PLC Interface Manual

IND700

Weighing Terminal

METTLER TOLEDO

IND700 Weighing Terminal

METTLER TOLEDO Service

Essential Services for Dependable Performance of Your IND700 Weighing Terminal

Congratulations on choosing the quality and precision of METTLER TOLEDO. Proper use of your new equipment according to this Manual and regular calibration and maintenance by our factory-trained service team ensures dependable and accurate operation, protecting your investment. Contact us about a service agreement tailored to your needs and budget. Further information is available at www.mt.com/service.

There are several important ways to ensure you maximize the performance of your investment:

1. **Register your product**: We invite you to register your product at www.mt.com/productregistration so we can contact you about enhancements, updates and important notifications concerning your product.
2. **Contact METTLER TOLEDO for service**: The value of a measurement is proportional to its accuracy – an out of specification scale can diminish quality, reduce profits and increase liability. Timely service from METTLER TOLEDO will ensure accuracy and optimize uptime and equipment life.
    a. **Installation, Configuration, Integration and Training**: Our service representatives are factory-trained, weighing equipment experts. We make certain that your weighing equipment is ready for production in a cost effective and timely fashion and that personnel are trained for success.
    b. **Initial Calibration Documentation**: The installation environment and application requirements are unique for every industrial scale so performance must be tested and certified. Our calibration services and certificates document accuracy to ensure production quality and provide a quality system record of performance.
    c. **Periodic Calibration Maintenance**: A Calibration Service Agreement provides on-going confidence in your weighing process and documentation of compliance with requirements. We offer a variety of service plans that are scheduled to meet your needs and designed to fit your budget.
    d. **GWP® Verification**: A risk-based approach for managing weighing equipment allows for control and improvement of the entire measuring process, which ensures reproducible product quality and minimizes process costs. GWP (Good Weighing Practice), the science-based standard for efficient life-cycle management of weighing equipment, gives clear answers about how to specify, calibrate and ensure accuracy of weighing equipment, independent of make or brand.

.

FCC Notice

This device complies with Part 15 of the FCC Rules and the Radio Interference Requirements of the Canadian Department of Communications. Operation is subject to the following conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his or her expense.

Warnings and Cautions

- READ this manual BEFORE operating or servicing this equipment and FOLLOW these instructions carefully.
- SAVE this manual for future reference.

Compliance information

National approval documents, e.g. the FCC Supplier Declaration of Conformity, are available online and/or included in the packaging.

www.mt.com/ComplianceSearch

Manuals download

Please scan the QR code below and download from www.mt.com/IND700-downloads.

⚠ WARNING

FOR CONTINUED PROTECTION AGAINST SHOCK HAZARD CONNECT THE TERMINAL TO PROPERLY GROUNDED OUTLET ONLY. DO NOT REMOVE THE GROUND PRONG.

⚠ WARNING

ONLY PERMIT QUALIFIED PERSONNEL TO SERVICE THE TERMINAL. EXERCISE CARE WHEN MAKING CHECKS, TESTS AND ADJUSTMENTS THAT MUST BE MADE WITH POWER ON. FAILING TO OBSERVE THESE PRECAUTIONS CAN RESULT IN BODILY HARM AND/OR PROPERTY DAMAGE.

⚠ WARNING

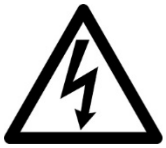
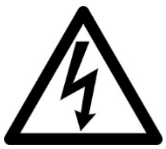
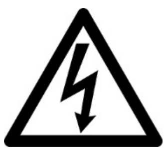
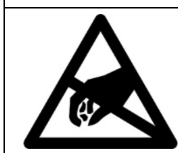
DO NOT INSTALL, DISCONNECT OR PERFORM ANY SERVICE ON THIS EQUIPMENT BEFORE POWER HAS BEEN SWITCHED OFF AND THE AREA HAS BEEN SECURED AS NON-HAZARDOUS BY PERSONNEL AUTHORIZED TO DO SO BY THE RESPONSIBLE PERSON ON-SITE.

⚠ WARNING

NOT ALL VERSIONS OF IND700 ARE DESIGNED FOR USE IN HAZARDOUS (EXPLOSIVE) AREAS. REFER TO THE DATA PLATE OF THE IND700 TO DETERMINE IF A SPECIFIC TERMINAL IS APPROVED FOR USE IN AN AREA CLASSIFIED AS HAZARDOUS BECAUSE OF COMBUSTIBLE OR EXPLOSIVE ATMOSPHERES. TERMINALS THAT ARE NOT FACTORY LABELED AS DIVISION 2 OR EUROPEAN CATEGORY 3 APPROVED MUST NOT BE INSTALLED IN A DIVISION 2 OR ZONE 2/22 ENVIRONMENT.

⚠ WARNING

THE INTERNAL DISCRETE I/O RELAY OPTIONS MUST NOT BE USED IN AREAS CLASSIFIED AS HAZARDOUS BECAUSE OF COMBUSTIBLE OR EXPLOSIVE ATMOSPHERES. FAILURE TO

	COMPLY WITH THIS WARNING COULD RESULT IN BODILY HARM AND/OR PROPERTY DAMAGE.
	**⚠ WARNING** **WHEN THIS EQUIPMENT IS INCLUDED AS A COMPONENT PART OF A SYSTEM, THE RESULTING DESIGN MUST BE REVIEWED BY QUALIFIED PERSONNEL WHO ARE FAMILIAR WITH THE CONSTRUCTION AND OPERATION OF ALL COMPONENTS IN THE SYSTEM AND THE POTENTIAL HAZARDS INVOLVED. FAILURE TO OBSERVE THIS PRECAUTION COULD RESULT IN BODILY HARM AND/OR PROPERTY DAMAGE.**
	**⚠ WARNING** **ONLY THE COMPONENTS SPECIFIED ON THE IND700 DOCUMENTATION CD CAN BE USED IN THIS TERMINAL. ALL EQUIPMENT MUST BE INSTALLED IN ACCORDANCE WITH THE INSTALLATION INSTRUCTIONS DETAILED IN THE INSTALLATION MANUAL. INCORRECT OR SUBSTITUTE COMPONENTS AND/OR DEVIATION FROM THESE INSTRUCTIONS CAN IMPAIR THE SAFETY OF THE TERMINAL AND COULD RESULT IN BODILY HARM AND/OR PROPERTY DAMAGE.**
	**⚠ CAUTION** **BEFORE CONNECTING/DISCONNECTING ANY INTERNAL ELECTRONIC COMPONENTS OR INTERCONNECTING WIRING BETWEEN ELECTRONIC EQUIPMENT ALWAYS REMOVE POWER AND WAIT AT LEAST THIRTY (30) SECONDS BEFORE ANY CONNECTIONS OR DISCONNECTIONS ARE MADE. FAILURE TO OBSERVE THESE PRECAUTIONS COULD RESULT IN DAMAGE TO OR DESTRUCTION OF THE EQUIPMENT AND/OR BODILY HARM.**
	***NOTICE*** **OBSERVE PRECAUTIONS FOR HANDLING ELECTROSTATIC SENSITIVE DEVICES.**

Disposal of Electrical and Electronic Equipment

In conformance with the European Directive 2012/19/EC on Waste Electrical and Electronic Equipment (WEEE) this device may not be disposed of in domestic waste. This also applies to countries outside the EU, per their specific requirements.

Please dispose of this product in accordance with local regulations at the collecting point specified for electrical and electronic equipment.

If you have any questions, please contact the responsible authority or the distributor from which you purchased this device.

Should this device be passed on to other parties (for private or professional use), the content of this regulation must also be related.

Thank you for your contribution to environmental protection.

Contents

Contents

1 Introduction

1.1. PLC Interfaces in the IND700

The IND700 supports two PLC interfaces:

- EtherNet/IP
- PROFINET

The protocol has two primary data types – cyclic data and asynchronous data (also known as acyclic or explicit messaging).

The interfaces use the same physical hardware (the interface board), running different firmware.

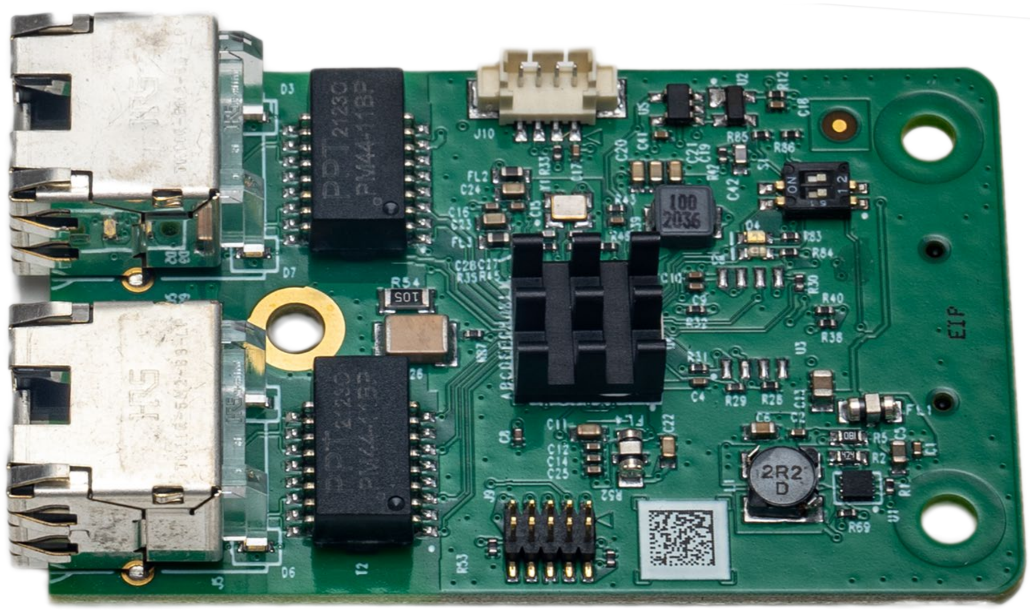


Figure 1-1: Industrial Ethernet Option Board

The board is installed in the IND700 in a dedicated PLC slot on the motherboard, indicated below:

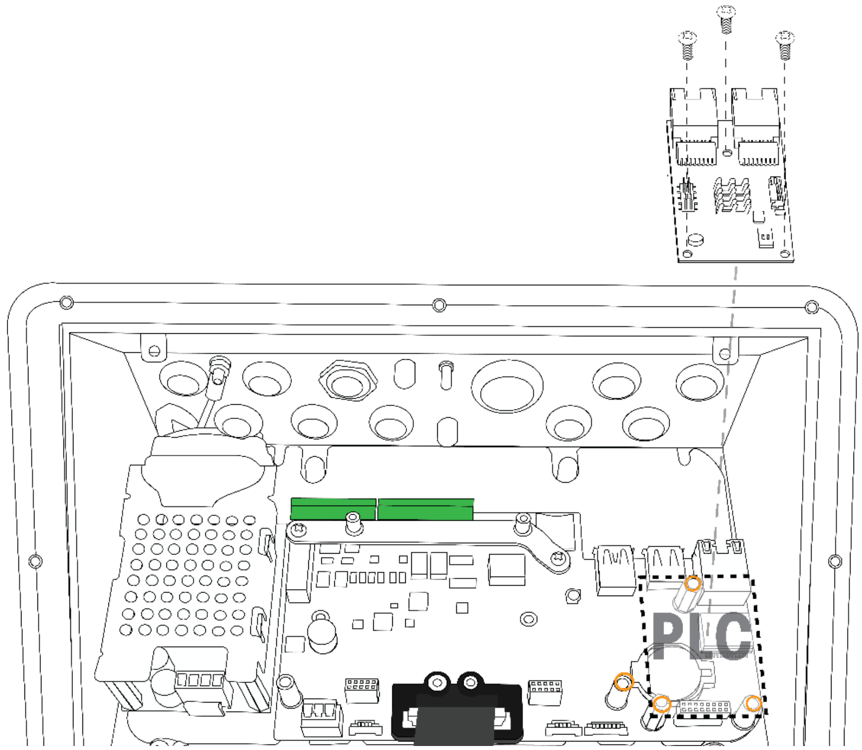


Figure 1-2: PLC Interface Location in IND700

1.2. SAI Overview

The Standard Automation Interface (SAI) is a protocol designed to exchange data between METTLER TOLEDO devices and automation systems. The goals of this standard are to provide: 1) a common data layout for load cells, terminals, and other devices regardless of the physical interface or automation network used, 2) a single protocol for the convenience of automation integrators, control system programmers, and our automation customers, and 3) a tiered approach to create a flexible protocol for diverse devices.

The protocol has two primary data types – cyclic data and asynchronous data (also known as acyclic or explicit messaging).

2 EtherNet/IP

2.1. Preface

There are minor differences in the Floating Point polled data between the terminals, so care should be taken to use the appropriate PLC data format guide for each terminal family. This chapter describes connections and setup that are specific to the EtherNet/IP option for IND700. The formats of the data that is transferred between the IND700 and the PLC are described in Appendix A and Appendix B.

2.2. EtherNet/IP Interface Board

Figure 2-1 shows an EtherNet/IP kit (part number 30726000) for the IND700. The board's part number is 30785339.

Figure 2-1: EtherNet/IP Interface Kit

2.3. Overview

EtherNet/IP, short for "EtherNet Industrial Protocol," is an open industrial networking standard that takes advantage of commercial, off-the-shelf EtherNet communication chips and physical media. This networking standard supports both implicit messaging (real-time I/O messaging) and explicit messaging (message exchange). The protocol is supported by the Open DeviceNet Vendor Association (ODVA).

EtherNet/IP utilizes commercial, off-the-shelf EtherNet hardware (for example, switches and routers) and is fully compatible with the Ethernet TCP/IP protocol suite. It uses the proven Control and Information Protocol (CIP) to provide control, configuration, and data collection capability.

The kit enables the IND700 terminal to communicate to Programmable Logic Controllers (PLCs) through direct connection to the EtherNet/IP network at either 10 or 100 MBPS speed. The kit consists of a backplane-compatible I/O module, mounting hardware, and a ferrite. Software to implement the data exchange resides in the IND700 terminal.

2.4. EtherNet/IP Characteristics

The EtherNet/IP Kit option has the following features:

- User-programmable IP addressing.
- Capability for bi-directional discrete mode communications (Class 1 Messaging) of weight or display increments, status, and control data between the PLC and the IND700.

2.4.1. Update Rates

The update rate for the EtherNet/IP interface varies depending on the terminal's configuration:

- HSALC 1 scale: 64 Hz
- HSALC 2 scales: 49 Hz
- POWERCELL 4 scales 14 Hz

2.4.2. Definition of Terms

Some terms (such as Target) used by the EtherNet/IP PLC application have a different sense from their use by the IND700 terminal. Table 2-1 provides definitions specific to EtherNet/IP.

Table 2-1: EtherNet/IP Definition of Terms

Term	Definition
Adapter Class	An Adapter Class product emulates functions provided by traditional rack-adapter products. This type of node exchanges real-time I/O data with a Scanner Class product. It does not initiate connections on its own.
Class 1 Messaging	In EtherNet/IP communication protocol scheduled (cyclic, or implicit) message transfer between a PLC and CIP Adapter Class device. Class 1 messages repeat on a continuous and deterministic basis.
Class 3 Messaging	In EtherNet/IP communication protocol unscheduled (acyclic, or explicit) message transfer between a PLC and CIP Adapter Class device. This is used by the IND700 for explicit messaging of specific data that has been requested for a one-time use, such as alarms, configuration data, or special commands such as for material transfer.

Term	Definition
Connected Messaging	A connection is a relationship between two or more application objects on different nodes. The connection establishes a virtual circuit between end points for transfer of data. Node resources are reserved in advance of data transfer and are dedicated and always available. Connected messaging reduces data handling of messages in the node. Connected messages can be Implicit or Explicit. See also Unconnected Messaging.
Connection Originator	Source for I/O connection or message requests. Initiates an I/O connection or explicit message connection.
Explicit Messaging	Explicit Messages (also known as Discrete, or Class 3, or Acyclic messages) can be sent as a connected or unconnected message. CIP defines an Explicit Messaging protocol that states the meaning of the message. This messaging protocol is contained in the message data. Explicit Messages provide a one-time transport of a data item. Explicit Messaging provide the means by which typical request/response oriented functions are performed (e.g. module configuration). These messages are typically point-to-point.
Implicit Messaging	Implicit Messages (also known as Class 1, or cyclic messages) are exchanged across I/O Connections with an associated Connection ID. The Connection ID defines the meaning of the data and establishes the regular/repeated transport rate and the transport class. No messaging protocol is contained within the message data as with Explicit Messaging. Implicit Messages can be point-to-point or multicast and are used to transmit application-specific I/O data. This term is used interchangeably with the term I/O Messaging.
I/O Client	Function that uses the I/O messaging services of another (I/O Server) device to perform a task. Initiates a request for an I/O message to the server module. The I/O Client is a Connection Originator.
I/O Messaging	Used interchangeably with the term Implicit Messaging.
I/O Server	Function that provides I/O messaging services to another (I/O Client) device. Responds to a request from the I/O Client. I/O Server is the target of the connection request.
Message Client	Function that uses the Explicit messaging services of another (Message Server) device to perform a task. Initiates an Explicit message request to the server device.
Message Server	Function that provides Explicit messaging services to another (Message Client) device. Responds to an Explicit message request from the Message Client.
Scanner Class	A Scanner Class product exchanges real-time I/O data with Adapter Class and Scanner Class products. This type of node can respond to connection requests and can also initiate connections on its own.
Target	Destination for I/O connection or message requests. Can only respond to a request, cannot initiate an I/O connection or message.

Term	Definition
Unconnected Messaging	Provides a means for a node to send message requests without establishing a connection prior to data transfer. More overhead is contained within each message and the message is not guaranteed destination node resources. Unconnected Messaging is used for non-periodic requests (e.g. network "Who" function). Explicit messages only. See also Connected Messaging.

2.4.3. Communications

The IND700 terminal utilizes component parts to ensure complete compatibility with the EtherNet/IP network. An IND700 terminal is recognized as a generic EtherNet/IP device by the PLC.

Each EtherNet/IP option connected to the EtherNet/IP network represents a physical IP Address. The connection is made via a RJ-45 connector on the option card (see Figure 2-1).

The wiring between the PLC and the IND700 EtherNet/IP connection uses EtherNet twisted pair cable. The cable installation procedures and specification including distance and termination requirements are the same as recommended by Allen-Bradley for the EtherNet/IP network.

The IND700 only uses Class 1 cyclic data for discrete data and Class 3 explicit messages for access to the IND700 Shared Data Variables. Explicit message blocks may be connected or unconnected; the PLC programmer must make this choice.

2.4.4. IP Address

Each EtherNet/IP option represents one physical IP Address. This address is chosen by the system designer, and then programmed into the IND700 terminal and PLC. The IND700 terminal's address is programmed at Communication > Industrial Network > EtherNet/IP-Modbus TCP in the terminal's setup menu. The IND700 IP Address entry must be unique for each IND700 terminal, and must not conflict with other devices on the network.

2.4.5. Supported Data Formats

The terminal's EtherNet/IP interface provides discrete data transfer and Class 1 messaging. Data transfer is accomplished via the PLC's cyclic messaging. The EtherNet/IP interface has its own logical IP address to send and receive information to and from the PLC continuously. The EtherNet/IP interface uses discrete data for its communication with PLCs.

Three formats of discrete data are available with the EtherNet/IP interface option: integer (the default), divisions and floating point.

Appendix A and B provide detailed information on data formats.

2.5. Data Definition

2.5.1. Assembly Instances of Class 1 Cyclic Communications

Class 1 cyclic communications is used for transfer of Discrete Data between the PLC and the IND700.

The PLC Input Assembly Instance is 101 (decimal). This instance is used for all Data Formats and data size requirements.

The PLC Output Assembly Instance is 100 (decimal). This instance is used for all Data Formats and data size requirements.

2.5.2. Discrete Data

Please refer to Appendix C, Common Data Features for a description of discrete data, and to Appendix A and Appendix B for a detailed description of the data available in each format, in order to determine which is most suitable.

2.5.3. Byte Order

For a general account of byte ordering, please refer to Appendix C, Common Data Features.

2.5.4. Message Slots

There may be up to 4 message slots for discrete data transfer, Class 1 messaging, in Integer, Divisions and Floating Point Data Formats. Each message slot represents the scale but may be controlled by the PLC to present different data in each message slot. The number of Message Slots is selected in the terminal's setup menu at Communication > Industrial Network > Data Format (Figure 2-2).

