Image Sensor Calibrating
11	Image Sensor Calibration Complete
12	Training
13	Training Complete
14	Optimizing
15	Optimization Complete
16	AutoImage Photometry Enabled
17	AutoImage Photometry Complete
18	Reserved
19	Reserved
20	Reserved
21	Buffer Overflow
22 - 31	Reserved

- Run Mode

Shows Read Cycle Enabled/Disabled status.

0 = Read Cycle Disabled cannot accept Trigger. However, it can receive a command.

1 = Read Cycle Enabled State in which trigger can be accepted.

- Trigger Acknowledged

This bit becomes *1* when the Trigger bit from the Output Assembly is received.

When the Trigger bit is OFF, Trigger Acknowledged also becomes *0*.

- Exposure Done

During exposure, this bit is set to *0*.

When Exposure is done, this bit becomes *1*.

- Decoding

During image processing, this bit is set to *1*.

When image processing is done, this bit becomes *0*.

- Data is Ready

When the data from Read Cycle Report and Data Cycle Report is confirmed, this bit becomes *1*.

When the next Read starts, this bit becomes *0*.

- Read Cycle Pass
On Good Read, this bit becomes *1*.
When the next Read starts, this bit becomes *0*.
- Read Cycle Fail
If the read cycle fails for any reason (No Read, Mismatch, etc,) this bit becomes *1*. This bit will be set to *0* at the start of a read cycle.
- General Fault
When a code reader error occurs, this bit becomes *1*. The user must resolve the problem by refering to the Fault Code field of the error code. After resolving the problem, the user must set "Reset General fault" in the Output Assembly Control to *0*.
- Matchcode Master Label Trained
When active, the unit has accepted the data read on the last trigger and the new master label used in the matchcode function.
- Matchcode Enabled
When Matchcode is Enabled, this bit becomes *1*.
- Image Sensor Calibrating
This bit is set to *1* while the device is executing the following calibrations.
Exposure
Gain
This bit is set to *0* when the device calibration is complete.
- Image Sensor Calibration Complete
This bit is set to *1* when the device completes executing the following calibrations.
Exposure
Gain
- Training
This bit is set to *1* while Training is in progress.
This bit is set to *0* when Training is complete.
- Training Complete
This bit will be set to *0* during training and will be set to *1* when training is successful. If an error occurs, the bit will remain at *0*.
- Optimizing
This bit is set to *1* while Optimization is in progress.
This bit is set to *0* when Optimization is complete.
- Optimization Complete
This bit is set to *1* when Optimization processing is complete. If an error occurs, it is output by Fault Code area.
- AutoImage Photometry Enabled
This bit is set to *1* when Auto Photometry is used.
This bit is set to *0* when AutoImage Photometry is complete.
- AutoImage Photometry Complete
This bit is set to *1* when AutoImage Photometry processing is complete. If an error occurs, it is output by Fault Code area.
- Buffer Overflow
This bit is set to *1* when the read string length exceeds the size of the Decode Data area.

- **Counters**
Various counters of Read results after starting the device are output.
These counters can be set from the Command Field/Area of the Output Assembly.

Counters	Size (Bytes)
No Read Read Cycle Counter	4
Mismatch per Read Cycle Counter	4
No Read Counter	4
Trigger Counter	4
Matchcode Counter	4
Mismatch Counter	4

- No Read Read Cycle Counter

Outputs the total number of Read Cycle No Reads.

- Mismatch per Read Cycle Counter

Outputs the total number of Read Cycle Mismatches.

- No Read Counter

Outputs the total number of No Reads.

- Trigger Counter

Outputs the total number of executed Triggers.

- Matchcode Counter

Outputs one of the following.

1. Total number of matches to the Master Symbol (When Matchcode: Enabled)
2. Total number of Good Reads (When Matchcode: Disabled)

- Mismatch Counter

Outputs the total number of Mismatches (not matching Master Symbol).

- **Read Cycle Report**

Read Cycle Report	Size (Bytes)
Capture Time	2
Decode Time	2
Total Read Cycle Time	2
Reserved	2

- Capture Time

The time required for image capture of the image that had the successful decode. If none of the images were decoded, this will be the first image capture time. (milliseconds)

- Decode Time

The time required for decoding a symbol. If none of the images were decoded, then the time of the first image decode will be reported. (milliseconds)

- Total Read Cycle Time

The total time taken to read symbols. This encompasses the total time of image capture, decoding and overhead. (milliseconds)

- **Decode Cycle Report**

Outputs symbol information.

Symbol Information	Size (Bytes)
Decode Location Top	2
Decode Location Left	2
Decode Location Height	2
Decode Location Width	2
Code Type	4

Symbol Information	Size (Bytes)
Pixels per Element	4

- Decode Location Top

The upper left Y coordinate of the Symbol Detection Region. (pixels)

- Decode Location Left

The upper left X coordinate of the Symbol Detection Region. (pixels)

- Decode Location Height

The height (Y size) of the Symbol Detection Region. (pixels)

- Decode Location Width

The width (X size) of the Symbol Detection Region. (pixels)

- Code Type

A bit indicating the Symbol Type of the decoded symbol is output.

Bit	State
0	Aztec Code
1	Micro QR Code
2	Postal Code
3	Code 39
4	Codabar
5	Interleaved 2 of 5
6	UPC/EAN
7	Code 128/EAN 128
8	Code 93
9	PDF417
10	Pharma Code
11	Data Matrix
12	QR Code
13	BC412
14	GS1 Databar
15	GS1 Databar Limited
16	GS1 Databar Expanded
17	Micro PDF
18	Composite
19	Dot Code
20 - 31	Reserved

- **Pixels Per Element**

  Outputs the number of pixels displayed in 1 cell size (or narrow element) on the image.

- **Decode Data Length**

  Stores the number of characters in the Read string.