The integer and division formats provide two 16-bit words of input and two 16-bit words of output data per Slot. Each Message Slot's first input word provides scale weight data. The type of data displayed, such as Gross, Tare, etc., is selected by the PLC using the Message Slot's second output word bits 0, bit 1 and bit 2. Table 2-2 and Table 2-3 provide input and output usage information.

Table 2-2: EtherNet/IP PLC Integer and Division Input Data and Data Usage

Input Data to PLC				Output Data from PLC			
Word Offset	Description		Input Size	Output Size		Description	Word Offset
0	Integer Value	Msg Slot 1	2 Words (4 Bytes)	2 Words (4 Bytes)	Msg Slot 1	Load Integer Value	0
1	Scale Status					Command	1
2	Integer Value	Msg Slot 2	4 Words (8 Bytes)	4 Words (8 Bytes)	Msg Slot 2	Load Integer Value	2
3	Scale Status					Command	3
4	Integer Value	Msg Slot 3	6 Words (12 Bytes)	6 Words (12 Bytes)	Msg Slot 3	Load Integer Value	4
5	Scale Status					Command	5
6	Integer Value	Msg Slot 4	8 Words (16 Bytes)	8 Words (16 Bytes)	Msg Slot 4	Load Integer Value	6
7	Scale Status					Command	7

I/O Size Summary				
Message Slot	Words		Bytes	
	Input	Output	Input	Output
1	2	2	4	4
2	4	4	8	8
3	6	6	12	12
4	8	8	16	16

The floating point format provides four 16-bit words of input data and three 16-bit words of output data) per Message Slot. Table 2-3 provides details.

Table 2-3: EtherNet/IP PLC Floating Point Input Words

Input Data to PLC			
Word Offset	**Description**		**Input Size**
0	Command Response	Message Slot 1	4 Words (8 Bytes)
1	4-Byte Floating Point Value		
2			
3	Scale Status		
4	Command Response	Message Slot 2	8 Words (16 Bytes)
5	4-Byte Floating Point Value		
6			
7	Scale Status		
8	Command Response	Message Slot 3	12 Words (24 Bytes)
9	4-Byte Floating Point Value		
10			
11	Scale Status		
12	Command Response	Message Slot 4	16 Words (32 Bytes)
13	4-Byte Floating Point Value		
14			
15	Scale Status		

Output Data from PLC			
Output Size	**Description**		**Word Offset**
4 Words (8 Bytes)		Reserved	0
	Message Slot 1	Command	1
		4-Byte Floating Point Load Value	2
			3
7 Words (14 Bytes)	Message Slot 2	Command	4
		4-Byte Floating Point Load Value	5
			6
10 Words (20 Bytes)	Message Slot 3	Command	7
		4-Byte Floating Point Load Value	8
			9
13 Words (26 Bytes)	Message Slot 4	Command	10
		4-Byte Floating Point Load Value	11
			12

I/O Size Summary				
Message Slot	**Words**		**Bytes**	
	Input	**Output**	**Input**	**Output**
1	4	4	8	8
2	8	7	16	14
3	12	10	24	20
4	16	13	32	26

2.5.5. Floating Point

For a general account of Floating Point operation, data format and compatibility, please refer to Appendix B, **Floating Point Format**.

2.5.5.1. Data Integrity

The IND700 uses two data integrity bits to maintain data integrity when communicating with the PLC. One bit is in the beginning word of the data; the second is in the ending byte of the data for a scale slot. The PLC program must verify that both data integrity bits have the same polarity for the data in the scale slot to be valid. There is a possibility that the PLC program will see several consecutive invalid reads when the terminal is freely sending weigh updates to the PLC, if the PLC program detects this condition, it should send a new command to the terminal.

The method of handling string and floating point data varies between Allen-Bradley PLC generations.

2.6. Shared Data Mode

The Shared Data mode PLC communications is provided using CIP explicit (Class 3) messages.

The IND700 Shared Data Reference manual lists the Shared Data Variables available to EtherNet/IP. This document also includes the hex Class Code, Instance and Attribute for the shared data. The PLC must use Get Attribute Single (service code e) to read a Shared Data Variable and Set Attribute Single (service code 10) to write a Shared Data Variable.

The **IND700 Shared Data Reference** can be downloaded from www.mt.com/IND700-downloads..

2.7. Controlling Discrete I/O Using a PLC Interface

Please refer to Appendix A, **Common Data Features**.

2.8. Software Setup

When the IND700 terminal detects the presence of a EtherNet/IP Kit option board, the EtherNet/IP parameters are enabled in a Setup program block at **Communication > Industrial Network**. Figure 2-2 shows a typical PLC Menu screen in setup.

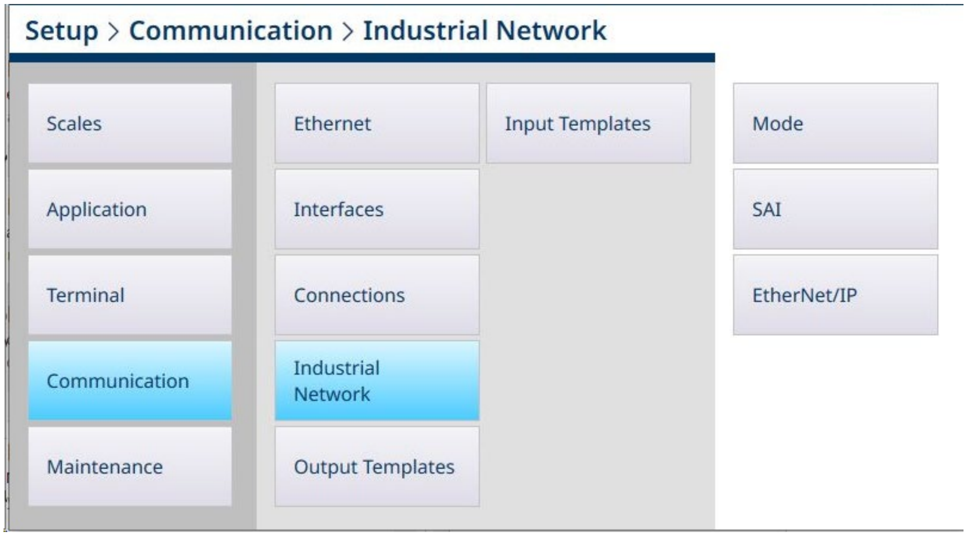


Figure 2-2: PLC Setup Menus

The **Mode** menu permits the selection of either **SAI** or **Custom**. The second menu is either **SAI** or **Custom** depending on the **Mode** selection. The third PLC menu configures the specific PLC interface detected – the example above shows a EtherNet Interface.

2.8.1. EtherNet/IP Setup Menu

Figure 2-3: EtherNet/IP Setup Menu

The EtherNet/IP interface IP address can be set either using **DHCP**, or by entering values manually in the **IP Address**, **Subnet Mask** and **Gateway Address** fields.

2.8.2. SAI Block Basic Configuration

If the PLC **Mode** is set to **Custom**, the following menu will be available:

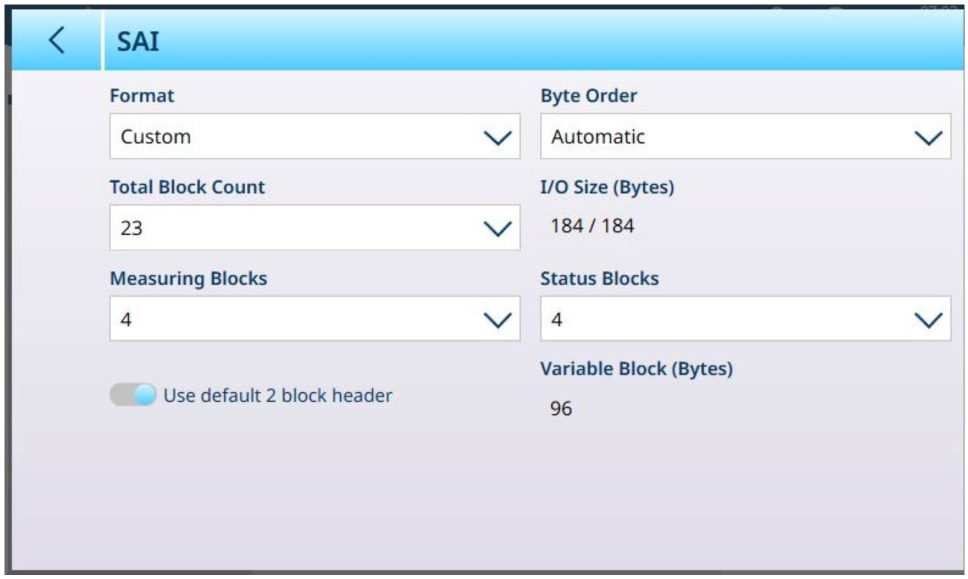


Figure 2-4: PLC SAI Setup Menu

2.8.3. Custom Menu Basic Configuration

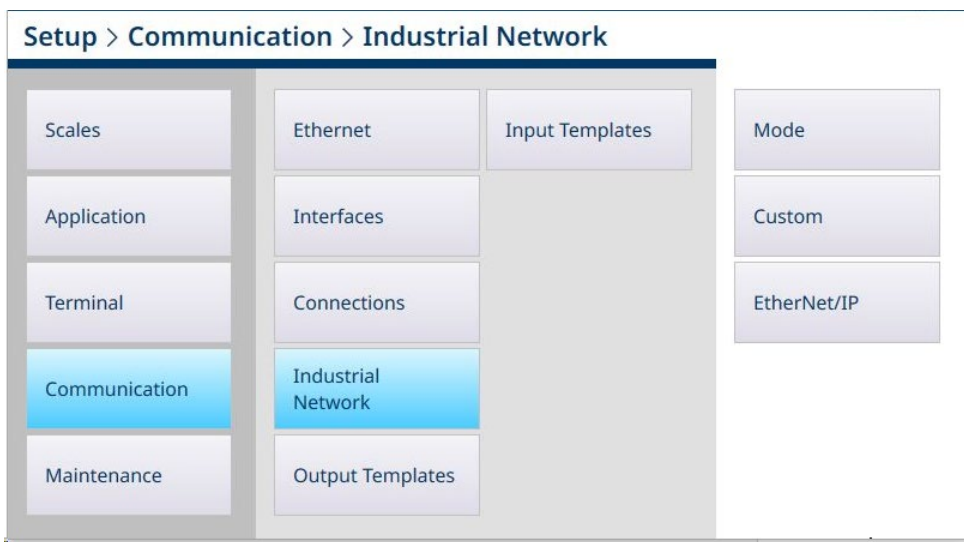


Figure 2-5: Industrial Network Menus, Mode = Custom

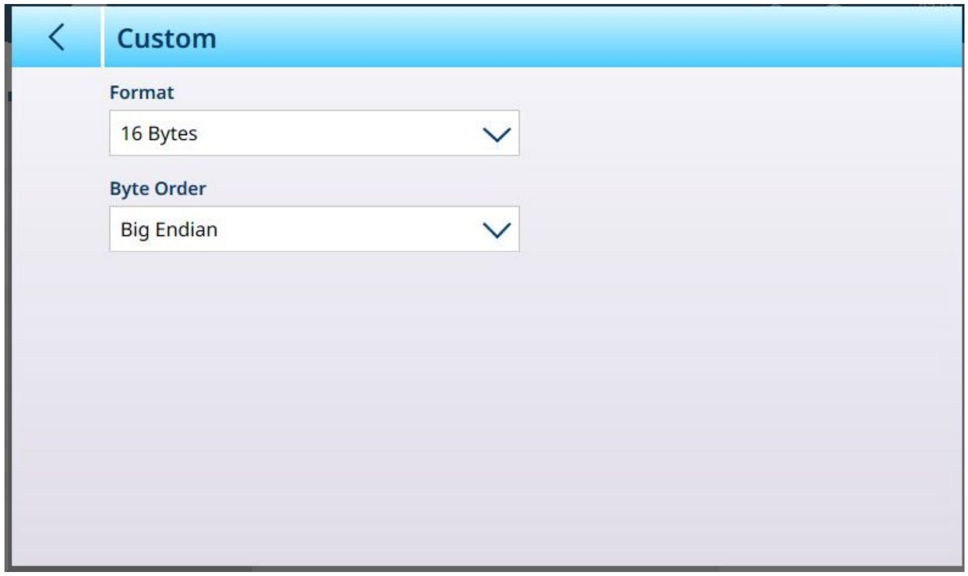


Figure 2-6: Custom Configuration Options

2.9. Troubleshooting

- Note: Some PLCs, such as Micrologix and SLC PLCs cannot exchange cyclic (class 1) messages. If these PLCs are used, they must use Explicit (class 3) Messaging to communicate with the IND700.

If the IND700 does not communicate with PLC, do the following:

- Confirm that the IND700 can respond to a Ping on the Network. If it doesn't, then check the wiring and network connections.
- Use the Status LED's (described below) to diagnose and correct specific Network error conditions such as IP address conflicts.
- Confirm that the IND700 settings for data type, I/O size and IP address assignment match those in the PLC, and that each IND700 has a unique IP address.
- Use the provided Add On Profile (AOP) when possible, to simplify the setup in the PLC.
    - Check the Electronic Keying in the Add On Profile (AOP) to confirm that the firmware revision specified matches the firmware of the Ethernet/IP module installed in the IND700. If necessary, change the firmware version specified in the AOP, or change the Electronic Keying designation from "Exact Match" to "Compatible Module" or "Disable Keying".
- Confirm that the PLC's Ethernet/IP module firmware is up to date: the IND700's module contains the latest protocol updates, which means that it may not connect to PLCs using older firmware.
- If the PLC interface PCB was changed from another type, like DeviceNet or ControlNet, a master reset of the IND700 should be performed. Contact Metter Toledo service for assistance.
- Contact METTLER TOLEDO service for replacement of the EtherNet/IP interface.

2.9.1. Diagnostic LEDs

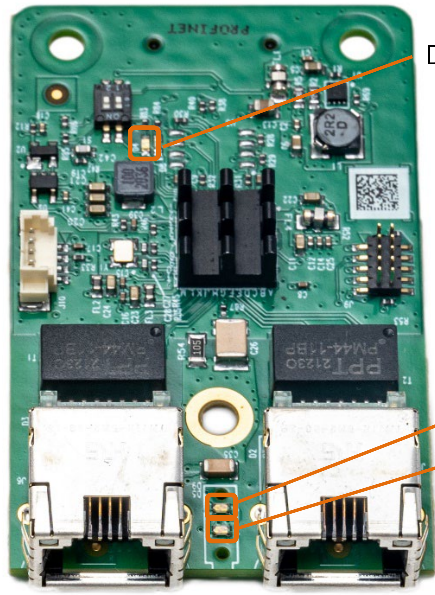


Figure 2-7: EtherNet/IP Board LED Locations

2.9.1.1. System

The System LED provides overall system information for the option board.

Table 2-4: EtherNet/IP System LED Indications

Color	State	Description
Gray	Off	• No supply voltage or hardware defect • Firmware reset in progress
Green	On Solid	Normal Operation
Green	Blinking	File system being formatted
Yellow	On Solid	A system error has occurred.
Yellow/Green	Alternating Colors, 3x Yellow then 3x Green	Unrecoverable firmware crash.
Yellow/Green	Alternating Colors, 1 Hz	Firmware update in progress.
Yellow/Green	Alternating Colors, 4 Hz	Firmware update in progress.

2.9.1.2. Module Status

Table 2-5: EtherNet/IP Module Status LED Indications

Color	State	Description
Gray	Off	No Power
Green/Red	Alternating Colors	Self Test
Green	Blinking	Standby

Color	State	Description
Green	On Solid	Operational
Red	Blinking	Major Recoverable Fault
Red	On Solid	Major Unrecoverable Fault

2.9.1.3. Network Status

Table 2-6: EtherNet/IP Network Status LED Indications

Color	State	Description
Gray	Off	No Power, No IP Address
Green	Blinking	No Connections
Green	On Solid	Connected
Green/Red	Alternating Colors	Self-test
Red	On Solid	Duplicate IP

2.10. EtherNet/IP Sample Code

- Check www.mt.com/IND700-downloads for the latest files and resources for the IND700 EtherNet/IP interface.

2.10.1. Introduction

- The configuration in this sample code is based on the default settings.

Rockwell Studio5000	Version 35
PLC	1756-L71
SAI Data Format	Block Format (default), 8-Block Format

2.10.2. Configure Development Environment

2.10.2.1. Confirm EDS Installation

To confirm installation of IND700 EDS file:

1. In any Studio 5000 project, right click on "Ethernet" within the I/O Configuration folder in the controller organizer.
2. Select New Module.

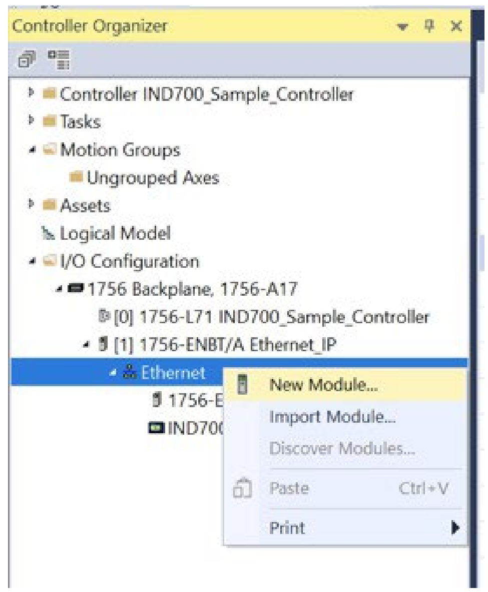


Figure 2-8: Adding a New Module to Confirm EDS is Installed

3. Search IND700

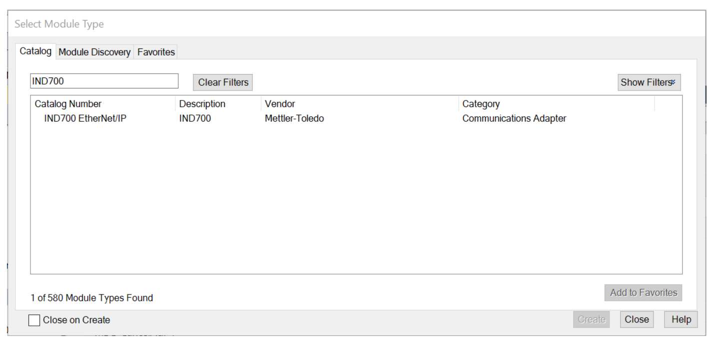


Figure 2-9: Search for IND700

4. If the EDS is installed, there should be an option for MT-IND700. If the search returns no results, follow these steps to install the EDS:
5. Use the EDS installation tool in Studio5000 to install the EDS.

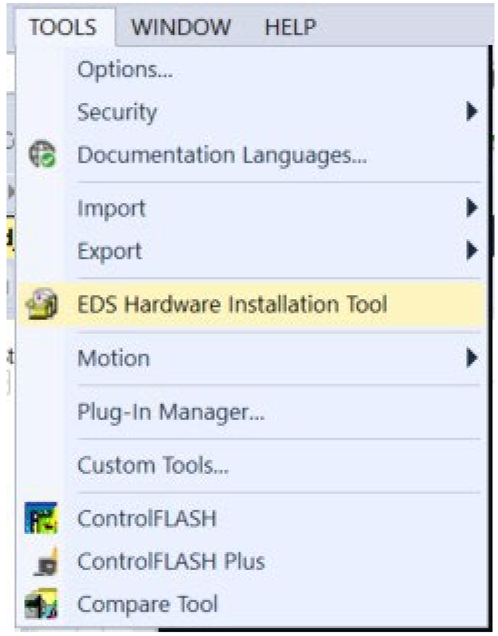


Figure 2-10: Use the EDS Hardware Installation Tool to complete installation

2.10.2.2. Import Example as a New Project

Studio5000 V24 to 35 is required to import the examples. Import the project in Studio5000 and click **File | Open.**

Select the .ACD file and click open. The project will load.

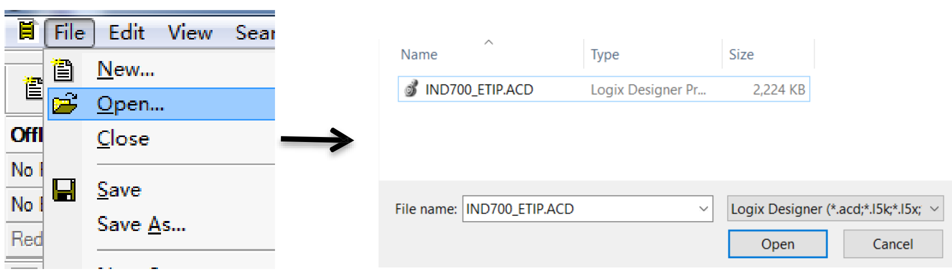


Figure 2-11: Import Project

2.10.2.3. Import Example to an Existing Project

1. Add an MT-IND700 to the I/O Configuration in the existing project. Refer to the first steps of Section 3 for more information on how to complete this. Using the name "IND700" and the IP Address 192.168.0.2 will require no changes to the sample code. If a different name or IP address is required, steps explaining what changes to make are provided below.

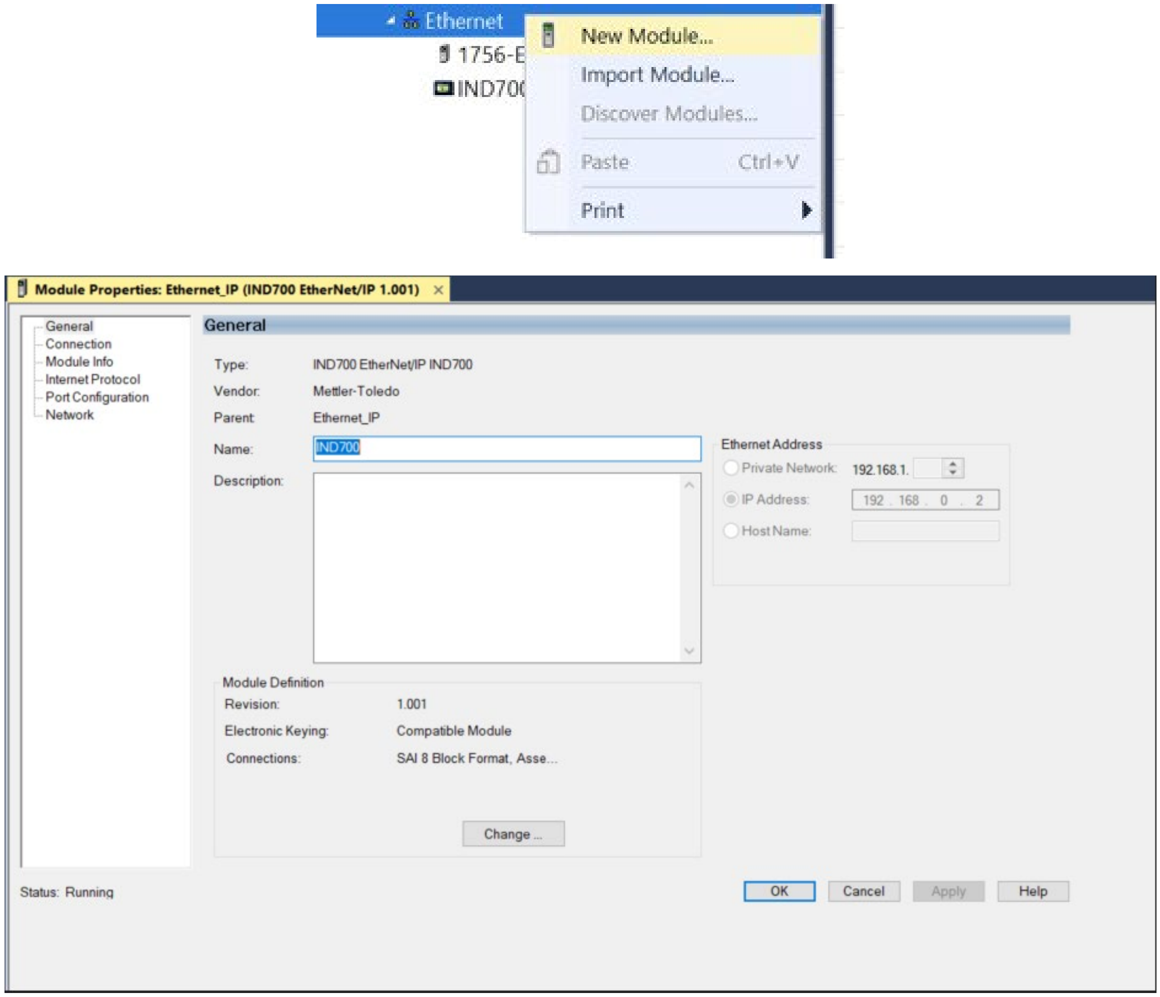


Figure 2-12: Module Properties – Add IND700 to the Existing Project

2. Copy the Add-On Instructions from the Add-On Instructions folder in the Controller Organizer of the sample project and paste in the same location in the existing project.

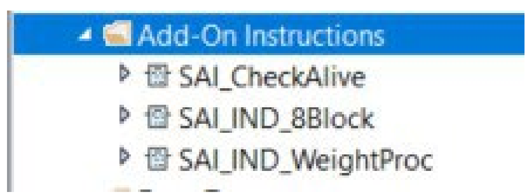


Figure 2-13: Copy/Paste AOIs

3. Copy the Main Program local tags from the sample project and paste in the tags for the existing project.

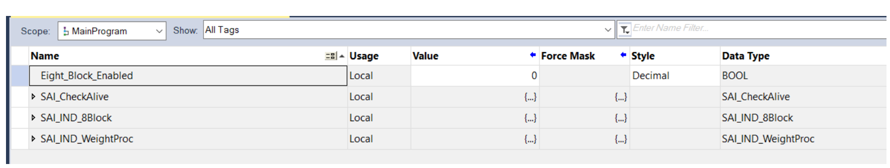


Figure 2-14: Copy/Paste Main Program Local Tags

4. Copy the "MT_IND_Application" routine from the sample project and paste in the existing project.

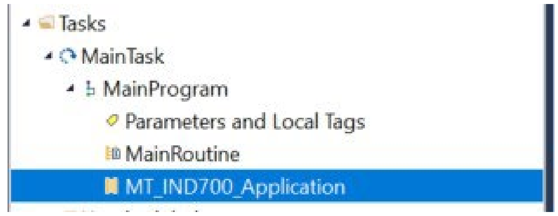


Figure 2-15: Copy/Paste the Routine

5. Make sure something in the existing project calls the MT_IND_Application. Any AOIs that automatically monitor weight conditions will not run if nothing calls this routine.
6. If a name other than "IND700" was used as the name of the transmitter in the project, replace every use of "IND700" in the AOI instances with the name given to the transmitter in the project.

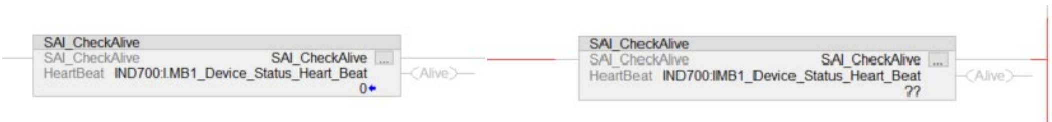


Figure 1-9: Example of Name "IND700:IMB1" Used in a Project

2.10.2.4. Configure Controller Type

Please note that this is only necessary if using the sample code as the basis for the PLC project. If importing the routine and AOIs into an already existing project, this is unnecessary.

Right-click the project's controller, select **Properties**, and set the controller type.

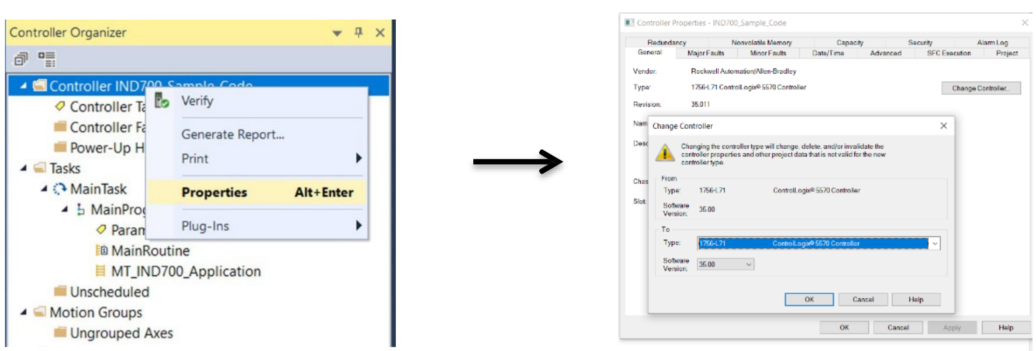


Figure 2-16: Controller Type Configuration

Download the project to the controller and test.

2.10.3. Add-On Instructions (AOI)

2.10.3.1. Cyclic Weight Data

Read the real-time and stable weight from the transmitter. When performing zero and tare commands, the weight will stop updating.

Trigger execution of stable tare, stable zero, immediate tare, immediate zero and clear tare by setting that particular bit high. The response can be read, and there are flags for execution success and failure to indicate the result.

After the zero and tare commands are completed, the AOI will automatically restore whatever command is in WeightCmd and weight will be reported again. Typical values for WeightCmd are 0 (report gross weight) or 3 (report net weight). The DataOK bit is reset to 0 during overload, underload, adjustment and several other scenarios which can be used to judge abnormal conditions.

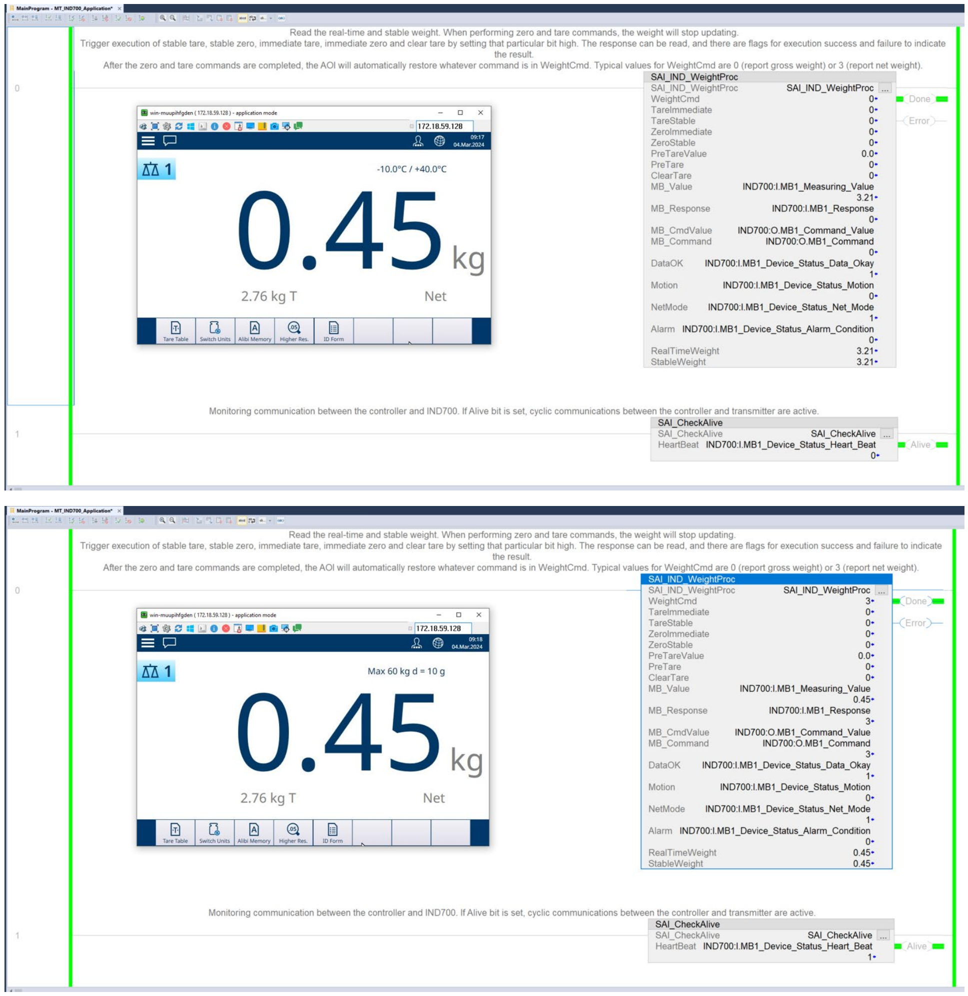


Figure 2-17: SAI_IND_WeightProc AOI

Table 2-7: Cyclic Data Description

Input Parameters	Data Type	Description
WeightCmd	INT	Use this value to request the IND700 to report weight. When a zero or tare cyclic command is sent, the IND700 stops reporting weight. This AOI will automatically restore this command once the zero or tare command completes. 0 or 1 = Report gross weight 2 = Report tare weight 3 = Report net weight 5 = Report gross weight value (with internal resolution) 6 = Report tare weight value (with internal resolution) 7 = Report net weight value (with internal resolution)
TareImmediate	BOOL	Set = 1 to issue tare command regardless of whether the weight value is stable or not. Net weight will not automatically be reported after tare command is issued. Recommended to use cyclic command 3 (report net weight) in WeightCmd input to receive net weight
TareStable	Bool	Set = 1 to issue tare command to IND700 when weight is stable. Command will timeout if remain within the stability criteria (+/- 1d within 0.3 seconds default) for a predefined timeout range (3 seconds default). Net weight will not automatically be reported after tare command is issued. Recommended to use cyclic command 3 (report net weight) in WeightCmd input to receive net weight
ZeroImmediate	BOOL	Set = 1 to issue zero command regardless of whether the weight value is stable or not. This is only intended for minor changes to the zero point due to drifting. For a formal zero adjustment, use the SAI_IND_ZeroAdjust AOI. Command will return an error if weight value is not within the zero range (+/- 2% default).
ZeroStable	BOOL	Set = 1 to issue zero command to IND700 when weight is stable. Command will timeout if remain within the stability criteria (+/- 1d within 0.3 seconds default) for a predefined timeout range (3 seconds default). This is only intended for minor changes to the zero point due to drifting. For a formal zero adjustment, use the SAI_IND_ZeroAdjust AOI. Command will return an error if weight value is not within the zero range (+/- 2% default).
PreTareValue	Real	Configure with preset tare value. This value will not be sent to IND700 until Pretare input is set to 1.
Pretare	BOOL	Set = 1 when ready to perform preset tare using the value from PreTareValue.
ClearTare	BOOL	Set = 1 to clear the current tare value.
MB_Value	Real	This should always be set to the MB1_Measuring_Value of the IND700. This will provide weight data for the AOI
MB_Response	INT	This should always be set to MB1_Response value of the IND700. Once a cyclic command is successfully executed, MB_Response = MB_Command. The AOI uses this information to detect if a command has been executed successfully or if an error has occurred.

Input Parameters	Data Type	Description
DataOK	BOOL	This bit gets set to 0 when the device is still operational but the value being reported cannot be guaranteed to be valid. The following conditions cause the Data Okay bit to be set to 0: • Device is powering up • Device is in setup mode • Device is in test mode • Over capacity condition occurs - When the A/D converter is at its limit - Product-dependent over capacity that occurs when the device determines it cannot trust the weight • Under capacity condition occurs - When the A/D converter is at its limit Product dependent under capacity that occurs when the device determines it cannot trust the weight
Motion	BOOL	This should always be set to Motion bit of IND700. Motion bit is high when the weight value is not stable. ZeroStable and TareStable commands will not complete while the Motion bit is high.
NetMode	BOOL	This should always be set to the Net_Mode bit of the IND700. NetMode = 1 after a tare command has been executed. Just because NetMode = 1, does not mean net weight is being reported by the IND700. Net weight must be requested by the PLC (cyclic command 3).
Alarm	BOOL	This should always be set to Alarm_Condition bit of IND700. Bit will go high when alarm conditions are present. Bit will automatically go low when no alarm conditions are present. Refer to the **SAI Reference for Transmitters and Terminals** for more information on what causes the Alarm_Condition to go high.
MB_Command	INT	This should always be set to MB1_Command value of the IND700. Value of the last cyclic command sent to the IND700. Once successfully executed, MB_Response = MB_Command.
RealTimeWeight	Real (32 bits)	Current weight on the scale. This value is updated constantly while the AOI is enabled.
StableWeight	Real (32 bits)	Latest stable weight reading from the scale. This value does not update whenever the Motion bit is high.
Done	BOOL	Will be latched high when zero or tare command has been executed successfully. When a new zero or tare command begins, this bit will be unlatched until the command completes successfully.

2.10.4. Communication Heart Beat Monitoring

Monitoring communication between the controller and IND700. If Alive bit is set, cyclic communications between the controller and transmitter are active.

Figure 2-18: SAI_CheckAlive AOI

Table 2-8: AOI Status Checking Parameters

Input Parameters	Data Type	Description
HeartBeat	BOOL	This should always be set to Heart_Beat bit of IND700. This bit will pulse on and off each second if cyclic communications between the IND700 and the controller are established
Alive	BOOL	This bit = 1 if cyclic communications are established between the IND700 and the controller.

2.10.5. Steps to Add New IND700s

Because EtherNet/IP uses IP addresses to distinguish different devices, when multiple IND700s are networked, the default IP address needs to be modified first. Each IND700 must have a different IP address.

1. In the IND700, access **Setup-> Communication-> Industrial Network -> EtherNet/IP**.

Figure 2-19: IND700 IP Address Menu

2. Add an MT-IND700 to **I / O Configuration | Ethernet** in Studio5000.

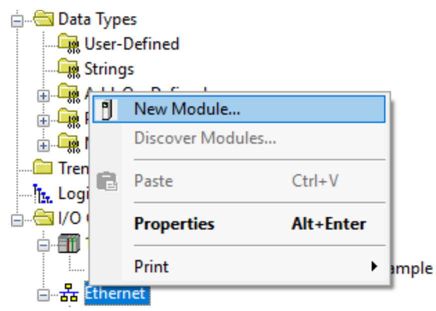


Figure 2-20: Add a Device

3. Configure the name and IP address. Each device needs a unique name and IP address, and then click **Change**.

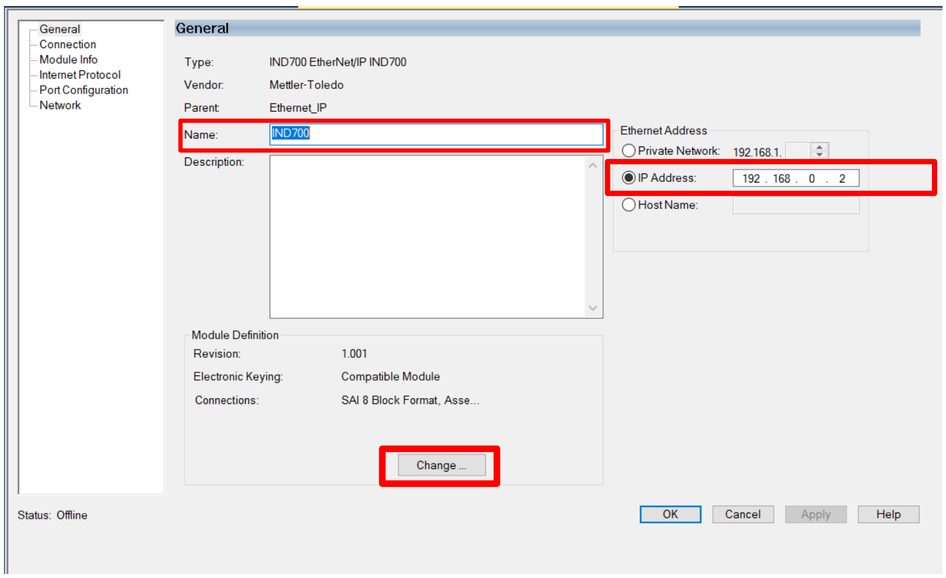


Figure 2-21: Configure Name and IP Address

4. Select **I/O 2 Block Format** to have the sample code function with minimal changes. Select 8 Block if required to receive multiple pieces of cyclic data simultaneously. For example, if it is required to read the gross weight, net weight and target weight at one time, 8 Block can easily accomplish this.