- **Decode Data String**

  Stores the Read string. When additional information such as Print Quality Grading Standard is set, it is stored following the Read string.

Memory Allocation

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
32 bit	**Reserved**	**SINT**		**1 Byte**	**0**
	Reserved	**SINT**		**1 Byte**	
	Reserved	**SINT**		**1 Byte**	

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
	Reserved	**SINT**		**1 byte**	
32 bit	**DeviceStatus**	**DINT**		**4 Byte**	**4**
	Run Mode		0	1 bit	
	Trigger Acknowledged		1	1 bit	
	Exposure Done		2	1 bit	
	Decoding		3	1 bit	
	Data is Ready		4	1 bit	
	Read Cycle Pass		5	1 bit	
	Read Cycle Fail		6	1 bit	
	General Fault		7	1 bit	
	Matchcode Master Label Trained		8	1 bit	
	Matchcode Enabled		9	1 bit	
	Image Sensor Calibrating		10	1 bit	
	Image Sensor Calibration Complete		11	1 bit	
	Training		12	1 bit	
	Training Complete		13	1 bit	
	Optimizing		14	1 bit	
	Optimizing Complete		15	1 bit	
	Auto Image Photometry Enabled		16	1 bit	
	Auto Image Photometry Complete		17	1 bit	
	Reserved		18	1 bit	
	Reserved		19	1 bit	
	Reserved		20	1 bit	
	BufferOverflow		21	1 bit	
	Reserved		22 - 31	10 bit	
32 bit	**Reserved**	**DINT**		**4 Byte**	**8**
192 bit	**Counters**	**UDINT[6]**		**24 byte**	**12**
	No Read Read Cycle Counter		0 - 31	4 byte	
	Mismatch per Read Cycle Counter		0 - 31	4 byte	
	No Read Counter		0 - 31	4 byte	
	Trigger Counter		0 - 31	4 byte	
	Matchcode Counter		0 - 31	4 byte	
	Mismatch Counter		0 - 31	4 byte	
64 bit	**Read Cycle Report**	**UINT[4]**		**8 byte**	**36**
	Capture Time		0 - 15	2 byte	
	Decode Time		0 - 15	2 byte	
	Total Read Cycle Time		0 - 15	2 byte	
	Reserved		0 - 15	2 byte	
128 bit	**Decode Cycle Report**			**16 Byte**	**44**
	Decode Location Top	**UINT**	0 - 15	2 byte	
	Decode Location Left	**UINT**	0 - 15	2 byte	
	Decode Location Height	**UINT**	0 - 15	2 byte	
	Decode Location Width	**UINT**	0 - 15	2 byte	

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
	Code Type	**DINT**		**4 Byte**	**52**
	Aztec Code		0	1 bit	
	Micro QR Code		1	1 bit	
	Postal Code		2	1 bit	
	Code 39		3	1 bit	
	Codabar		4	1 bit	
	Interleaved 2 of 5		5	1 bit	
	UPC EAN		6	1 bit	
	Code 128 EAN 128		7	1 bit	
	Code 93		8	1 bit	
	PDF417		9	1 bit	
	Pharma Code		10	1 bit	
	Data Matrix		11	1 bit	
	QR Code		12	1 bit	
	BC412		13	1 bit	
	GS1 Databar		14	1 bit	
	GS1 Databar Limited		15	1 bit	
	GS1 Databar Expanded		16	1 bit	
	Micro PDF		17	1 bit	
	Composite		18	1 bit	
	Dot Code		19	1 bit	
	Reserved		20 - 31	12 bit	
	Pixels per Element	**REAL**	0 - 31	**4 byte**	**56**
32 bit	**Decode Data Length**	**DINT**	0 - 31	**4 byte**	**60**
	Decode Data String	**SINT[184]**	0 - 1471	**184 byte**	**64**

1 Decode Input (Instance ID: 103)

1 Decode Input is designed to hold a 436 byte Read result string. When reading multiple symbols, the Read strings are output delimited by Separator Characters.

1 Decode Input Member Structure

Member Name	Size (Bytes)
RESERVED	1
RESERVED	1
RESERVED	1
RESERVED	1
DEVICE STATUS	4
RESERVED	4
COUNTERS	24
READ CYCLE REPORT	8
DECODE CYCLE REPORT	16
DECODE DATA LENGTH	4
DECODE DATA STRING	436

Total Size: 500 Bytes

Member Description

- **Device Status**
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as
- **Counters**
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as
- **Read Cycle Report**
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as
- **Decode Cycle Report**
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as
- **Decode Data Length**
  Stores the number of characters in the Read string.
- **Decode Data String**
  Stores the Read string. When additional information such as Print Quality Grading Standard is set, it is stored following the Read string.

Memory Allocation

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
32 bit	**Reserved**	**SINT**		**1 Byte**	**0**
	Reserved	**SINT**		**1 Byte**	
	Reserved	**SINT**		**1 Byte**	
	Reserved	**SINT**		**1 byte**	
32 bit	**DeviceStatus**	**DINT**		**4 Byte**	**4**
	Run Mode		0	1 bit	
	Trigger Acknowledged		1	1 bit	
	Exposure Done		2	1 bit	
	Decoding		3	1 bit	
	Data is Ready		4	1 bit	
	Read Cycle Pass		5	1 bit	
	Read Cycle Fail		6	1 bit	
	General Fault		7	1 bit	
	Matchcode Master Label Trained		8	1 bit	
	Matchcode Enabled		9	1 bit	
	Image Sensor Calibrating		10	1 bit	
	Image Sensor Calibration Complete		11	1 bit	
	Training		12	1 bit	
	Training Complete		13	1 bit	
	Optimizing		14	1 bit	
	Optimizing Complete		15	1 bit	
	Auto Image Photometry Enabled		16	1 bit	
	Auto Image Photometry Complete		17	1 bit	
	Reserved		18	1 bit	
	Reserved		19	1 bit	
	Reserved		20	1 bit	
	BufferOverflow		21	1 bit	
	Reserved		22 - 31	10 bit	
32 bit	**Reserved**	**DINT**		**4 Byte**	**8**
192 bit	**Counters**	**UDINT[6]**		**24 byte**	**12**

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
	No Read Read Cycle Counter		0 - 31	4 byte	
	Mismatch per Read Cycle Counter		0 - 31	4 byte	
	No Read Counter		0 - 31	4 byte	
	Trigger Counter		0 - 31	4 byte	
	Matchcode Counter		0 - 31	4 byte	
	Mismatch Counter		0 - 31	4 byte	
64 bit	**Read Cycle Report**	**UINT[4]**		**8 byte**	**36**
	Capture Time		0 - 15	2 byte	
	Decode Time		0 - 15	2 byte	
	Total Read Cycle Time		0 - 15	2 byte	
	Reserved		0 - 15	2 byte	
128 bit	**Decode Cycle Report**			**16 Byte**	**44**
	Decode Location Top	**UINT**	0 - 15	2 byte	
	Decode Location Left	**UINT**	0 - 15	2 byte	
	Decode Location Height	**UINT**	0 - 15	2 byte	
	Decode Location Width	**UINT**	0 - 15	2 byte	
	Code Type	**DINT**		**4 Byte**	**52**
	Aztec Code		0	1 bit	
	Micro QR Code		1	1 bit	
	Postal Code		2	1 bit	
	Code 39		3	1 bit	
	Codabar		4	1 bit	
	Interleaved 2 of 5		5	1 bit	
	UPC EAN		6	1 bit	
	Code 128 EAN 128		7	1 bit	
	Code 93		8	1 bit	
	PDF417		9	1 bit	
	Pharma Code		10	1 bit	
	Data Matrix		11	1 bit	
	QR Code		12	1 bit	
	BC412		13	1 bit	
	GS1 Databar		14	1 bit	
	GS1 Databar Limited		15	1 bit	
	GS1 Databar Expanded		16	1 bit	
	Micro PDF		17	1 bit	
	Composite		18	1 bit	
	Dot Code		19	1 bit	
	Reserved		20 - 31	12 bit	
	Pixels per Element	**REAL**	0 - 31	**4 byte**	**56**
32 bit	**Decode Data Length**	**DINT**	0 - 31	**4 byte**	**60**
	Decode Data String	**SINT[436]**	0 - 3487	**436 byte**	**64**

4 Decode Input (Instance ID: 104)

4 Decode Input is designed to hold the Read result information of 4 symbols. The first Read result is stored in a 160 byte field. The remaining Read results are stored in a 72 byte field. Use this when you want to execute a multiple symbol Read for up to 4 symbols and query symbol information such as symbol position coordinates for each symbol.

ATTENTION:
If **Format Output is NOT enabled on the reader**, the **DECODE 'X' DATA**, **DECODE 'X' LENGTH** and **DECODE 'X' CYCLE REPORT** will reflect the data of each code, grouped all together, i.e., **DECODE 1 DATA**, **DECODE 1 LENGTH** and **DECODE 1 CYCLE REPORT** related to the same code.
If **Format Output is ENABLED on the reader**, the **DECODE 'X' DATA** and **DECODE 'X' LENGTH** will reflect the format defined by user. However, the **DECODE 'X' CYCLE REPORT** cannot be affected by the format defined by user, keeping its information in the same way as if the **Format Output is NOT enabled on the reader**.
Summarizing:

- If **Format Output is ENABLED on the reader**, the **DECODE 'X' CYCLE REPORT** cannot be related to such code.
- If you need them, to guarantee the correct information, enable "Output Coordinates" and "Pixel Per Element" and "Include Symbology Identifier" options (I/O tab) to be added to the **Format Output**, and do consider only the information presented by **DECODE 'X' DATA** and **DECODE 'X' LENGTH**, ignoring the information presented by **DECODE 'X' CYCLE REPORT**.

4 Decode Input Member Structure

Member Name	Size (Bytes)
RESERVED	1
RESERVED	1
RESERVED	1
RESERVED	1
DEVICE STATUS	4
RESERVED	4
COUNTERS	24
READ CYCLE REPORT	8
DECODE 1 CYCLE REPORT	16
DECODE 1 LENGTH	4
DECODE 1 DATA	160
DECODE 2 CYCLE REPORT	16
DECODE 2 LENGTH	4
DECODE 2 DATA	72
DECODE 3 CYCLE REPORT	16
DECODE 3 LENGTH	4
DECODE 3 DATA	72
DECODE 4 CYCLE REPORT	16
DECODE 4 LENGTH	4
DECODE 4 DATA	72

Total Size: 500 Bytes

Member Description

- **Device Status**

MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as

- **Counters**
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as
- **Read Cycle Report**
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as
- **Decode 1 Cycle Report**
  The information for the 1st symbol.
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as Decode Cycle Report
- **Decode 1 Data Length**
  Stores the number of characters that comprise the 1st symbol.
- **Decode 1 Data String**
  Stores the Read string of the 1st symbol. When additional information such as Print Quality Grading Standard is set, it is stored following the Read string.
- **Decode 2 Cycle Report**
  The information for the 2nd symbol.
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as Decode Cycle Report
- **Decode 2 Data Length**
  Stores the number of characters that comprise the 2nd symbol.
- **Decode 2 Data String**
  Stores the Read string of the 2nd symbol. When additional information such as Print Quality Grading Standard is set, it is stored following the Read string.
- **Decode 3 Cycle Report**
  The information for the 3rd symbol.
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as Decode Cycle Report
- **Decode 3 Data Length**
  Stores the number of characters that comprise the 3rd symbol.
- **Decode 3 Data String**
  Stores the Read string of the 3rd symbol. When additional information such as Print Quality Grading Standard is set, it is stored following the Read string.
- **Decode 4 Cycle Report**
  The information for the 4th symbol.
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as Decode Cycle Report
- **Decode 4 Data Length**
  Stores the number of characters that comprise the 4th symbol.
- **Decode 4 Data String**
  Stores the Read string of the 4th symbol. When additional information such as Print Quality Grading Standard is set, it is stored following the Read string.