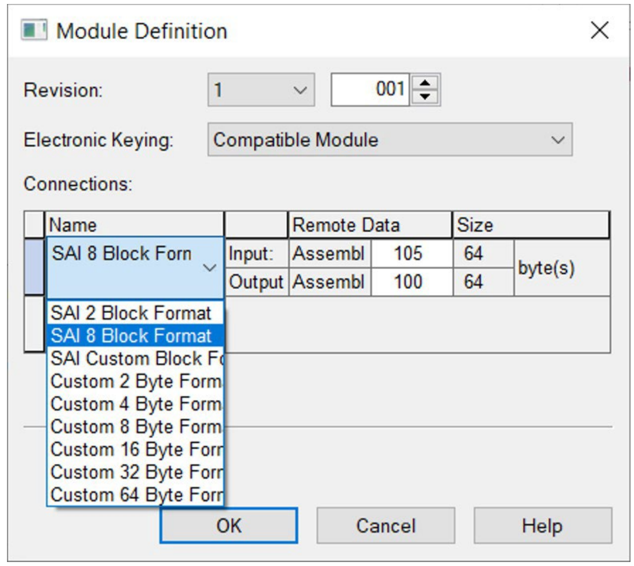


Figure 2-22: Module Definition Configuration

5. Copy and paste the Add-On Instructions and configure the instance name along with the input and output parameters. Refer to Section 2 for additional instructions for configuring the Message parameters. Each device must correspond to a unique instance of the AOI. As shown in the figure below, both devices call the AOI SAI_CheckAlive, but the corresponding instances are SAI_CheckAlive and SAI_CheckAlive_1. Notice that the Heartbeat parameter is also configured with different devices for these two instances. Refer to section 2.10.3 for information on configuring parameters for a particular AOI. Make sure that all tags for the second device for instance now end in "_2" as opposed to "_1" for the first device.

Figure 2-23: Two instances of the SAI_CheckAlive AOI for two IND700s

6. Repeat steps 1 to 5 until the configuration of all devices is completed.

2.10.6. Steps to Use 8 Block Format Instead of 2 Block Format

The 8 Block Format for SAI is incredibly powerful for viewing more information simultaneously compared to the 2 Block Format. The sample code by default is configured for the 2 Block Format, but changing the format is very simple:

1. Right click on the **IND700** in the Controller Organizer
2. Click **Properties**

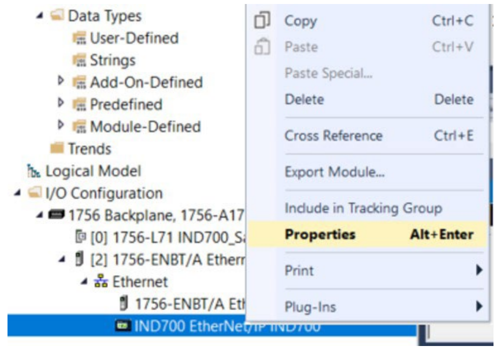


Figure 2-24: Select Properties

3. Click **Change** under the Module Definition

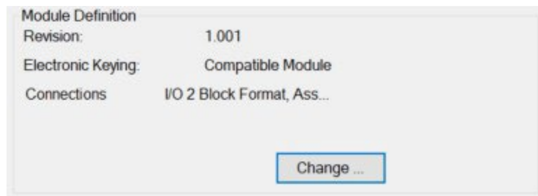


Figure 2-25: Click "Change"

4. Select the drop-down arrow next to **I/O 2 Block Format** and select **I/O 8 Block Format**.

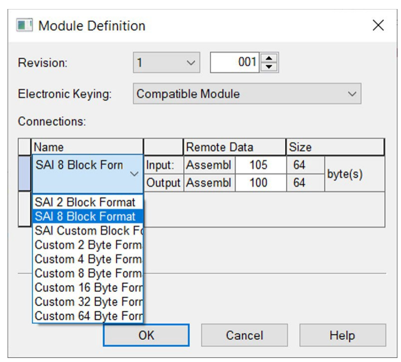


Figure 2-26: Select "I/O 8 Block Format"

At this point, the project has the IND700 configured for the 8 block format. No changes to the AOIs are required since the 8 Block Format just expands upon the 2 Block format used by the AOIs. However, there is now an additional AOI in the sample project that can be used. The final step is to confirm the IND700 itself is configured for the 8 block format. This setting can be found in the IND700 setup menu or via the web interface at **Communication -> Industrial Network ->Format**.

2.10.6.1. Add-On Instruction for 8 Block Format

The AOI for the 8 block format is available in the controller organizer under **Assets | Add-On Instructions | SAI_IND_8Bloc**k. This AOI can be dragged from the controller organizer into the "MT_IND_700" routine. Information for configuring the AOI can be found below.

In the 8 block format, there is one measuring block and seven status blocks. The first measuring block and status block are also used in the 2 block format. What is gained by changing to the 8 block format is measuring blocks 2-7.

This AOI allows the user to enter a cyclic command (MBx_Command_In) and an optional parameter associated with the command (MBx_Value_In), to see the result from the command (MBx_Measuring Value), to determine whether the command executed (Done_MBx), and to see if there was an error executing the command (Error_MBx).

Figure 2-27: SAI_IND_8Block

Table 2-9: 8-Block Format Parameters

Input Parameters	Data Type	Description
MB2_Measuring_Value	Real	This should always be set to the MB2_Measuring_Value of the IND700. Measured value returned from the last successfully executed command for MB2. Ignore value if neither Done_MB2 nor Error_MB2 is set since that means a command is in the process of executing.
MB3_Measuring_Value	Real	This should always be set to the MB3_Measuring_Value of the IND700. Measured value returned from the last successfully executed command for MB3. Ignore value if neither Done_MB3 nor Error_MB3 is set since that means a command is in the process of executing.
MB4_Measuring_Value	Real	This should always be set to the MB4_Measuring_Value of the IND700. Measured value returned from the last successfully executed command for MB4. Ignore value if neither Done_MB4 nor Error_MB4 is set since that means a command is in the process of executing.
MB5_Measuring_Value	Real	This should always be set to the MB5_Measuring_Value of the IND700. Measured value returned from the last successfully executed command for MB5. Ignore value if neither Done_MB5 nor Error_MB5 is set since that means a command is in the process of executing.

Input Parameters	Data Type	Description
MB6_Measuring_Value	Real	This should always be set to the MB6_Measuring_Value of the IND700. Measured value returned from the last successfully executed command for MB6. Ignore value if neither Done_MB6 nor Error_MB6 is set since that means a command is in the process of executing.
MB7_Measuring_Value	Real	This should always be set to the MB7_Measuring_Value of the IND700. Measured value returned from the last successfully executed command for MB7. Ignore value if neither Done_MB7 nor Error_MB7 is set since that means a command is in the process of executing.
MB2_Command_In	INT	Place the cyclic command to be executed using MB2 here. Refer to the **SAI Manual for Terminals and Transmitters** to see all valid cyclic commands. Data reported will be found in MB2_Measuring Value. Optional parameter information should be placed in MB2_Value_In before changing the command. Some examples of common cyclic commands: 0 or 1 = Report gross weight 2 = Report tare weight 3 = Report net weight 5 = Report gross weight value (with internal resolution) 6 = Report tare weight value (with internal resolution) 7 = Report net weight value (with internal resolution)
MB3_Command_In	INT	Place the cyclic command to be executed using MB3 here. Refer to the **SAI Manual for Terminals and Transmitters** to see all valid cyclic commands. Data reported will be found in MB3_Measuring Value. Optional parameter information should be placed in MB3_Value_In before changing the command. Some examples of common cyclic commands: 0 or 1 = Report gross weight 2 = Report tare weight 3 = Report net weight 5 = Report gross weight value (with internal resolution) 6 = Report tare weight value (with internal resolution) 7 = Report net weight value (with internal resolution)
MB4_Command_In	INT	Place the cyclic command to be executed using MB4 here. Refer to the **SAI Manual for Terminals and Transmitters** to see all valid cyclic commands. Data reported will be found in MB4_Measuring Value. Optional parameter information should be placed in MB4_Value_In before changing the command. Some examples of common cyclic commands: 0 or 1 = Report gross weight 2 = Report tare weight 3 = Report net weight 5 = Report gross weight value (with internal resolution)

Input Parameters	Data Type	Description
		6 = Report tare weight value (with internal resolution) 7 = Report net weight value (with internal resolution)
MB5_Command_In	INT	Place the cyclic command to be executed using MB5 here. Refer to the **SAI Manual for Terminals and Transmitters** to see all valid cyclic commands. Data reported will be found in MB5_Measuring Value. Optional parameter information should be placed in MB5_Value_In before changing the command. Some examples of common cyclic commands: 0 or 1 = Report gross weight 2 = Report tare weight 3 = Report net weight 5 = Report gross weight value (with internal resolution) 6 = Report tare weight value (with internal resolution) 7 = Report net weight value (with internal resolution)
MB6_Command_In	INT	Place the cyclic command to be executed using MB6 here. Refer to the **SAI Manual for Terminals and Transmitters** to see all valid cyclic commands. Data reported will be found in MB6_Measuring Value. Optional parameter information should be placed in MB6_Value_In before changing the command. Some examples of common cyclic commands: 0 or 1 = Report gross weight 2 = Report tare weight 3 = Report net weight 5 = Report gross weight value (with internal resolution) 6 = Report tare weight value (with internal resolution) 7 = Report net weight value (with internal resolution)

Input Parameters	Data Type	Description
MB7_Command_In	INT	Place the cyclic command to be executed using MB7 here. Refer to the **SAI Manual for Terminals and Transmitters** to see all valid cyclic commands. Data reported will be found in MB7_Measuring Value. Optional parameter information should be placed in MB7_Value_In before changing the command. Some examples of common cyclic commands: 0 or 1 = Report gross weight 2 = Report tare weight 3 = Report net weight 5 = Report gross weight value (with internal resolution) 6 = Report tare weight value (with internal resolution) 7 = Report net weight value (with internal resolution)
MB2_Value_In	Real	Place the optional parameter associated with MB2_Command_In here. For example, if executing a pretare command, MB2_Value_In would be set to the pretare value before changing MB2_Command_In
MB3_Value_In	Real	Place the optional parameter associated with MB3_Command_In here. For example, if executing a pretare command, MB3_Value_In would be set to the pretare value before changing MB3_Command_In
MB4_Value_In	Real	Place the optional parameter associated with MB4_Command_In here. For example, if executing a pretare command, MB4_Value_In would be set to the pretare value before changing MB4_Command_In
MB5_Value_In	Real	Place the optional parameter associated with MB5_Command_In here. For example, if executing a pretare command, MB5_Value_In would be set to the pretare value before changing MB2_Command_In
MB6_Value_In	Real	Place the optional parameter associated with MB6_Command_In here. For example, if executing a pretare command, MB6_Value_In would be set to the pretare value before changing MB6_Command_In
MB7_Value_In	Real	Place the optional parameter associated with MB7_Command_In here. For example, if executing a pretare command, MB7_Value_In would be set to the pretare value before changing MB7_Command_In
MB2_Response	INT	This should always be set to MB2_Response value of the IND700. Once a cyclic command is successfully executed, MB2_Response = MB2_Command_In. The AOI uses this information to detect if a command has been executed successfully or if an error has occurred.
MB3_Response	INT	This should always be set to MB3_Response value of the IND700. Once a cyclic command is successfully executed, MB3_Response = MB3_Command_In. The AOI uses this information to detect if a

Input Parameters	Data Type	Description
		command has been executed successfully or if an error has occurred.
MB4_Response	INT	This should always be set to MB4_Response value of the IND700. Once a cyclic command is successfully executed, MB4_Response = MB4_Command_In. The AOI uses this information to detect if a command has been executed successfully or if an error has occurred.
MB5_Response	INT	This should always be set to MB5_Response value of the IND700. Once a cyclic command is successfully executed, MB5_Response = MB5_Command_In. The AOI uses this information to detect if a command has been executed successfully or if an error has occurred.
MB6_Response	INT	This should always be set to MB6_Response value of the IND700. Once a cyclic command is successfully executed, MB6_Response = MB6_Command_In. The AOI uses this information to detect if a command has been executed successfully or if an error has occurred.
MB7_Response	INT	This should always be set to MB7_Response value of the IND700. Once a cyclic command is successfully executed, MB7_Response = MB7_Command_In. The AOI uses this information to detect if a command has been executed successfully or if an error has occurred.
MB2_Command_Out	INT	This should always be set to MB2_Command of the IND700. Value of the last cyclic command sent to the IND700. Once successfully executed, MB2_Response = MB2_Command_Out = MB2_Command_In.
MB3_Command_Out	INT	This should always be set to MB3_Command of the IND700. Value of the last cyclic command sent to the IND700. Once successfully executed, MB3_Response = MB3_Command_Out = MB3_Command_In.
MB4_Command_Out	INT	This should always be set to MB4_Command of the IND700. Value of the last cyclic command sent to the IND700. Once successfully executed, MB4_Response = MB4_Command_Out = MB4_Command_In.
MB5_Command_Out	INT	This should always be set to MB5_Command of the IND700. Value of the last cyclic command sent to the IND700. Once successfully executed, MB5_Response = MB5_Command_Out = MB5_Command_In.
MB6_Command_Out	INT	This should always be set to MB6_Command of the IND700. Value of the last cyclic command sent to the IND700. Once successfully executed, MB6_Response = MB6_Command_Out = MB6_Command_In.

Input Parameters	Data Type	Description
MB7_Command_Out	INT	This should always be set to MB7_Command of the IND700. Value of the last cyclic command sent to the IND700. Once successfully executed, MB7_Response = MB7_Command_Out = MB7_Command_In.
MB2_CmdValue_Out	Real	This should always be set to MB2_Command_Value of IND700
MB3_CmdValue_Out	Real	This should always be set to MB3_Command_Value of IND700
MB4_CmdValue_Out	Real	This should always be set to MB4_Command_Value of IND700
MB5_CmdValue_Out	Real	This should always be set to MB5_Command_Value of IND700
MB6_CmdValue_Out	Real	This should always be set to MB6_Command_Value of IND700
MB7_CmdValue_Out	Real	This should always be set to MB7_Command_Value of IND700
Done_MB2	BOOL	Will be latched high when command has successfully completed for MB2. When a new command begins, bit will be unlatched until command completes successfully
Done_MB3	BOOL	Will be latched high when command has successfully completed for MB3. When a new command begins, bit will be unlatched until command completes successfully
Done_MB4	BOOL	Will be latched high when command has successfully completed for MB4. When a new command begins, bit will be unlatched until command completes successfully
Done_MB5	BOOL	Will be latched high when command has successfully completed for MB5. When a new command begins, bit will be unlatched until command completes successfully
Done_MB6	BOOL	Will be latched high when command has successfully completed for MB6. When a new command begins, bit will be unlatched until command completes successfully
Done_MB7	BOOL	Will be latched high when command has successfully completed for MB7. When a new command begins, bit will be unlatched until command completes successfully
Error_MB2	BOOL	Will be latched high when command fails to complete for MB2. When a new command begins, bit will be unlatched until a command fails to complete.
Error_MB3	BOOL	Will be latched high when command fails to complete for MB3. When a new command begins, bit will be unlatched until a command fails to complete.
Error_MB4	BOOL	Will be latched high when command fails to complete for MB4. When a new command begins, bit will be unlatched until a command fails to complete.
Error_MB5	BOOL	Will be latched high when command fails to complete for MB5. When a new command begins, bit will be unlatched until a command fails to complete.

Input Parameters	Data Type	Description
Error_MB6	BOOL	Will be latched high when command fails to complete for MB6. When a new command begins, bit will be unlatched until a command fails to complete.
Error_MB7	BOOL	Will be latched high when command fails to complete for MB7. When a new command begins, bit will be unlatched until a command fails to complete.

2.10.7. Frequently Asked Questions

5. **Q**: How do I access the parameters in the AOI variables within my PLC program?

   A: Use the format **instance_name.parameter** to access parameters in the PLC program. For example, creating an instance of the SAI_CheckAlive AOI and naming the instance **IND700_Comm**, allows the alive bit to be monitored by looking at **IND700_Comm.Alive**.

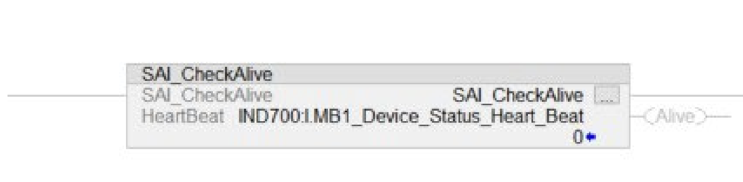


Figure 2-28: SAI_CheckAlive AOI With Different Instance Mame

6. **Q**: Does my AOI instance always have to match the name of the AOI?

   A: No, the AOI instance can be named anything, but the name must be unique. This ensures that, if we are using multiples of the same AOI, we can distinguish between them in the code. Refer to Figure 2-28 for an example of an AOI instance name that does not match the AOI name but is still valid.

7. **Q**: How do I read gross, tare or net weight?

   A: Use the WeightCmd parameter of the SAI_IND_WeightProc AOI to issue different weight commands. Valid parameter values are:

   0 = report gross weight

   2 = report tare weight

   3 = report net weight.

   The weight will be updated in StableWeight if the weight value is stable.

   StableWeight will freeze with the last reported stable weight while the scale is in motion. RealTimeWeight will constantly update the weight value regardless of whether the scale is in motion.

8. **Q**: How do I know the source of the error in the SAI_IND_WeightProc AOI?

   A: Typical errors in this AOI include:

   - **Pushbutton zero failure**: Tried to zero when the weight value is outside of the pushbutton zero range (+/- 2% by default). If a substantial zero adjustment needs to be made, use the SAI_IND_ZeroAdjust AOI instead.

- **Tare failure**: Typically seen if trying to tare a negative gross weight value. Try to tare again with a positive gross weight.
- **Stability failure**: Can occur with either the ZeroStable or Tare Stable command. The weight value must meet the stability criteria (no more than 1 division of change occurring in a 0.3 second period by default) at some point before a timeout occurs (3 seconds by default).

9. **Q**: An AOI is very close to what I want to do in my PLC logic, but I need to make a few changes. How can I do that?

   A: If necessary to view or modify the logic of an AOI, simply use the Controller Organizer view in Studio 5000 to navigate to Add-On Instructions, expand the AOI to view, and double-click **Logic**. This will display the ladder logic used in the AOI and can be changed as necessary for the application.

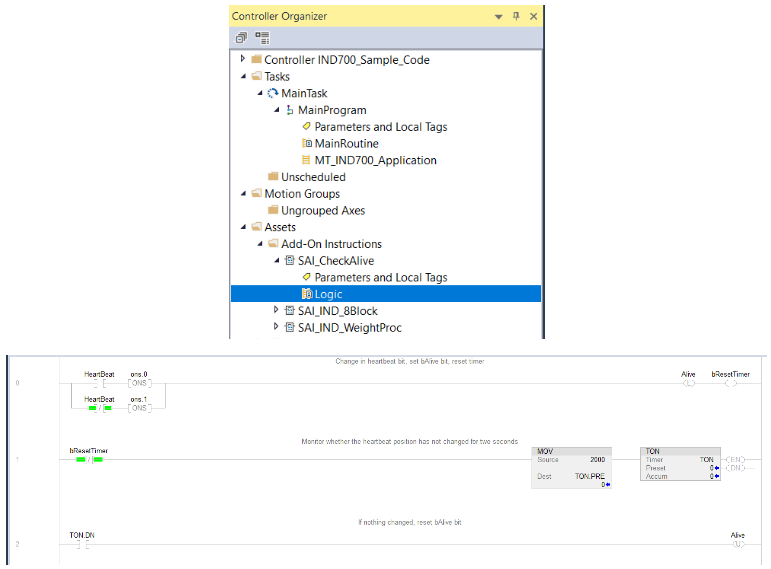


Figure 2-29: Example of AOI Ladder Logic

3 PROFINET

3.1. Overview

PROFINET is an open industrial networking standard that was developed by Siemens as an Ethernet replacement for its widely popular PROFIBUS Network. The network supports Cyclic and Acyclic messaging, both of which have been implemented in the IND700. The PROFINET Standard is supported and Maintained by the PROFIBUS and PROFINET International (PI) Organization. PROFINET utilizes commercial, off-the-shelf Ethernet hardware (for example, switches and routers) and is fully compatible with the Ethernet TCP/IP protocol suite.

The IND700 PROFINET option implements PROFINET IO for cyclic data exchange with the PLC, and uses acyclic messages for Shared Data Access by the PLC.

The PROFINET option enables the IND700 terminal to communicate to PROFINET enabled Programmable Logic Controllers (PLCs) through direct connection to the PROFINET network at 100 MBPS speed. This solution consists of an internal module and internal software to implement the data exchange.

3.2. PROFINET Interface

The part # of the IND700 PROFINET option kit is 30725998. Figure 3-1 shows a PROFINET interface board together with the kit contents. The board part number is 30785338.

Figure 3-1: PROFINET Interface Kit

3.2.1. Update Rates

The update rate for the PROFINET interface varies depending on the terminal's configuration:

- HSALC 1 scale: 66 Hz
- HSALC 2 scales: 50 Hz
- POWERCELL 4 scales 15 Hz

3.2.2. Definition of Terms

The following terms are used in this chapter.

Table 3-1: PROFINET Definition of Terms

Term	Definition
DAP	Device Access Point
DCP	Discovery and basic Configuration Protocol. Used for IP configuration over PROFINET.
DHCP	De-facto standard for dynamic IP address management
GSDML	XML-based descriptive language for GSD-files
Initial Record Data	Record Data write-requests destined for a sub-module. Comparable to PROFIBUS-DP User Parameter Data.
IOCS	IO Consumer Status
IOPS	IO Provider Status
IO Controller	Controlling device which acts as a client for several IO devices. Usually a PLC. Comparable to a PROFIBUS-DP Class 1 master.
IO Device	Field device assigned to an IO Controller. Comparable to a PROFIBUS DPV1 slave.
IO Supervisor	Programming device with commissioning and diagnostic functions. Comparable to a PROFIBUS-DP Class 2 master.
Module	Hardware or logical component of a network device.
MRP	Media Redundancy Protocol. An Ethernet Ring Topology used with PROFINET IO to provide media redundant communications. Messages are sent out one Ethernet port of the PLC and come back in the other. If the PLC detects a media break in the ring then it reconfigures the network within 200 milliseconds so that messages will be sent out both ports of the PLC. Requires PLC's and devices that are MRP enabled. Any switches on the network must also be MRP enabled. Non-MRP enabled devices may be connected to the loop by using MRP enabled switches.
Submodule	Hardware or logical component of a module
PDEV	Physical DEVice. From specification version 2.0 it is possible to describe the physical Ethernet interface and its ports (PDEV, or Physical Device) with a special mechanism. This is done with special sub-modules at slot 0 (the module at slot 0 is the access point for the device).
PNIO	Short for PROFINET IO.

Term	Definition
PROFINET IO	PROFINET IO is a communication concept for the implementation of modular, decentralized applications. Comparable to PROFIBUS-DP, where I/O data of field devices are cyclically transmitted to the process image of a PLC. The real time capabilities of PROFINET IO are further divided into RT and IRT (see below).
PROFINET IO RT	PROFINET IO with Real Time capabilities. Optimized real time communication channel for time critical I/O data and Alarms. Implemented in software.
PROFINET IRT	PROFINET IO with Isochronous Real Time capabilities. Necessary for motion control application which require an update rate of 1ms, or less, with no jitter. Implemented in hardware.
PROFINET CBA	PROFINET Component Based Automation. Comparable to PROFIBUS FMS.
Record Data	Comparable to PROFIBUS DPV1 acyclic Read/Write.

3.2.3. Communications

The IND700 terminal uses component parts to ensure complete compatibility with the Siemens PROFINET network. An IND700 terminal is recognized as a generic PROFINET device by the PLC.

3.2.4. IP Address

Each PROFINET option represents one physical IP Address. This address can be chosen by the system designer, and then programmed into the IND700 terminal and PLC, or the address can be automatically assigned by the PLC. Each IND700 within a system must have a unique PROFINET IP Address.

The IND700 terminal's PROFINET IP address is programmed in the terminal's setup menu at Communication > Industrial Network > PROFINET.

3.2.5. Supported Data Transfer

The PROFINET interface provides both discrete data transfer and an acyclic messaging capability that is used for Shared Data access. Access to Shared Data is done in a manner that is very similar to the method used by the ControlNet and Ethernet/IP modules.

3.2.6. Connection Methods

The dual ports on the PROFINET Interface module provide several possible methods for connecting the IND700 to the control Network. Those methods are described in this section. It is important that in both the Daisy Chain and MRP Redundant Loop configurations, the physical network wiring matches the network topology defined on the PLC as it relates to Port 1 and Port 2. If the wiring does not match the defined topology, errors will be reported.

3.2.6.1. Star Network

A star network consists of multiple devices being attached to one or more Ethernet switches.

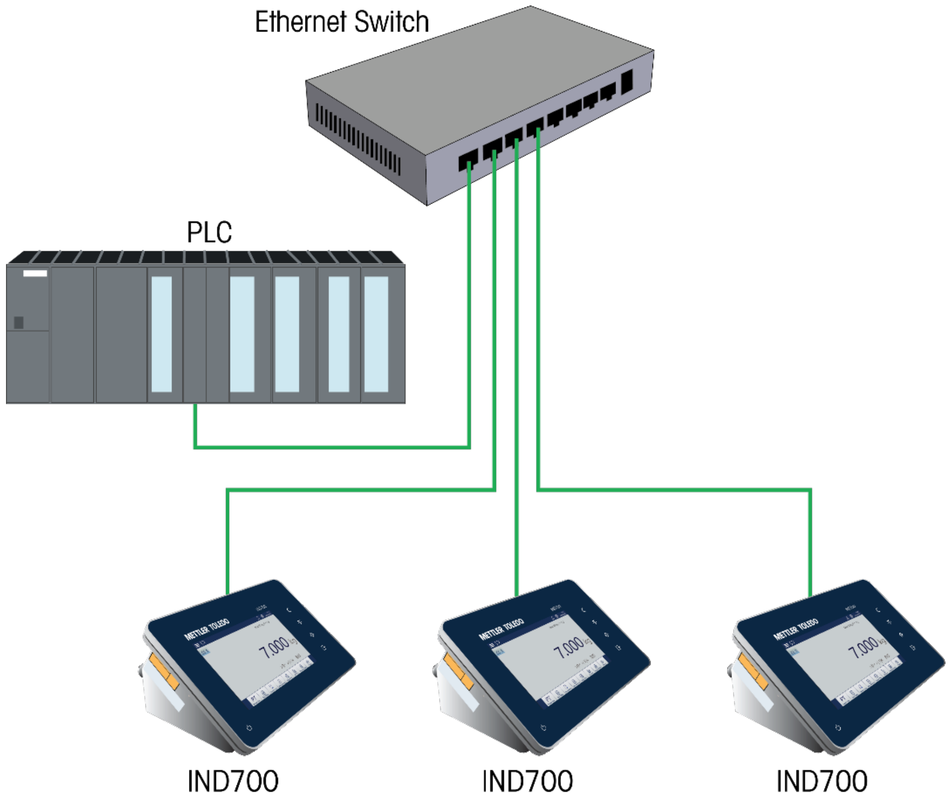


Figure 3-2: Star Network Example

3.2.6.2. Daisy Chain

A Daisy Chain network has the advantage of not requiring switches for multiple devices to be connected to the Controller. This has advantages in a cabinet or tight space where there may not be sufficient area to run individual cables all of the way back to a central point such as a switch.

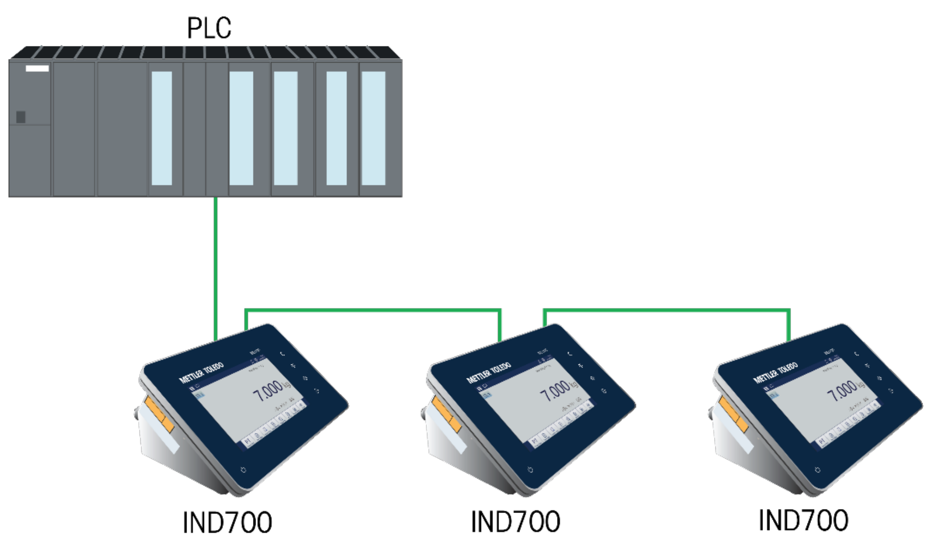


Figure 3-3: Daisy Chain Example

3.2.6.3. MRP Redundant Loop

An MRP redundant loop is very similar to the Daisy Chain topology, where the PLC is connected on one end of the loop, and devices are daisy chained along the loop until the loop is terminated back at the same PLC on a second Ethernet port. This provides a 'Ring' topology where messages can be routed either direction around the ring, and has the advantage of not requiring any switches as long as the PLC and the devices are MRP capable. If a break in the Ring occurs, the PLC will quickly detect it by noticing that messages are no longer making it back to the PLC on the opposite end of the ring that is attached to it. Under those conditions the PLC will then start transmitting the messages out both ports so that all devices on the ring can still get the messages. The result is a network of daisy chains out each port that continue to function regardless of the break. PROFINET MRP is designed to make the break detection and switch over in less than 200 milliseconds. NOTE that your process must be able to tolerate a loss of communications for up to 200 ms.

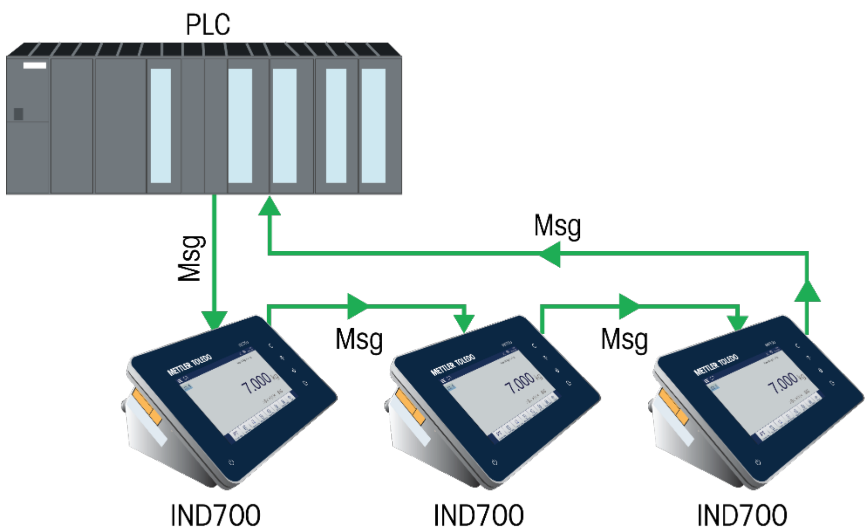


Figure 3-4: Intact MRP Ring

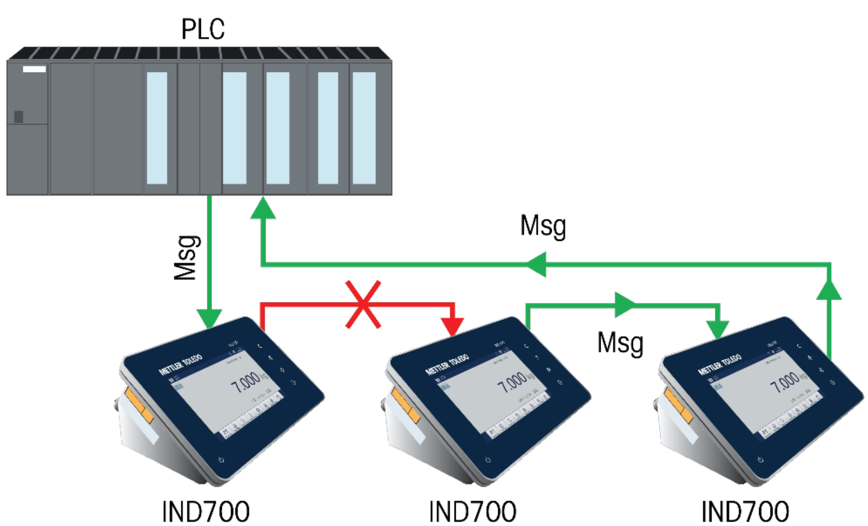


Figure 3-5: MRP Ring with Break

- Note that messages still reach all devices, because the network is self-healing.

3.3. Data Definition

3.3.1. Data Integrity

The IND700 has specific bits to allow the PLC to confirm that data was received without interruption and that the IND700 is not in an error condition. It is important to monitor these bits. Any PLC code should use them to confirm the integrity of the data received by the IND700.

Refer to the data charts in Appendix A and Appendix B for specific information regarding the Data OK, Update in Progress and Data Integrity bits and their usage.

3.3.2. Discrete Data

The terminal's PROFINET interface has three discrete data formats that may be selected. The data types are: Integer, Divisions and Floating Point.

Please refer to Appendix C, **Common Data Features** for a description of discrete data, and to Appendix A and Appendix B for a detailed description of the information available in each data format.

3.3.3. Byte Order

For a general account of byte ordering, please refer to Appendix C, **Common Data Features**.

3.3.4. Message Slots

There may be up to 4 message slots for discrete data transfer of cyclic messages in Integer, Divisions and Floating Point Data Formats. Each message slot is assigned to a local or remote scale and scales may be repeated in additional message slots. The integer and division formats provide two 16-bit words of input and two 16-bit words of output data per Message Slot. Each Message Slot's first input word provides scale weight data and the input weight data may be selected by the PLC using the Message Slot's second output word bit 0, bit 1 and bit 2. The following two Tables provide input and output usage information.

The floating point format provides four 16-bit words of input data and three 16-bit words of output data) per Message Slot. Refer to Table 3-2 and Table 3-3 for details.

The number of Message Slots is selected in the terminal's setup menu at **Communication > PLC Interface > Custom** or **SA**, depending which mode is selected in the **Mode** menu..

Table 3-2: Message Slot and PLC I/O Sizes (Integer/ Division)

IND700 Integer/ Division Data		
Message Slots	Bytes (8 Bit)	
	IND700 >> PLC Input	PLC Output >> IND700
1	4	4
2	8	8
3	12	12
4	16	16

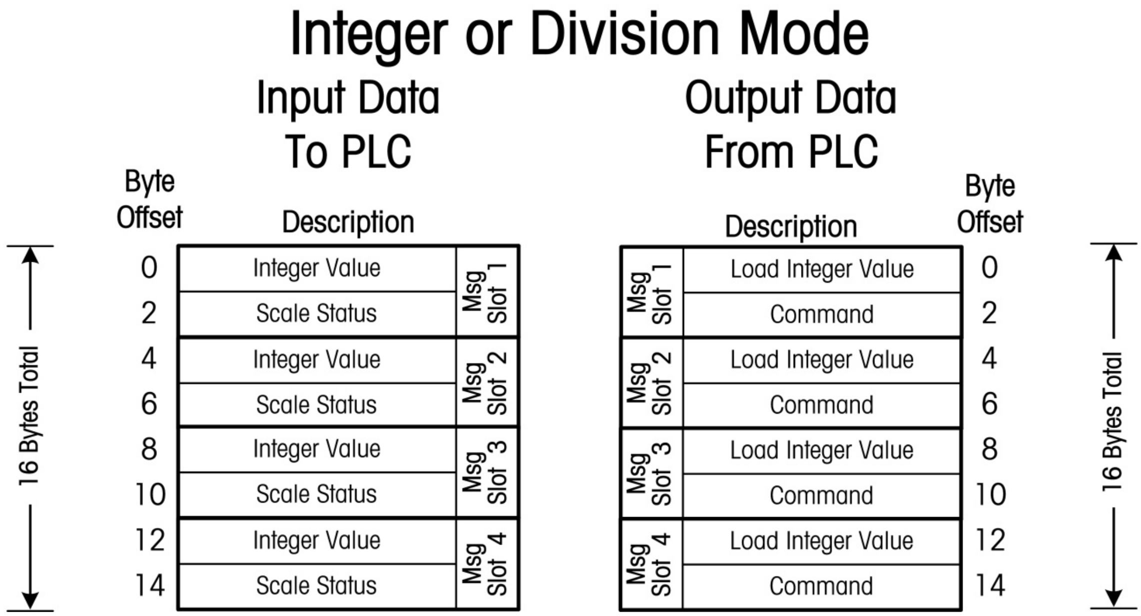


Figure 3-6 Integer/Divisions Message Slot I/O Mapping

Table 3-3: Message Slot and PLC I/O Sizes (Floating Point)

IND700 Floating Point Data		
Message Slots	Bytes (8 Bit)	
	IND700 >> PLC Input	PLC Output >> IND700
1	8	8
2	16	14
3	24	20
4	32	26

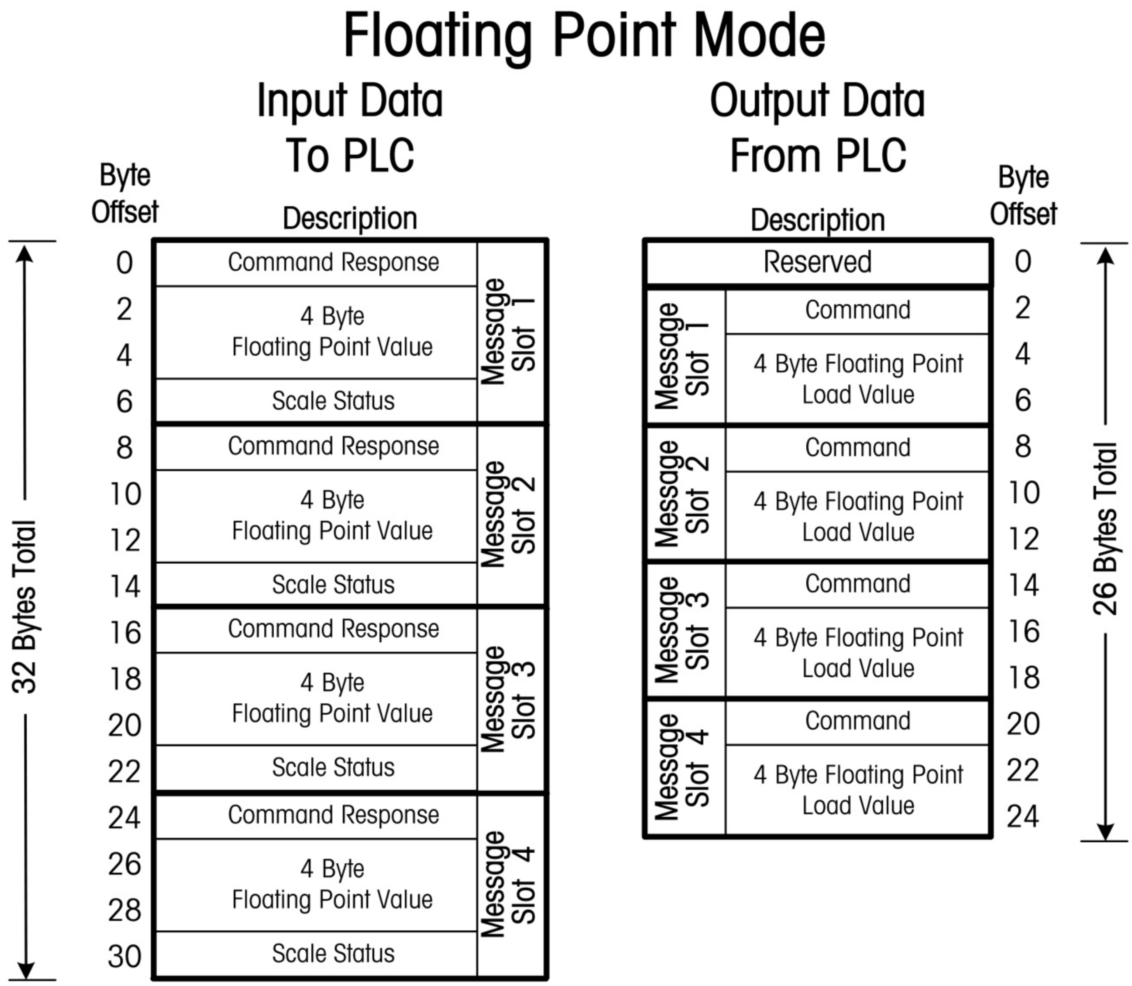


Figure 3-7 Floating Point Message Slot I/O Mapping

3.4. Controlling the Discrete I/O Using a PLC Interface

The IND700 terminal provides the ability to directly control its discrete outputs and read its discrete inputs via the (digital) PLC interface options. System integrators should be aware that the IND700 terminal's discrete I/O updates are synchronized with the terminal's interface update rate and not with the PLC I/O scan rate. This may cause a noticeable delay in reading inputs or updating outputs as observed from the PLC to real world signals. Consult the IND700 Terminal Technical Manual for discrete I/O wiring.