Memory Allocation

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
32 bit	**Reserved**	**SINT**		**1 Byte**	**0**
	Reserved	**SINT**		**1 Byte**	
	Reserved	**SINT**		**1 Byte**	
	Reserved	**SINT**		**1 byte**	
32 bit	**DeviceStatus**	**DINT**		**4 Byte**	**4**
	Run Mode		0	1 bit	
	Trigger Acknowledged		1	1 bit	

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
	Exposure Done		2	1 bit	
	Decoding		3	1 bit	
	Data is Ready		4	1 bit	
	Read Cycle Pass		5	1 bit	
	Read Cycle Fail		6	1 bit	
	General Fault		7	1 bit	
	Matchcode Master Label Trained		8	1 bit	
	Matchcode Enabled		9	1 bit	
	Image Sensor Calibrating		10	1 bit	
	Image Sensor Calibration Complete		11	1 bit	
	Training		12	1 bit	
	Training Complete		13	1 bit	
	Optimizing		14	1 bit	
	Optimizing Complete		15	1 bit	
	Auto Image Photometry Enabled		16	1 bit	
	Auto Image Photometry Complete		17	1 bit	
	Reserved		18	1 bit	
	Reserved		19	1 bit	
	Reserved		20	1 bit	
	BufferOverflow		21	1 bit	
	Reserved		22 - 31	10 bit	
32 bit	**Reserved**	**DINT**		**4 Byte**	**8**
192 bit	**Counters**	**UDINT[6]**		**24 byte**	**12**
	No Read Read Cycle Counter		0 - 31	4 byte	
	Mismatch per Read Cycle Counter		0 - 31	4 byte	
	No Read Counter		0 - 31	4 byte	
	Trigger Counter		0 - 31	4 byte	
	Matchcode Counter		0 - 31	4 byte	
	Mismatch Counter		0 - 31	4 byte	
64 bit	**Read Cycle Report**	**UINT[4]**		**8 byte**	**36**
	Capture Time		0 - 15	2 byte	
	Decode Time		0 - 15	2 byte	
	Total Read Cycle Time		0 - 15	2 byte	
	Reserved		0 - 15	2 byte	
128 bit	**Decode 1 Cycle Report**			**16 byte**	**44**
	Decode Location Top	**UINT**	0 - 15	2 byte	
	Decode Location Left	**UINT**	0 - 15	2 byte	
	Decode Location Height	**UINT**	0 - 15	2 byte	
	Decode Location Width	**UINT**	0 - 15	2 byte	
	Code Type	**DINT**		**4 Byte**	**52**
	Aztec Code		0	1 bit	
	Micro QR Code		1	1 bit	
	Postal Code		2	1 bit	
	Code 39		3	1 bit	
	Codabar		4	1 bit	
	Interleaved 2 of 5		5	1 bit	

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
	UPC EAN		6	1 bit	
	Code 128 EAN 128		7	1 bit	
	Code 93		8	1 bit	
	PDF417		9	1 bit	
	Pharma Code		10	1 bit	
	Data Matrix		11	1 bit	
	QR Code		12	1 bit	
	BC412		13	1 bit	
	GS1 Databar		14	1 bit	
	GS1 Databar Limited		15	1 bit	
	GS1 Databar Expanded		16	1 bit	
	Micro PDF		17	1 bit	
	Composite		18	1 bit	
	Dot Code		19	1 bit	
	Reserved		20 - 31	12 bit	
	Pixels per Element	**REAL**	0 - 31	**4 byte**	**56**
32 bit	**Decode 1 Data Length**	**DINT**	0 - 31	**4 byte**	**60**
	Decode 1 Data String	**SINT[160]**	0 - 1279	**160 byte**	**64**
128 bit	**Decode 2 Cycle Report**			**16 byte**	**224**
	Decode Location Top	**UINT**	0 - 15	2 byte	
	Decode Location Left	**UINT**	0 - 15	2 byte	
	Decode Location Height	**UINT**	0 - 15	2 byte	
	Decode Location Width	**UINT**	0 - 15	2 byte	
	Code Type	**DINT**		**4 Byte**	**232**
	Aztec Code		0	1 bit	
	Micro QR Code		1	1 bit	
	Postal Code		2	1 bit	
	Code 39		3	1 bit	
	Codabar		4	1 bit	
	Interleaved 2 of 5		5	1 bit	
	UPC EAN		6	1 bit	
	Code 128 EAN 128		7	1 bit	
	Code 93		8	1 bit	
	PDF417		9	1 bit	
	Pharma Code		10	1 bit	
	Data Matrix		11	1 bit	
	QR Code		12	1 bit	
	BC412		13	1 bit	
	GS1 Databar		14	1 bit	
	GS1 Databar Limited		15	1 bit	
	GS1 Databar Expanded		16	1 bit	
	Micro PDF		17	1 bit	
	Composite		18	1 bit	
	Dot Code		19	1 bit	