3.5. Shared Data Access

The Shared Data mode PLC communications is provided using Acyclic messaging to the IND700 terminal.

The IND700 Shared Data document lists the Shared Data Variables available to Ethernet/IP, ControlNet, and PROFINET. This document also includes the hex Class Code, Instance and Attribute for the shared data. The PLC must use a combination of RDREC (SFB52) and WRREC (SFB53) to read a Shared Data Variable and WRREC (SFB53) to write a Shared Data Variable.

3.6. Software Setup

When the IND700 terminal detects the presence of a PROFINET Kit option board, the PROFINET parameters are enabled in a Setup program block at **Communication > Industrial Network.** Figure 3-8 shows a typical PLC Menu screen in setup.

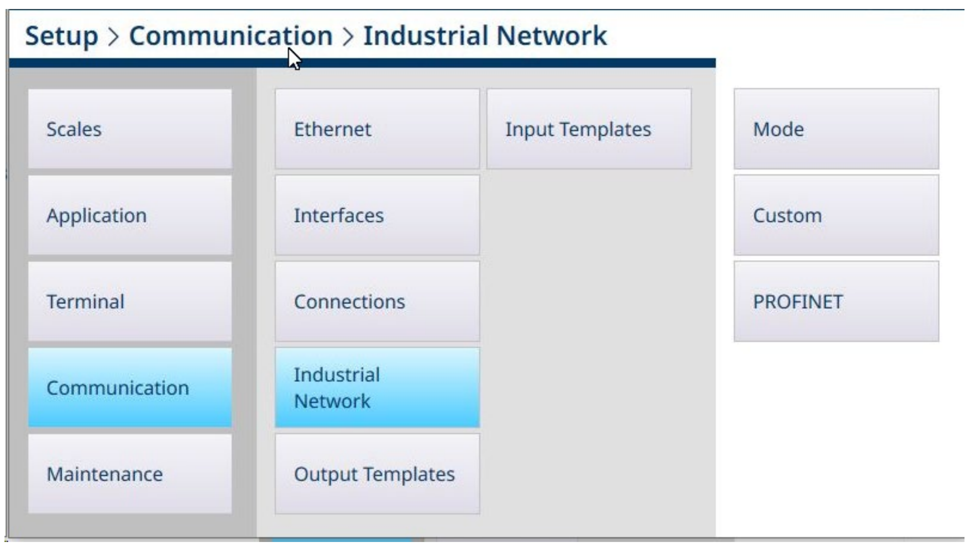


Figure 3-8: Industrial Network Setup Menus

The **Mode** menu permits the selection of either **SAI** or **Custom**. The second menu is either **SAI** or **Custom** depending on the **Mode** selection. The third PLC menu configures the specific PLC interface detected.

3.6.1. Profinet Setup Menu

Figure 3-9: PROFINET Setup Menu

The Profinet interface IP address can be set either using DHCP, or by entering values manually in the IP Address, Subnet Mask and Gateway Address fields.

3.6.2. SAI Block Basic Configuration

If the PLC Mode is set to Custom, the following menu will be available:

Figure 3-10: Industrial Network Options, Mode = Custom

3.6.3. Custom Menu Basic Configuration

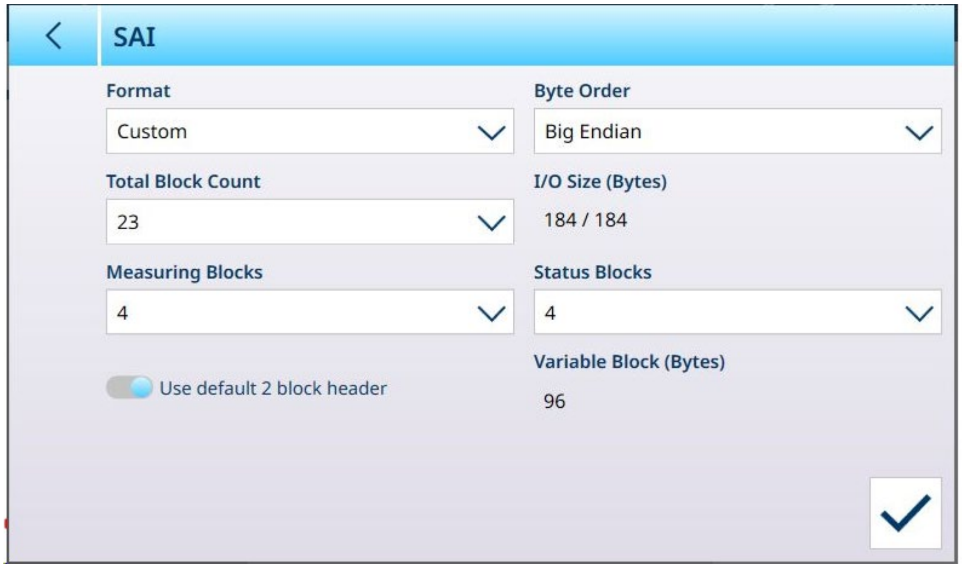


Figure 3-11: SAI Configuration Options

3.7. PROFINET GSDML File

The PROFINET GSDML file contains eight (8) Input configurations, and eight (8) Output configurations. It is very important that the sizes selected for the input and output configurations match each other. For example, if "FLOAT 1 Slot" is the Input selection, then "FLOAT 1 Slot" must also be the Output selection.

The number of slots designated in each configuration references the number of Message Slots configured in the IND700 itself.

- Note: The PROFINET GSDML file for the IND700 and complete versions of the programming examples can be downloaded from www.mt.com/IND700-downloads. The following screen images are provided for illustrative purposes only.

Figure 3-12 shows two IND700's placed on the PROFINET I/O Network. Node 1(IND700) is configured as a Floating Point device, while node 2 (IND700-1) is configured as an Integer/Divisions type device.

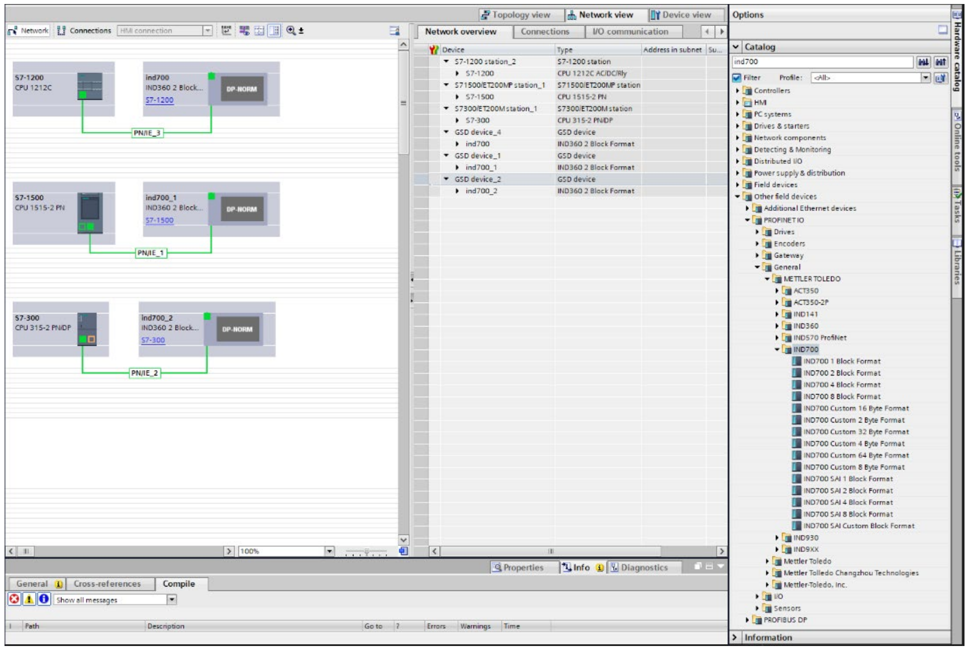


Figure 3-12: Hardware Network Setup

3.8. Assigning the IP Address and Device Name

This function is accessed via the PLC Engineering Software as shown below.

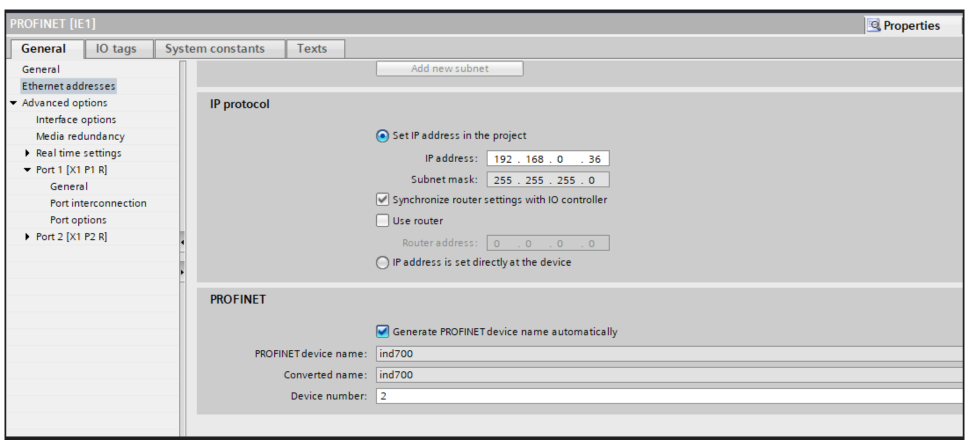


Figure 3-13: IND700 Device Properties- Ethernet Address

Setup the IP address and PROFINET device name in IND700, go to Setup > **Communications** > **Industrial Network** > **PROFINET** by clicking on it, then clicking the OK button to continue.

Figure 3-14: PROFINET MAC Address in IND700 Setup Screen

3.9. Troubleshooting

If the IND700 does not communicate with the PLC, do the following:

- Confirm that both the IP Address configuration and the Device Name configuration have been assigned in the PLC (note that the Device Name must always be assigned using DCP). Cycle power on the IND700 to ensure that any updated settings take effect.
- Check for IP Address conflicts. Use a Ping command from a PC to verify addresses.
- Check physical wiring and network connections.
- Confirm that the IND700 settings for data type and IP Address assignment match those in the PLC and that each IND700 on the network has a unique address.
- Confirm that the number of message slots assigned within the IND700's setup menu match both the Input and Output assignment in the Siemen's HW Configuration Tool.
- If the communication interface in the IND700 was changed from another type (i.e. EtherNet/IP), a master reset of the IND700 may need to be performed.
- Replace the PROFINET interface kit if communication problems persist.

3.9.1. Diagnostic LEDs

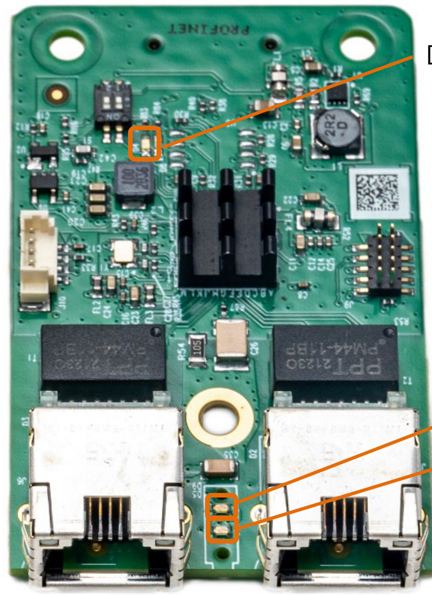


Figure 3-15: Profibus Board LED Locations

3.9.1.1. System

The System LED provides overall system information for the option board.

Table 3-4: Profibus System LED Indications

Color	State	Description
Gray	Off	• No supply voltage or hardware defect • Firmware reset in progress
Green	On Solid	Normal Operation
Green	Blinking	File system being formatted
Yellow	On Solid	A system error has occurred.
Yellow/Green	Alternating Colors, 3x Yellow then 3x Green	Unrecoverable firmware crash.
Yellow/Green	Alternating Colors, 1 Hz	Firmware update in progress.
Yellow/Green	Alternating Colors, 4 Hz	Firmware update in progress.

3.9.1.2. Module Status

Table 3-5: PROFIBUS Module Status LED Indications

Color	State	Description
Gray	Off	No Power
Green/Red	Alternating Colors	Self Test
Green	Blinking	Standby

Color	State	Description
Green	On Solid	Operational
Red	Blinking	Major Recoverable Fault
Red	On Solid	Major Unrecoverable Fault

3.9.1.3. Network Status

Table 3-6: Profibus Network Status LED Indications

Color	State	Description
Gray	Off	No Power, No IP Address
Green	Blinking	No Connections
Green	On Solid	Connected
Green/Red	Alternating Colors	Self-test
Red	On Solid	Duplicate IP

3.10. PROFINET Sample Code

3.10.1. Overview

This section describes the integration of the IND700 with a PROFINET PLC. For additional documentation and code sample, visit www.mt.com/IND700-downloads, or scan the QR code:

- The configuration used in this sample code is based on the following default settings:
  - Siemens TIA Portal V15
  - SAI Data Format: 2-Block
  - Device Name: [empty]
  - IP Address: [empty]
  - IND700 firmware version: 5.41.xx
  - GSDML File: **GSDML-V2.42-MT-IND700-20230424.xml**

METTLER TOLEDO recommends integrating one IND700 into the PLC PROFINET network, and then going through the sample code to understand the functionality of each Function Block. To add more IND700s into the PROFINET network, follow the steps listed in section 3.10.5., **Adding a New IND700**.

3.10.2. Setup of Project Development Environment

3.10.2.1. Hardware Integration

Connect the Ethernet cable from the PLC Ethernet port to IND700 industrial Ethernet port (X1.1 or X1.2).

3.10.2.2. LLDP Function

LLDP (Link Layer Discovery Protocol) is a protocol used for topology discovery in the Siemens PROFINET IO systems. It provides the option of communicating data between neighboring devices (e.g. device name, port, MAC address). IND700 terminals which include the PROFINET option board support this protocol.

With LLDP, the downtime caused by IND700 replacement can be minimized. There is no need to reconfigure the device's IP Address and Device Name, as long as the new device is connected to the PROFINET network via the same physical network port as the previous device.

3.10.2.3. Open the Sample Code

To open and use the **IND700_SampleCode_V15.ap15**, Siemens TIA Portal version 15 SP1 or higher is required. All the required GSDML files will be installed automatically when the sample code is opened.

3.10.2.4. Switching Project Languages

Under **Tools > Project Languages > Editing Language**, choose the preferred language for your project. Selections are English (United States) and Chinese (People's Republic of China).

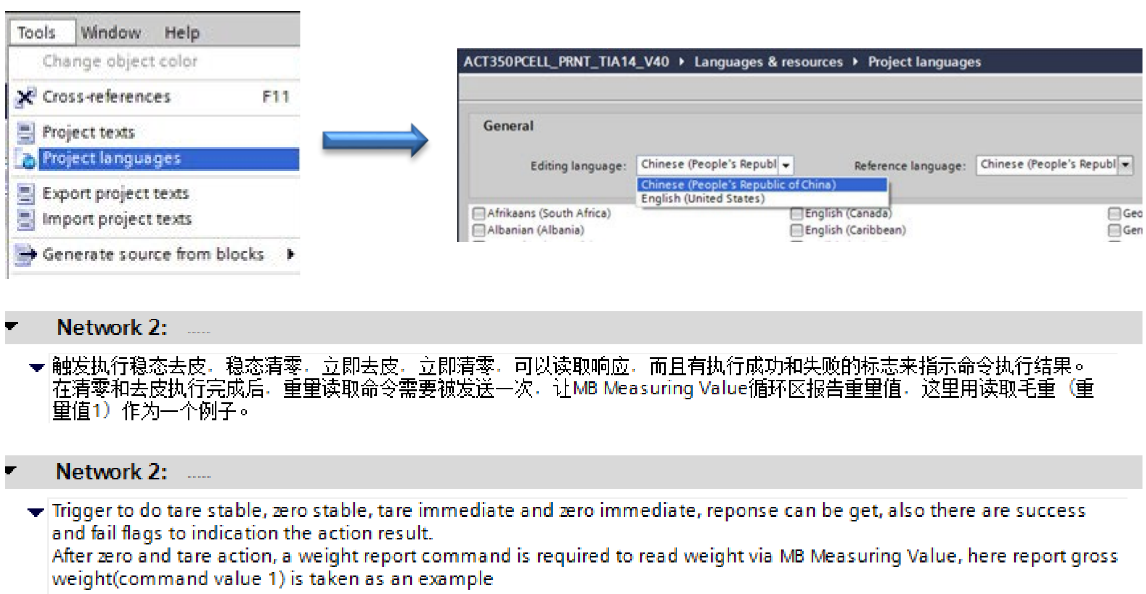


Figure 3-16: Switching Project Languages Between English and Chinese

3.10.2.5. Select the Correct Controller Model

Three projects are included in one sample code. Each project uses different Siemens PLC model:

1. "S7-300" uses S7-300 series PLC with IND700 weighing terminal;
2. "S7-1200" uses S7-1200 series PLC with IND700 weighing terminal;
3. "S7-1500" uses S7-1500 series PLC with IND700 weighing terminal;

Choose the project most relevant to your PLC type, and download it to the PLC.

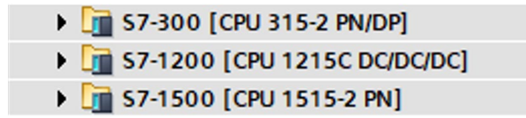


Figure 3-17: Three Projects in the Sample Code

To change the PLC model: access **Device Configuration** under the project folder, right click on the current controller, select **Change Device**, and choose the new controller and its firmware version.

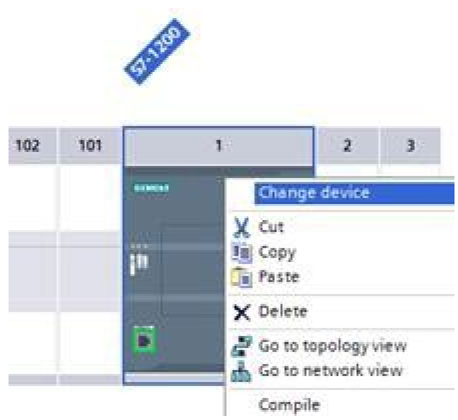


Figure 3-18: Change Controller Type

Compile and download the project into the controller.

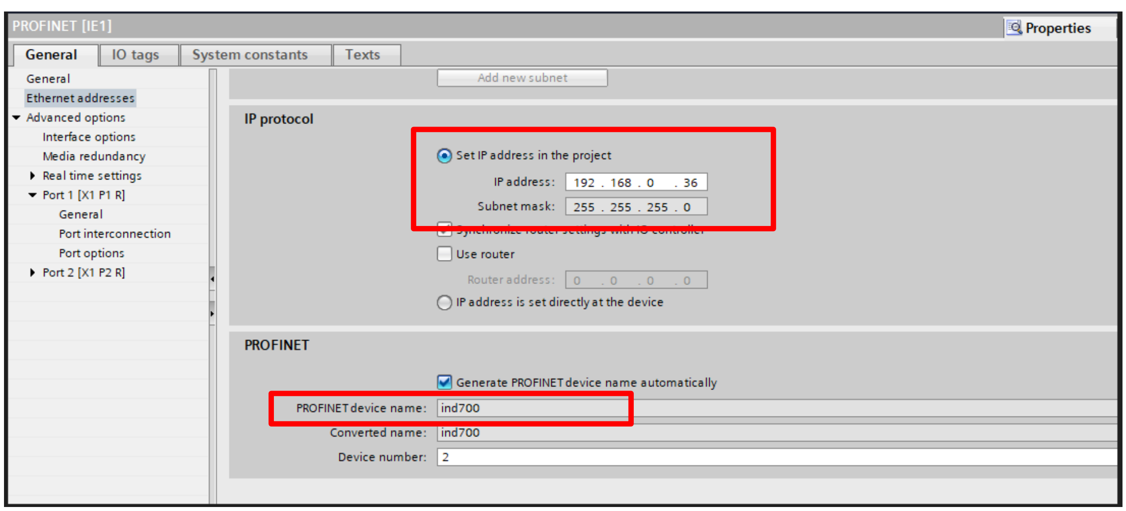


Figure 3-19: IND700 Device Properties – Ethernet Addresses

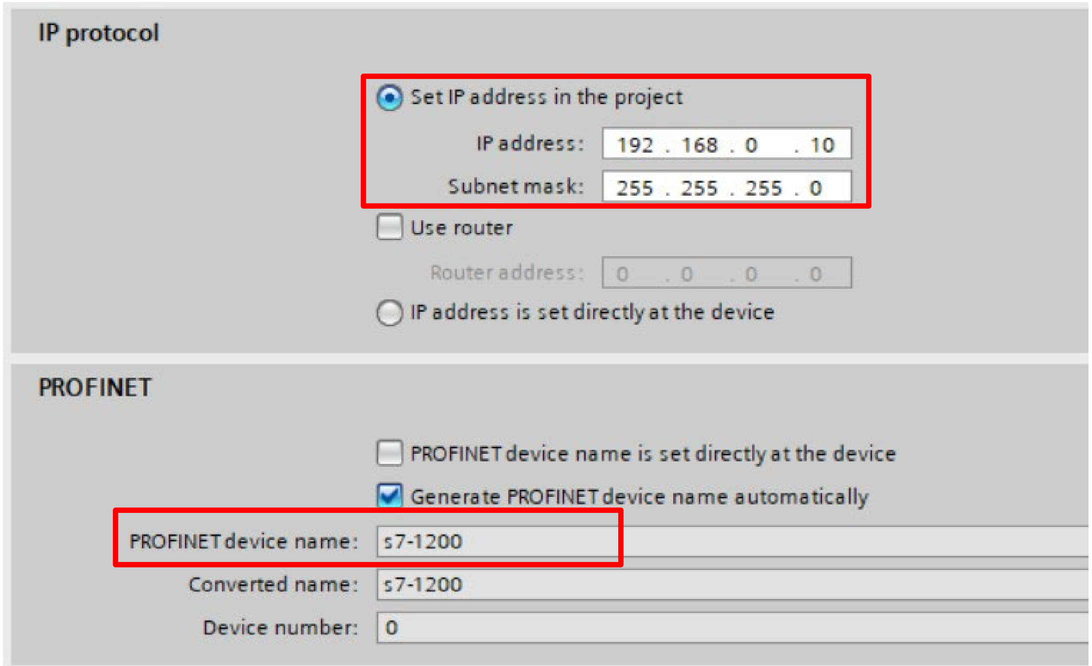


Figure 3-20: PLC Device Properties – Ethernet Addresses

Select the **MT_IND_Application** program and click the **Go Online**" button to start using the sample code.

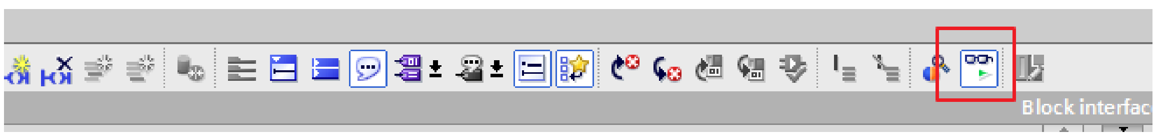


Figure 3-21: Go Online with MT_IND_Application

3.10.3. SAI Data Structure in Device Overview

In the Device Overview tab, the SAI input and output data structure has been assigned with the respective I and Q addresses as shown below. For more details on SAI data structure, please refer to the **SAI Reference for Transmitters and Terminals** (document 30587511).

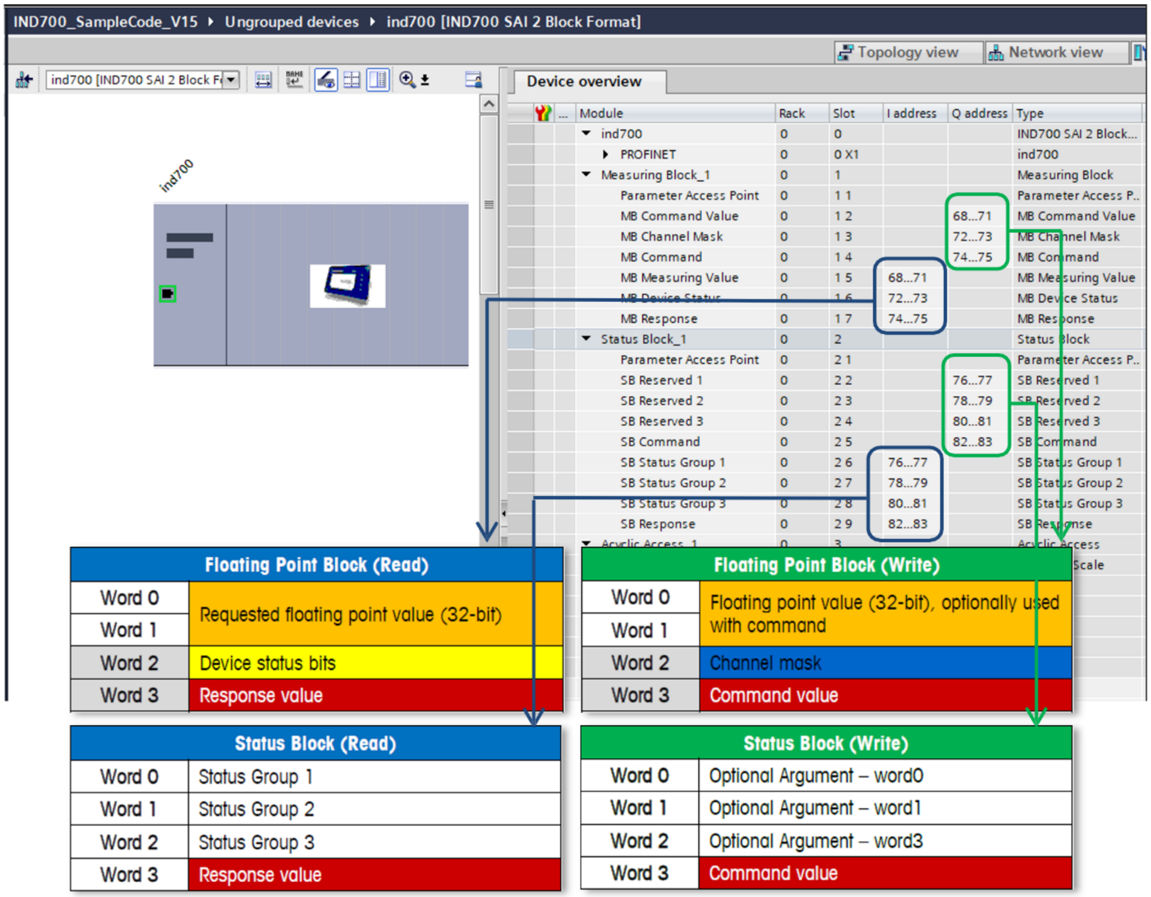


Figure 3-22: SAI Data Structure As Shown in the Device Overview

3.10.4. Function Blocks

- About the **ID** input parameter for all acyclic communication function blocks:
  - For all function blocks which involve acyclic communication between the PLC and the weighing transmitter, the **ID** input parameter is required.
  - For an S7-300, **ID** can be found in the **Device Overview** tab at the Diagnostics Address of Rack 0, Slot 0. In the example shown below, the ID is **2042**.

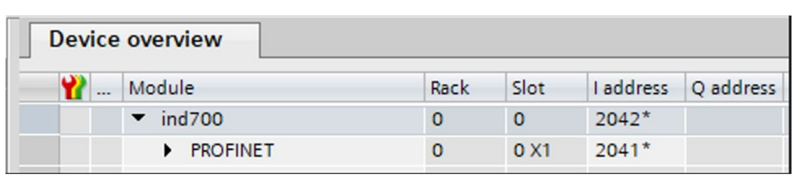


Figure 3-23: ID Parameter for S7-300

3.10.4.1. Cyclic Weight Data Processing

This function block reads in all the important real-time, cyclical weighing data such as weight value, Data OK bit, Motion bit, Net mode bit and critical alarm bit.

Set the scale command bits one at a time to trigger different commands such as tare stable, zero stable, tare immediate, zero immediate, preset tare and clear tare. A successful execution of a scale command will set the **Done** bit on. If a scale command does not execute successfully, the **Error** bit will be set on instead.

Cyclic weight data can be reported automatically immediately after any scale command. The type of weight data (gross, net, or tare) being reported depends on the setting for **WeightCmd**. By default, the **WeightCmd** is decimal "3" and the function block will return a net weight value every time after any scale command such as tare or zero. Similarly, if the **WeightCmd** parameter is configured as decimal "0" or "1" the function block will return a gross weight after any scale command.

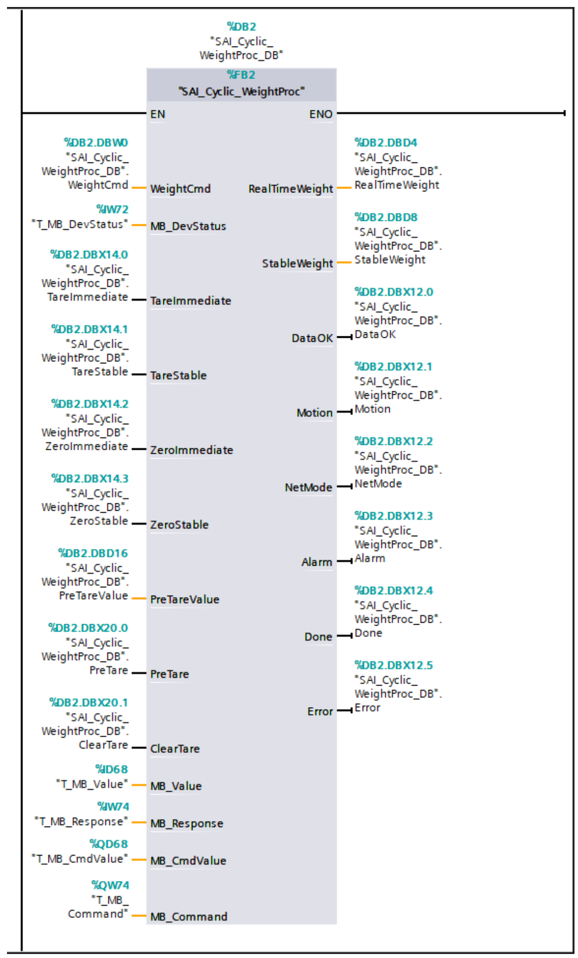


Figure 3-24: SAI_Cyclic_WeightProc Function Block

Table 3-7: SAI_Cyclic_WeightProc Function Block Input Parameters

Parameter	Data Type	Values	Description
WeightCmd	Word	0, 1	Report gross weight value
		2	Report tare weight value
		3 (default)	Report net weight value
		5	Report gross weight value (with internal resolution)
		6	Report tare weight value (with internal resolution)
		7	Report net weight value (with internal resolution)
MB_DevStatus	Word		Refer to the Device Overview tab, input address of MB Device Status
TareImmediate	Bool		Trigger this bit to perform immediate tare command. This tare command doesn't check for stability criteria. Upon completion of this command, the input bit will be reset.
TareStable	Bool		Trigger this bit to perform stable tare command. This tare command requires the weight value to remain stable within the stability criteria (+-1d within 0.3 second) for a predefined timeout range (3 seconds by default), failing which, the command will return an error. Upon completion of this command, the input bit will be reset.
ZeroImmediate	Bool		Trigger this bit to perform immediate zero command. The zero command can only be executed when the weight value is within the zero range (+-2% by default). Else, the command will return an error. Upon completion of this command, the input bit will be reset.
ZeroStable	Bool		Trigger this bit to perform a stable zero command. This zero command requires the weight value to remain stable within the stability criteria (+-1d within 0.3 second) for a predefined timeout range (3 seconds by default). Furthermore the weight value has to be within the zero range to trigger this command. If either condition fails, the command will return an error. When this command completes, the input bit will be reset.
PreTareValue	Real		The preset tare value which must be configured before issuing the PreTare command. Valid PreTare values are from the scale's zero point up to its maximum capacity.
PreTare	Bool		Trigger this bit to perform a preset tare command. The PreTareValue must be configured prior to issuing this PreTare command. When this command completes, the input bit will be reset.
ClearTare	Bool		Trigger this bit to perform a clear tare command. This command removes the tare and brings the scale into gross mode.

Parameter	Data Type	Values	Description
			When this command completes, the input bit will be reset.
MB_Value	Real		Refer to the Device Overview tab,, input address of MB Measuring Value
MB_Response	Word		Refer to the Device Overview tab,, input address of MB Response
MB_CmdValue	Real		Refer to the Device Overview tab,, output address of MB Command Value
MB_Command	Word		Refer to the Device Overview tab,, output address of MB Command

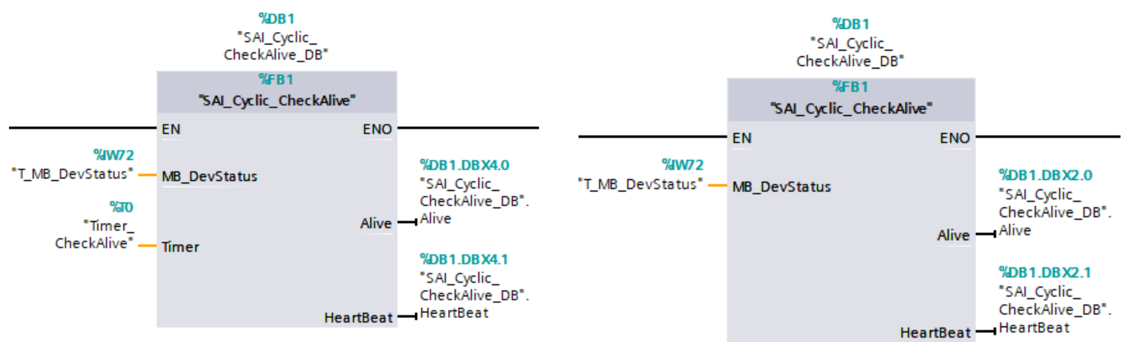
Table 3-8: SAI_Cyclic_WeightProc Function Block Output Parameters

Parameter	Data Type	Values	Description
RealTimeWeight	Real		Real-time weight value, can be gross, tare or net weight
StableWeight	Real		Stable weight value, the last real-time weight during Motion = 0
DataOK	Bool	0	This bit gets set to 0 when the device is still operational but the value being reported cannot be guaranteed to be valid. The following conditions cause the Data Okay bit to be set to 0: Device is powering up Device is in setup mode Device is in test mode Over capacity condition occurs When the A/D converter is at its limit Product dependent over capacity that occurs when the device determines it cannot trust the weight Under capacity condition occurs When the A/D converter is at its limit Product dependent under capacity that occurs when the device determines it cannot trust the weight
		1	Weight data is normal, valid
Motion	Bool	0	Weight value is stable
		1	Weight value is in motion
NetMode	Bool	0	Weighing is in gross mode
		1	Weighing is in net mode
Alarm	Bool	0	No alarm
		1	Also called the RedAlert alarm. When this bit is true this indicates that the control device should stop until the source of the alarm is evaluated and corrected. The control system should use a Field Value command or evaluate the RedAlert status block to determine the nature of the alarm.
Done	Bool	0	Zero, tare or clear tare command is in process, or failed

Parameter	Data Type	Values	Description
		1	Zero, tare or clear tare command is successful
Error	Bool	0	Zero, tare or clear tare command is in process, or succeeded
		1	Zero, tare or clear tare command is not completed due to error

3.10.4.2. Device HeartBeat Monitoring

This function block monitors the Heart Beat bit of the weighing transmitter and outputs an "Alive" flag.



S7-300 S7-1200 and S7-1500

Figure 3-25: SAI-Cyclic_CheckAlive Function Block

Table 3-9: SAI-Cyclic_CheckAlive Function Block Input Parameters

Parameter	Data Type	Values	Description
MB_DevStatus	Word		Refer to the Device Overview tab, input address of MB Device Status
Timer (S7-300)	Timer		Timer, use independent timer for each function block, do not replicate.

Table 3-10: SAI-Cyclic_CheckAlive Function Block Output Parameters

Parameter	Data Type	Values	Description
Alive	Bool	0	Device has lost communication
		1	Device is communicating OK
HeartBeat	Bool		To ensure that the device is working as expected and updating data in Words 0, 1 and 2, this heart beat bit is toggled between off and on states. The frequency is dependent on the specific device’s ability to cycle this bit. For example, a 1 second heartbeat would be sufficient for most applications.

3.10.4.3. Hardware Configuration

1. Under **Devices & Networks > Network View**, add (or drag over) an IND700 2P 2 Block Format.

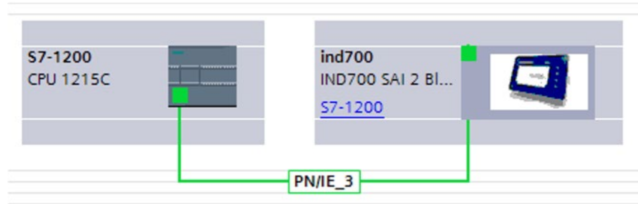


Figure 3-26: Add a PROFINET Device in the Network View

2. Assign the independent PROFINET device name and IP address for the added device.

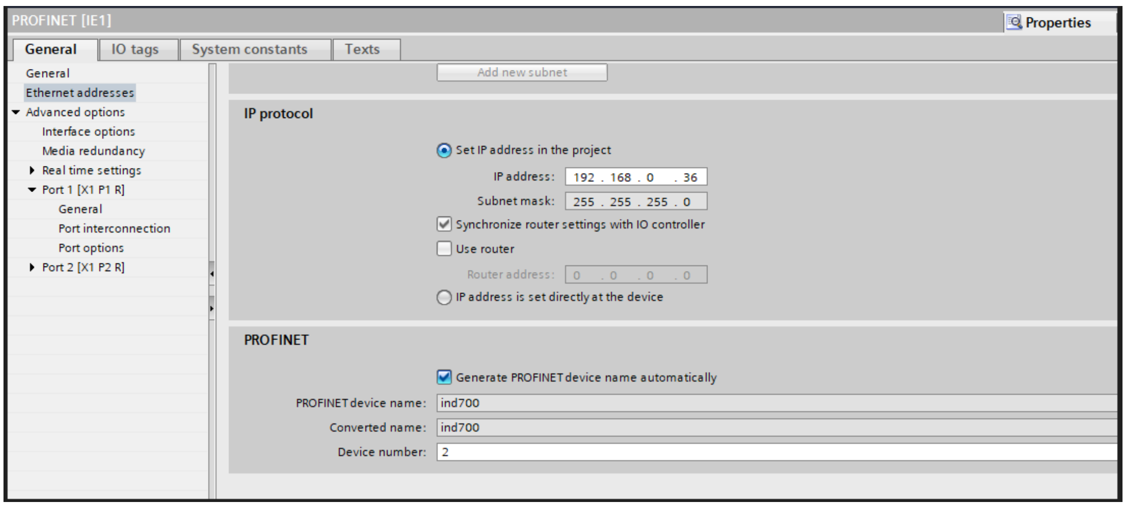
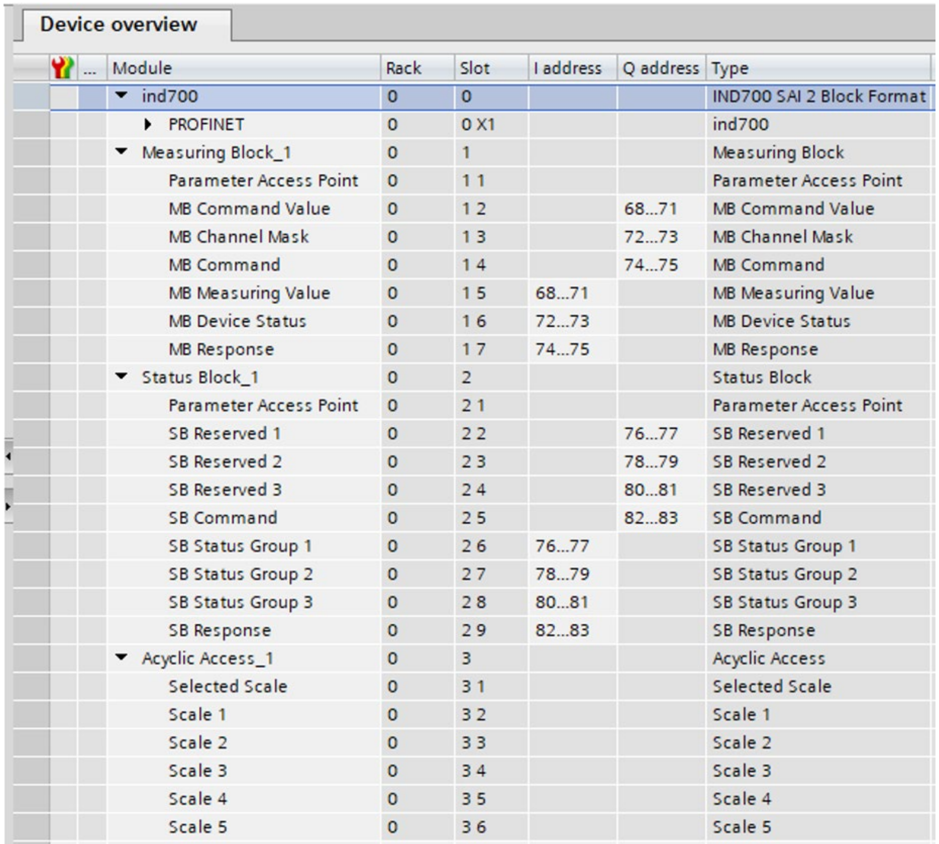


Figure 3-27: PROFINET Device Name and IP Address

3. Under **Devices & Networks > Topology View**, link up the PLC and the IND700's network port 1 (left, NW1).

Figure 3-28: Devices and Networks, Topology View

4. The sample code follows the default I and Q addresses assignment shown below. To minimize modifications to the code, consider using the same I and Q address assignments.