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
	Reserved		20 - 31	12 bit	
	Pixels per Element	**REAL**	0 - 31	**4 byte**	**236**
32 bit	**Decode 2 Data Length**	**DINT**	0 - 31	**4 byte**	**240**
	Decode 2 Data String	**SINT[72]**	0 - 575	**72 byte**	**244**
128 bit	**Decode 3 Cycle Report**			**16 byte**	**316**
	Decode Location Top	**UINT**	0 - 15	2 byte	
	Decode Location Left	**UINT**	0 - 15	2 byte	
	Decode Location Height	**UINT**	0 - 15	2 byte	
	Decode Location Width	**UINT**	0 - 15	2 byte	
	Code Type	**DINT**		**4 Byte**	**324**
	Aztec Code		0	1 bit	
	Micro QR Code		1	1 bit	
	Postal Code		2	1 bit	
	Code 39		3	1 bit	
	Codabar		4	1 bit	
	Interleaved 2 of 5		5	1 bit	
	UPC EAN		6	1 bit	
	Code 128 EAN 128		7	1 bit	
	Code 93		8	1 bit	
	PDF417		9	1 bit	
	Pharma Code		10	1 bit	
	Data Matrix		11	1 bit	
	QR Code		12	1 bit	
	BC412		13	1 bit	
	GS1 Databar		14	1 bit	
	GS1 Databar Limited		15	1 bit	
	GS1 Databar Expanded		16	1 bit	
	Micro PDF		17	1 bit	
	Composite		18	1 bit	
	Dot Code		19	1 bit	
	Reserved		20 - 31	12 bit	
	Pixels per Element	**REAL**	0 - 31	**4 byte**	**328**
32 bit	**Decode 3 Data Length**	**DINT**	0 - 31	**4 byte**	**332**
	Decode 3 Data String	**SINT[72]**	0 - 575	**72 byte**	**336**
128 bit	**Decode 4 Cycle Report**			**16 byte**	**408**
	Decode Location Top	**UINT**	0 - 15	2 byte	
	Decode Location Left	**UINT**	0 - 15	2 byte	
	Decode Location Height	**UINT**	0 - 15	2 byte	
	Decode Location Width	**UINT**	0 - 15	2 byte	
	Code Type	**DINT**		**4 Byte**	**416**
	Aztec Code		0	1 bit	
	Micro QR Code		1	1 bit	
	Postal Code		2	1 bit	
	Code 39		3	1 bit	
	Codabar		4	1 bit	

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
	Interleaved 2 of 5		5	1 bit	
	UPC EAN		6	1 bit	
	Code 128 EAN 128		7	1 bit	
	Code 93		8	1 bit	
	PDF417		9	1 bit	
	Pharma Code		10	1 bit	
	Data Matrix		11	1 bit	
	QR Code		12	1 bit	
	BC412		13	1 bit	
	GS1 Databar		14	1 bit	
	GS1 Databar Limited		15	1 bit	
	GS1 Databar Expanded		16	1 bit	
	Micro PDF		17	1 bit	
	Composite		18	1 bit	
	Dot Code		19	1 bit	
	Reserved		20 - 31	12 bit	
	Pixels per Element	**REAL**	0 - 31	**4 byte**	**420**
32 bit	**Decode 4 Data Length**	**DINT**	0 - 31	**4 byte**	**424**
	Decode 4 Data String	**SINT[72]**	0 - 575	**72 byte**	**428**

N Decode Input (Instance ID: 105)

N Decode Input supports any number of multiple symbol readings. Use this when you want to execute arbitrary number of multiple symbol readings and query symbol information such as symbol position coordinates for each symbol. Since the data structure of the Read result is a variable length up to a maximum of 456 bytes, the user needs to access the data such as the read character string by referencing the data offset value.

ATTENTION:
If **Format Output is NOT enabled on the reader**, the **DECODE 'X' DATA**, **DECODE 'X' LENGTH** and **DECODE 'X' CYCLE REPORT** ("Read Data Structure" from RAW INPUT DATA, please see next table) will reflect the data of each code, grouped all together, i.e., **DECODE 1 DATA, DECODE 1 LENGTH** and **DECODE 1 CYCLE REPORT** related to the same code.
If **Format Output is ENABLED on the reader**, the **DECODE 'X' DATA** and **DECODE 'X' LENGTH** will reflect the format defined by user. However, the **DECODE 'X' CYCLE REPORT** cannot be affected by the format defined by user, keeping its information in the same way as if the **Format Output is NOT enabled on the reader**.
Summarizing:

- If **Format Output is ENABLED on the reader**, the **DECODE 'X' CYCLE REPORT** cannot be related to such code.
- If you need them, to guarantee the correct information, enable "Output Coordinates" and "Pixel Per Element" and "Include Symbology Identifier" options (I/O tab) to be added to the **Format Output**, and do consider only the information presented by **DECODE 'X' DATA** and **DECODE 'X' LENGTH**, ignoring the information presented by **DECODE 'X' CYCLE REPORT**.

N Decode Input Member Structure

Member Name	Size (Bytes)
RESERVED	1
RESERVED	1
RESERVED	1
RESERVED	1
DEVICE STATUS	4
RESERVED	4
COUNTERS	24
READ CYCLE REPORT STATIC MEMBERS	8
RAW INPUT DATA	456

Total Size: 500 Bytes

Member Description

- **Device Status**
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as
- **Counters**
  MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as
- **Read Cycle Report**

Read Cycle Report	Size (Bytes)
Capture Time	2
Decode Time	2
Total Read Cycle Time	2
Number of Decodes in Read Cycle	1
Number of Decode Reports	1

- Capture Time
The time required for image capture of the image that had the successful decode. If none of the images were decoded, this will be the first image capture time. (milliseconds)
- Decode Time
The time required for decoding a symbol. If none of the images were decoded, then the time of the first image decode will be reported. (milliseconds)
- Total Read Cycle Time
The total time taken to read symbols. This encompasses the total time of image capture, decoding and overhead. (milliseconds)
- Number of Decodes in Read Cycle
The total number of detected symbols in the Read Cycle.
- Number of Decode Reports
The total number of Decode information data related to detected symbols.
Equal to the total number of detected symbols in the Read Cycle.

- **Raw Input Data**
  Variable length Read data is stored.

Read Data Structure	Size (Bytes)	Offset
Offset of Report 1	4	
Offset of Report 2	4	
…	...	
Offset of Report N	4	

Read Data Structure	Size (Bytes)	Offset
Decode Cycle Report 1	16	Offset 1
Decode Length 1	4	
Decode Data 1	Variable length	
Decode Cycle Report 2	16	Offset 2
Decode Length 2	4	
Decode Data 2	Variable length	
…	...	
Decode Cycle Report N*1	16	Offset N
Decode Data Length N*1	4	
Decode Data String N*1	Variable length	

*1. N is the value output for Number of Decodes in Read Cycle.

- Offset of Report (n)

This is the offset value from the Start Address for Raw Input Data to the address where the nth Read result is stored.

- Decode Cycle Report (n)

Information of the nth Symbol.

MXL/SLC Input (Instance ID: 102) on page A-7 - Same structure as Decode Cycle Report

- Decode Data Length (n)

Stores the number of characters that comprise the nth symbol.

- Decode Data String (n)

Stores the Read string of the nth symbol.

Memory Allocation

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
32 bit	**Reserved**	**SINT**		**1 Byte**	**0**
	Reserved	**SINT**		**1 Byte**	
	Reserved	**SINT**		**1 Byte**	
	Reserved	**SINT**		**1 byte**	
32 bit	**DeviceStatus**	**DINT**		**4 Byte**	**4**
	Run Mode		0	1 bit	
	Trigger Acknowledged		1	1 bit	
	Exposure Done		2	1 bit	
	Decoding		3	1 bit	
	Data is Ready		4	1 bit	
	Read Cycle Pass		5	1 bit	
	Read Cycle Fail		6	1 bit	
	General Fault		7	1 bit	
	Matchcode Master Label Trained		8	1 bit	
	Matchcode Enabled		9	1 bit	
	Image Sensor Calibrating		10	1 bit	
	Image Sensor Calibration Complete		11	1 bit	
	Training		12	1 bit	
	Training Complete		13	1 bit	
	Optimizing		14	1 bit	
	Optimizing Complete		15	1 bit	

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
	Auto Image Photometry Enabled		16	1 bit	
	Auto Image Photometry Complete		17	1 bit	
	Reserved		18	1 bit	
	Reserved		19	1 bit	
	Reserved		20	1 bit	
	BufferOverflow		21	1 bit	
	Reserved		22 - 31	10 bit	
32 bit	**Reserved**	**DINT**		**4 Byte**	**8**
192 bit	**Counters**	**UDINT[6]**		**24 byte**	**12**
	No Read Read Cycle Counter		0 - 31	4 byte	
	Mismatch per Read Cycle Counter		0 - 31	4 byte	
	No Read Counter		0 - 31	4 byte	
	Trigger Counter		0 - 31	4 byte	
	Matchcode Counter		0 - 31	4 byte	
	Mismatch Counter		0 - 31	4 byte	
64 bit	**Read Cycle Report**	**UINT[4]**		**8 byte**	**36**
	Capture Time		0 - 15	2 byte	
	Decode Time		0 - 15	2 byte	
	Total Read Cycle Time		0 - 15	2 byte	
	Number of Decodes in Read Cycle		0 - 7	1 byte	
	Number of Decode Reports		0 - 7	1 byte	
	RAW Input Data			**456 byte**	**44 byte**

Output (Instance ID: 197)

The Output can send several commands to the code reader.
This assembly is used with MXL/SLC Input (ID: 102), 1 Decode Input (ID: 103), 4 Decode Input (ID: 104), N Decode Input (ID: 105).

Output Member Structure

Member Name	Size (Bytes)
COMMANDS	4

Total Size: 4 Bytes

Member Description

- **Commands**

  An explanation of commands that can be sent to the code reader.

Bit	Command
0	Run Mode
1	Trigger
2	Enable Matchcode
3	Reset General Fault
4	Clear No Read Read Cycle Count
5	Clear Mismatch Read Cycle Count
6	Clear No Read Count

Bit	Command
7	Clear Trigger Count
8	Clear Matchcode Count
9	Clear Mismatch Count
10 - 31	Reserved

- Run Mode

Enables / Disables Read Cycle. Immediately after the code reader is started, via serial command or WebLink$_{HH}$, Read Cycle will be enabled regardless of this command.

0 = Read Cycle Disabled. No trigger can be accepted. However, other commands can be executed.

1 = Enables Read Cycle.

- Trigger

Executes Read. The code reader recognizes this bit changing from 0 to 1 as the rising edge of the trigger and its change from 1 to 0 as the falling edge of the trigger.

- Enabled Matchcode

Enable / Disable the Matchcode function. Immediately after the code reader is started, the previously saved setting is in effect regardless of this command.

0 = Disable Matchcode function.

1= Enable Matchcode function.

- Reset General Fault

If an error occurs on the code reader, after resolving the error, this bit is used to reset the Fault Code Area of the Input Assembly.

- Clear No Read Read Cycle Count

Resets the No Reads per Read Cycle counter to 0.

- Clear Mismatch Read Cycle Count

Resets the Mismatch per Read Cycle counter to 0.

- Clear No Read Count

Resets the No Reads counter to 0.

- Clear Triger Count

Resets the Trigger counter to 0.

- Clear Matchcode Count

Resets the Matchcode counter to 0.

- Clear Mismatch Count

Resets the Mismatch counter to 0.

Memory Allocation

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
32 bit	**Commands**	**DINT**		**4 Byte**	**0**
	Run Mode		0	1 bit	
	Trigger		1	1 bit	
	Enable Matchcode		2	1 bit	
	Reset General Fault		3	1 bit	
	Clear No Read Read Cycle Count		4	1 bit	
	Clear Mismatch Read Cycle Count		5	1 bit	
	Clear No Read Count		6	1 bit	
	Clear Trigger Count		7	1 bit	
	Clear Matchcode Count		8	1 bit	
	Clear Mismatch Count		9	1 bit	
	Reserved		10 - 31	22 bit	

Output (Legacy) (Instance ID: 198)

The Output (Legacy) can be used to send multiple commands and Command Echo for fixed data to the code reader.
This assembly is used with the Small Input (ID: 100), Large Input (ID: 101).

Output (Legacy) Member Structure

Member Name	Size (Bytes)
USER-DEFINED TAGS	4
COMMANDS	4
RESERVED	4

Total Size: 12 Bytes

Member Description

- **User-Defined Tags**
  Data set for this Member is echoed back to the USER-DEFINED TAG ECHO area of the Small Input or the Large Input. It is used when you want to uniquely identify multiple code readers.
- **Commands**
  An explanation of commands that can be sent to the code reader.

Bit	Command
0	Trigger
1	New Master
2 - 7	Reserved
8	Disable Scanning
9 - 15	Reserved
16	Clear Read Cycle Report and Counters
17 - 31	Reserved

- Trigger
Executes Read. The code reader recognizes this bit changing from 0 to 1 as the rising edge of the trigger and its change from 1 to 0 as the falling edge of the trigger.

- New Master

When this bit is ON, the next Read result is registered as the Master Symbol.

- Disable Scanning

Enables / Disables Read Cycle.

0 = Read Cycle Enabled.

1 = Read Cycle Disabled. However, other commands can be executed.

- Clear Read Cycle Report and Counters

Resets the Counter and Decode data area in the Small or Large Input assembly. The Decode Data Length of the string is set to 0, and the first byte of the Decode Data String area is set to "\0" (null), which is the standard string terminator.

Memory Allocation

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
32 bit	**User Defined Tag**	**DINT**		**4 Byte**	**0**
	UserTag_1		0	1 bit	
	UserTag_2		1	1 bit	
	UserTag_3		2	1 bit	
	UserTag_4		3	1 bit	
	UserTag_5		4	1 bit	
	UserTag_6		5	1 bit	
	UserTag_7		6	1 bit	
	UserTag_8		7	1 bit	
	UserTag_9		8	1 bit	
	UserTag_10		9	1 bit	
	UserTag_11		10	1 bit	
	UserTag_12		11	1 bit	
	UserTag_13		12	1 bit	
	UserTag_14		13	1 bit	
	UserTag_15		14	1 bit	
	UserTag_16		15	1 bit	
	UserTag_17		16	1 bit	
	UserTag_18		17	1 bit	
	UserTag_19		18	1 bit	
	UserTag_20		19	1 bit	
	UserTag_21		20	1 bit	
	UserTag_22		21	1 bit	
	UserTag_23		22	1 bit	
	UserTag_24		23	1 bit	
	UserTag_25		24	1 bit	
	UserTag_26		25	1 bit	
	UserTag_27		26	1 bit	
	UserTag_28		27	1 bit	
	UserTag_29		28	1 bit	
	UserTag_30		29	1 bit	
	UserTag_31		30	1 bit	
	UserTag_32		31	1 bit	

	Member Name	Data Type	Bit Number	Data Length	Byte Offset
32 bit	**Commands**	**DINT**		**4 Byte**	**4**
	Trigger		0	1 bit	
	New Master		1	1 bit	
	Reserved		2 - 7	6 bit	
	Disable Scanning		8	1 bit	
	Reserved		9 - 15	7 bit	
	Clear Read Cycle Report and Counters		16	1 bit	
	Reserved		17 - 31	15 bit	
32 bit	**Reserved**	**DINT**		**4 Byte**	**8**

A-3 PROFINET - V460-H Input and Output Modules

This section lists the commands you can use with the V460-H and the PROFINET industrial protocol.

Function Blocks Library and Sample Program for Omron Controllers are available for download. Please, visit Omron website for Function Blocks Library and Sample Program for additional PLC / Controllers.

A-3-1 Module Types

There are 7 Input Modules and 2 Output Modules. The layout of each module and the definitions of the data in them will be shown in this appendix.

Model Item ID	Name	Total Size in Bytes	PROFINET Slot Allowed	PNT21 Supported
100	Small Legacy Input Module	84	1	Yes
101	Big Legacy Input Module	176	1	Yes
102	MXL Input Module	248	1	Yes
103	1 Decode Input Module	500	1	No
104	4 Decode Input Module	500	1	No
105	N Decode Input Module	500	1	No
106	Omron Decode Input Module	442	1	Yes
197	Premier Output Module	4	2	Yes
198	Legacy Output Module	12	2	Yes

Additional Information

The maximum Input CR size for the PNT21 is 450 bytes.

Input/Output Modules

All Input/Output modules and module descriptions are the same as in *A-2 EtherNet/IP Specifications* on page A-3, except the following new addition: Omron Decode Input Module.

● Omron Decode Input ID:106

This input is identical to the 1 Decode Input (103), except for Decode Data String being 378 bytes.

SHORT DESCRIPTION		SIZE (BYTES)
MODULE HEADER		
	RESERVED	1
	RESERVED	1
	RESERVED	1
	RESERVED	1
DEVICE STATUS		4
RESERVED		4
COUNTERS		24
READ CYCLE REPORT		8
DECODE CYCLE REPORT		
	DECODE CYCLE REPORT TABLE	16
	DECODE DATA LENGTH	4
	DECODE DATA STRING	378

Total Size: 442 Bytes

A-3-2 Data Types

User Data Types for Input/Output Modules Table

MODULE NAME	USER DATA TYPE NAME
SMALL LEGACY INPUT MODULE	• Input_Legacy_Small • Legacy_UserTag_Echo • Legacy_Command_Echo
BIG LEGACY INPUT MODULE	• Input_Legacy_Big • Legacy_UserTag_Echo • Legacy_Command_Echo • Legacy_Device_Status
MXL INPUT MODULE	• Input_MXL_Decode • Input_Header • ReadCycle_Report • Input_MXL_Decode_Report
1 DECODE INPUT MODULE	• Input_1_Decode • Input_Header • ReadCycle_Report • Decode_Report_436Bytes
4 DECODE INPUT MODULE	• Input_4_Decode • Input_Header • ReadCycle_Report • Decode_Report_160Bytes • Decode_Report_72Bytes
N DECODE INPUT MODULE	• Input_N_Decode • Input_N_Header • Input_N_ReadCycle_Report • Decode_Report_436Bytes
LEGACY OUTPUT MODULE	• Output_Legacy • Legacy_User_Defined_Tags • Legacy_Cmds
PREMIER OUTPUT MODULE	• Premier_Cmds

A-3-3 PROFINET Base Information

Device Identity

The PROFINET device identity information is as follows:

- **Vendor ID**
  The Vendor ID is 0x0257.
- **Device ID**
  Refer to the *PROFINET Files by Firmware Version* on page A-32 table below to determine the correct Device ID.
- **Vendor Name**
  The Vendor Name is OMRON MICROSCAN SYSTEMS, INC.
- **Device Function**
  The Device Function is:
  - MainFamily = Ident Systems
  - ProductFamily = V460-H

GSDML File

Refer to the *PROFINET Files by Firmware Version* on page A-32 table below to determine the correct GSDML file for your device.

PROFINET Files by Firmware Version

Product	Firmware Version	GSDML File	Version	Device ID
V460-H	1.0.0.xxxx	GSDML-V2.35-OmronMicroscanSystemsInc-V460-H-20210827.xml	V2.35	0x3414

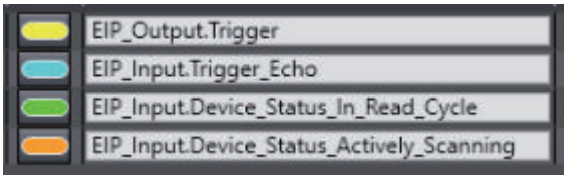
Connection Properties: RT Cyclic Messaging

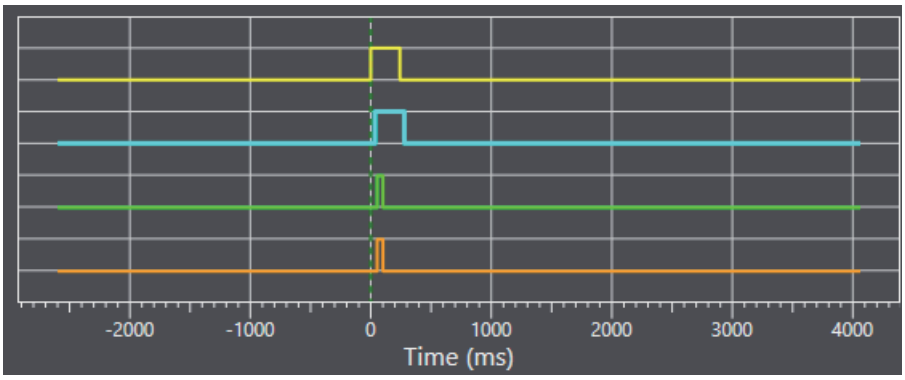
Cycle Time: 8 ms
Definition: The GSD file contains element MinDeviceInterval, which is 256. Multiply this by 31.25 µs. This is the cycle time. See the PROFINET GSDML specification for more information.

A-3-4 Timing Diagrams

Big Legacy Input Module

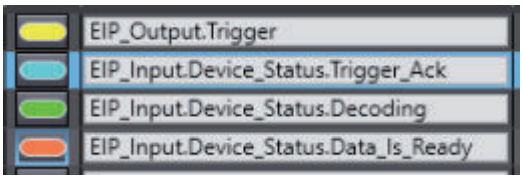
As the Legacy Input modules have very little user feedback, timing is limited.

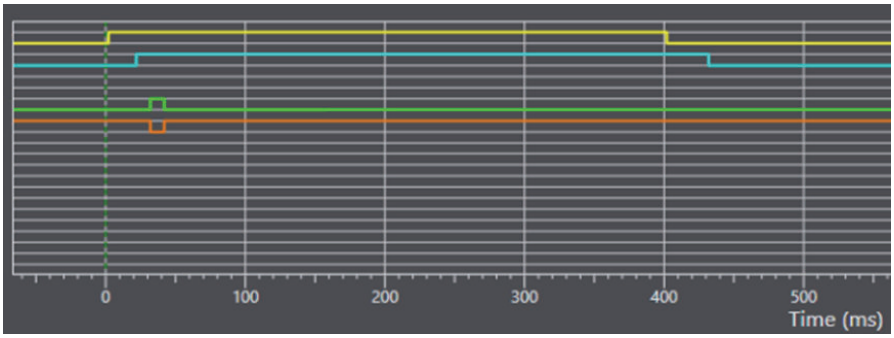




Omron Decode Input Module

- In this example, the trigger is set high for 400ms.
- The trigger was acknowledged 20ms after trigger high and stays high for 30ms after trigger low.
- Decoding is complete and data is ready 39ms after trigger start.





OMRON Corporation **Industrial Automation Company**

Kyoto, JAPAN **Contact : www.ia.omron.com**

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-3011

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-6023-0333 Fax: (86) 21-5037-2388

Authorized Distributor:

Cat. No. Z462-E1-01 1023