Device overview

Module	Rack	Slot	I address	Q address	Type
ind700	0	0			IND700 SAI 2 Block Format
PROFINET	0	0 X1			ind700
Measuring Block_1	0	1			Measuring Block
Parameter Access Point	0	1 1			Parameter Access Point
MB Command Value	0	1 2		68...71	MB Command Value
MB Channel Mask	0	1 3		72...73	MB Channel Mask
MB Command	0	1 4		74...75	MB Command
MB Measuring Value	0	1 5	68...71		MB Measuring Value
MB Device Status	0	1 6	72...73		MB Device Status
MB Response	0	1 7	74...75		MB Response
Status Block_1	0	2			Status Block
Parameter Access Point	0	2 1			Parameter Access Point
SB Reserved 1	0	2 2		76...77	SB Reserved 1
SB Reserved 2	0	2 3		78...79	SB Reserved 2
SB Reserved 3	0	2 4		80...81	SB Reserved 3
SB Command	0	2 5		82...83	SB Command
SB Status Group 1	0	2 6	76...77		SB Status Group 1
SB Status Group 2	0	2 7	78...79		SB Status Group 2
SB Status Group 3	0	2 8	80...81		SB Status Group 3
SB Response	0	2 9	82...83		SB Response
Acyclic Access_1	0	3			Acyclic Access
Selected Scale	0	3 1			Selected Scale
Scale 1	0	3 2			Scale 1
Scale 2	0	3 3			Scale 2
Scale 3	0	3 4			Scale 3
Scale 4	0	3 5			Scale 4
Scale 5	0	3 6			Scale 5

Figure 3-29: Device I and Q Addresses

3.10.4.4. PLC Settings

1. Under the **PLC Device Properties > Advanced Options**, check the two options shown below to support the LLDP feature.

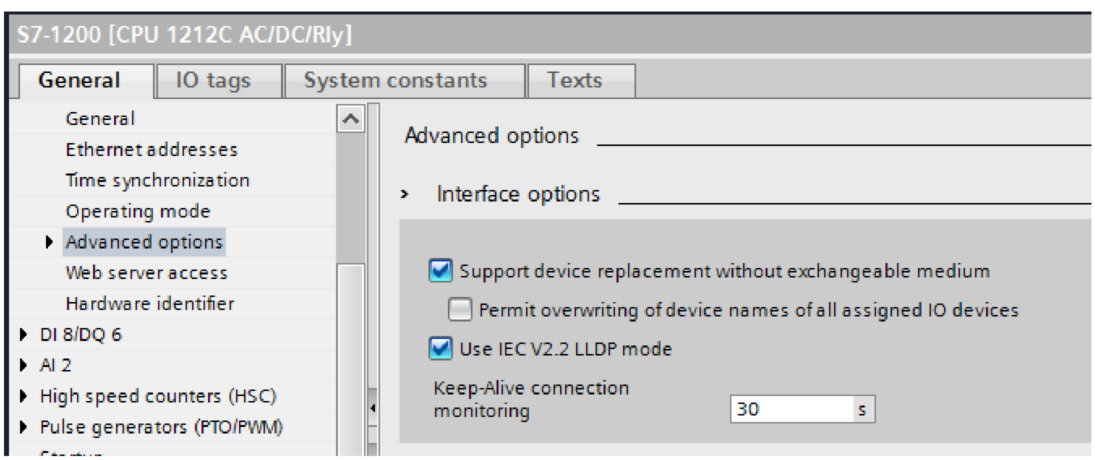


Figure 3-30: Enabling the LLDP Feature

2. Under the **PLC Device Properties > System and Clock Memory**, check **Enable the use of system memory byte** (this feature is not available in the S7-300 series PLC).

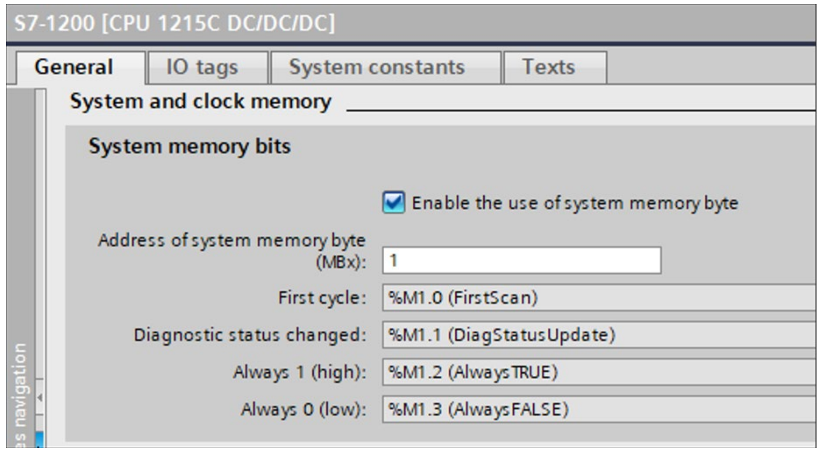


Figure 3-31: Enabling the System Memory Byte

3.10.4.5. Duplicate Programming Files

1. The required program blocks are:
   - MT_IND_Application(FC)
   - SAI_Copy(FC) (for S7-1200 and S7-1500, not for S7-300）
   - SAI_Cyclic_WeightProc(FB), SAI_ Cyclic_WeightProc_DB
   - SAI_Cyclic_CheckAlive(FB), SAI_Cyclic_CheckAlive_DB
   - SAI_Buffer(DB600). **Note:** do not modify this Data Block's number as other Function Blocks are referring directly to its DB number.

   For S7-300, it is necessary to add the COMPLETE RESTART(OB100) and ErrorHandle programs, as shown below, to support PROFINET auto reconnection feature.

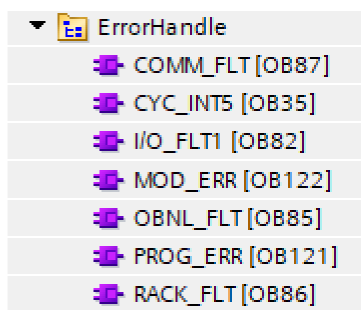


Figure 3-32: ErrorHandle Programs of S7-300

2. Delete the other, unused, program blocks in **MT_ACT_Application**.
3. Duplicate the **IND** under the PLC tags.

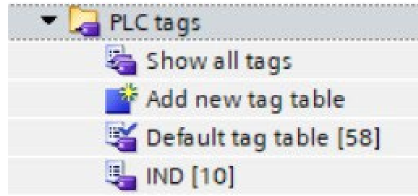


Figure 3-33: Duplicating the PLC Tags

4. Duplicate all the PLC data types.

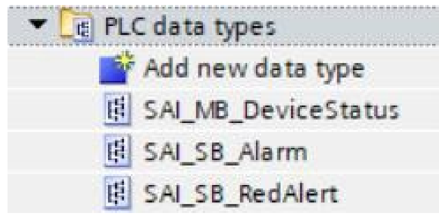


Figure 3-34: Duplicating the PLC Data Types

5. Finally, in the Main (OB1), call up the MT_IND_Application function.

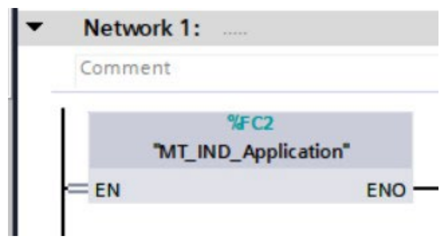


Figure 3-35: Calling Up MT_IND_Application in the Main OB

3.10.5. Adding a New IND700

In a PROFINET system, each PROFINET device is identified with different individual **Device Name**. The same rule applies to a network of multiple IND700s.

1. In **Devices and Networks > Network View**, add another IND700 2 Block Structure.

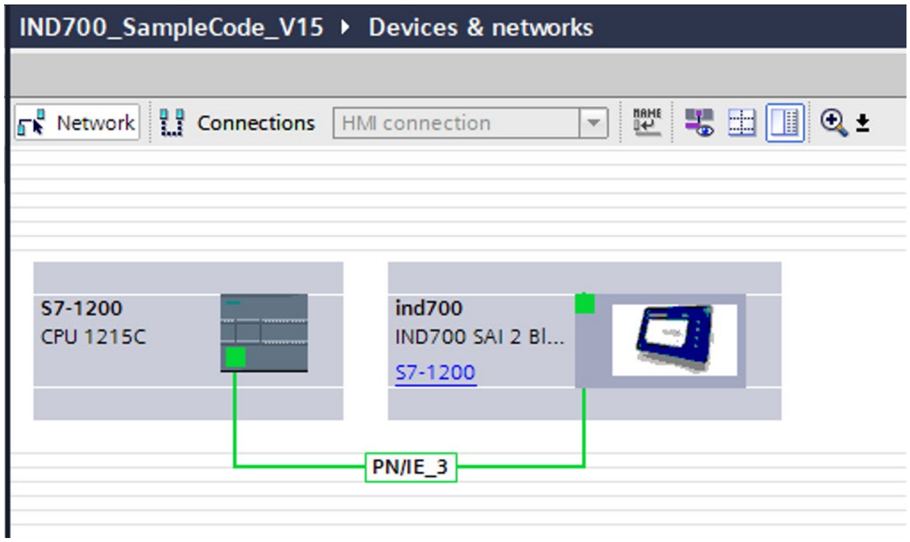


Figure 3-36: Adding Another IND700 Into the Network

2. Configure a dedicated PROFINET device name and IP address for the new IND700.
   - Note: Only use lower case letters for the device name.

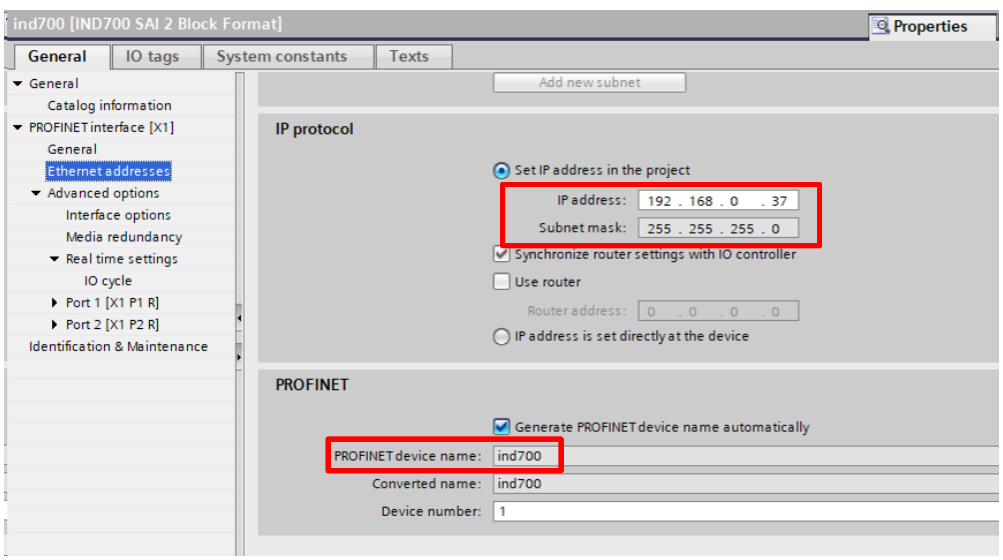


Figure 3-37: PROFINET Device Name and IP Address

3. In **Devices and Networks > Network Overview**, connect the device IND700's second Ethernet port NW2 to the device IND700's first Ethernet port NW1.

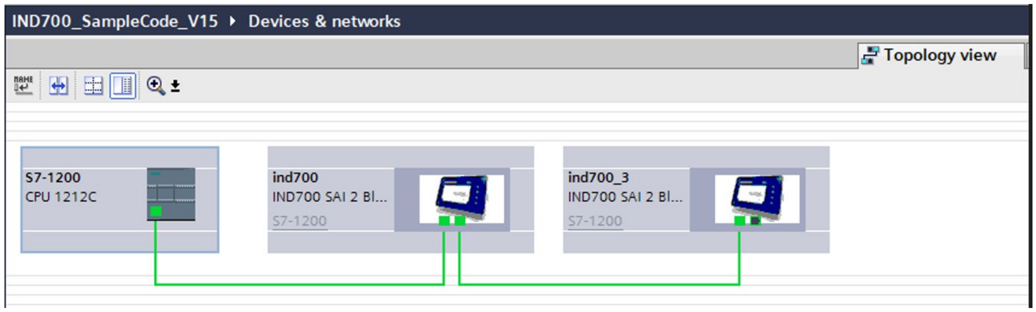


Figure 3-38: Connecting Multiple IND700s to the PROFINET Network

4. When necessary, edit the automatically allocated I and Q addresses of the PROFINET device.

Device overview

Module	Rack	Slot	I address	Q address	Type
ind700_3	0	0			IND700 SAI 2 Block Format
PROFINET	0	0 X1			ind700
Measuring Block_1	0	1			Measuring Block
Parameter Access Point	0	1 1			Parameter Access Point
MB Command Value	0	1 2		80...83	MB Command Value
MB Channel Mask	0	1 3		84...85	MB Channel Mask
MB Command	0	1 4		86...87	MB Command
MB Measuring Value	0	1 5	84...87		MB Measuring Value
MB Device Status	0	1 6	88...89		MB Device Status
MB Response	0	1 7	90...91		MB Response
Status Block_1	0	2			Status Block
Parameter Access Point	0	2 1			Parameter Access Point
SB Reserved 1	0	2 2		88...89	SB Reserved 1
SB Reserved 2	0	2 3		90...91	SB Reserved 2
SB Reserved 3	0	2 4		92...93	SB Reserved 3
SB Command	0	2 5		94...95	SB Command
SB Status Group 1	0	2 6	92...93		SB Status Group 1
SB Status Group 2	0	2 7	94...95		SB Status Group 2
SB Status Group 3	0	2 8	96...97		SB Status Group 3
SB Response	0	2 9	98...99		SB Response
Acyclic Access_1	0	3			Acyclic Access
Selected Scale	0	3 1			Selected Scale
Scale 1	0	3 2			Scale 1
Scale 2	0	3 3			Scale 2
Scale 3	0	3 4			Scale 3
Scale 4	0	3 5			Scale 4
Scale 5	0	3 6			Scale 5

Figure 3-39: I and Q Addresses

5. Duplicate the function blocks and configure all the required input and output parameters. Each function block FB must have an independent data block DB. As shown below, there are two **SAI_Cyclic_CheckAlive** function blocks but both FBs are assigned with different DBs which are **SAI_Cyclic_CheckAlive_DB** (DB1) and **SAI_WeightProc_DB2** (DB3).
   - A small trick can be used here to add adjacent function blocks – drag the function block from the Project Tree pane into the destination network.

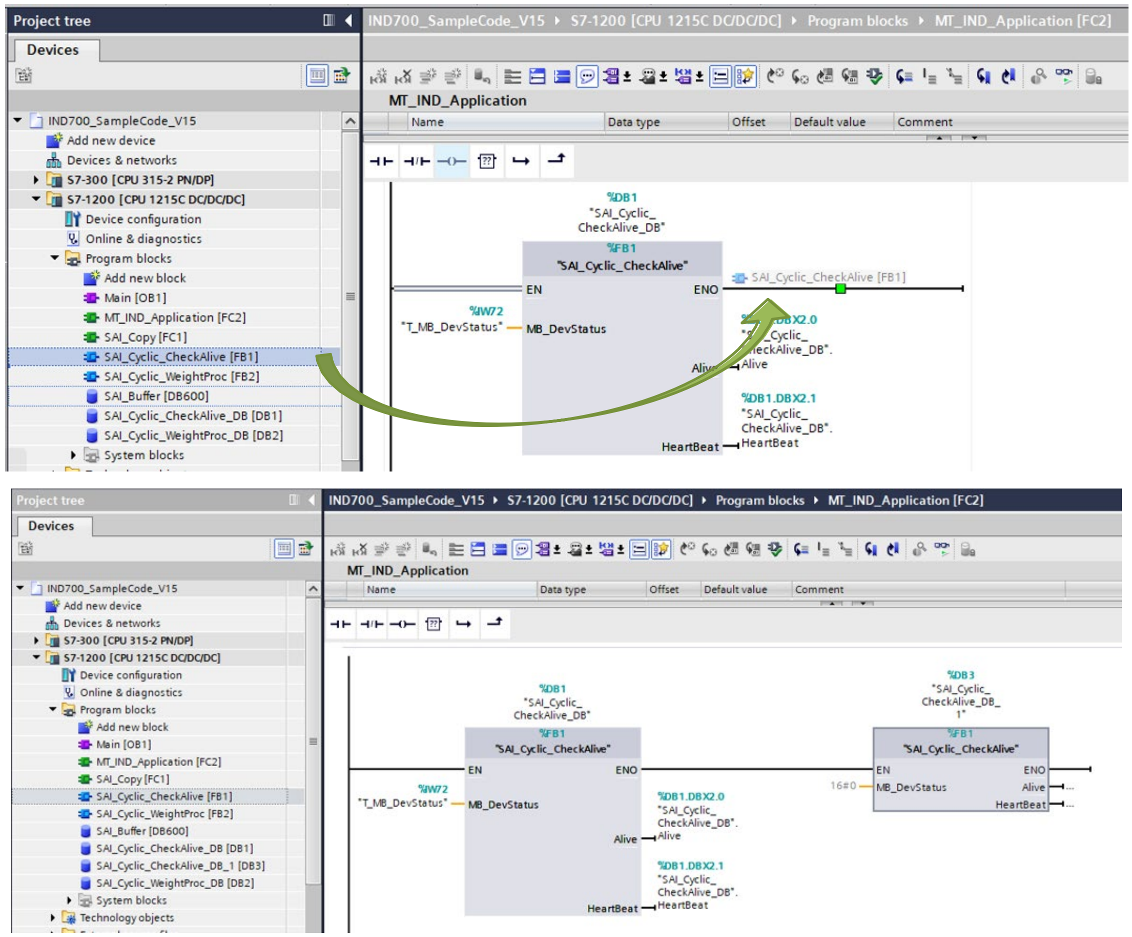


Figure 3-40: Adding Another IND700 Into the Network

6. Repeat steps 1 to 5 until all the new IND700s have been integrated into the PROFINET network.
7. Download the project into the PLC. Assign the IND700 device name and IP address according to the project configuration.

3.10.6. Frequently Asked Questions

1. Q: I have duplicated the SAI_Cyclic_WeightProc function block and SAI_Cyclic_WeightProc_DB data block into another project, but I was not able to read the weight data.
   - A: Make sure the device I and Q addresses are assigned accordingly between the Device overview and the function block assignment. If it is an S7-300 PLC, there is a need to edit the default cyclic data range (128 byte) to cover the device I and Q address range. In this sample code, the PLC's cyclic data range has been configured to 512 bytes.

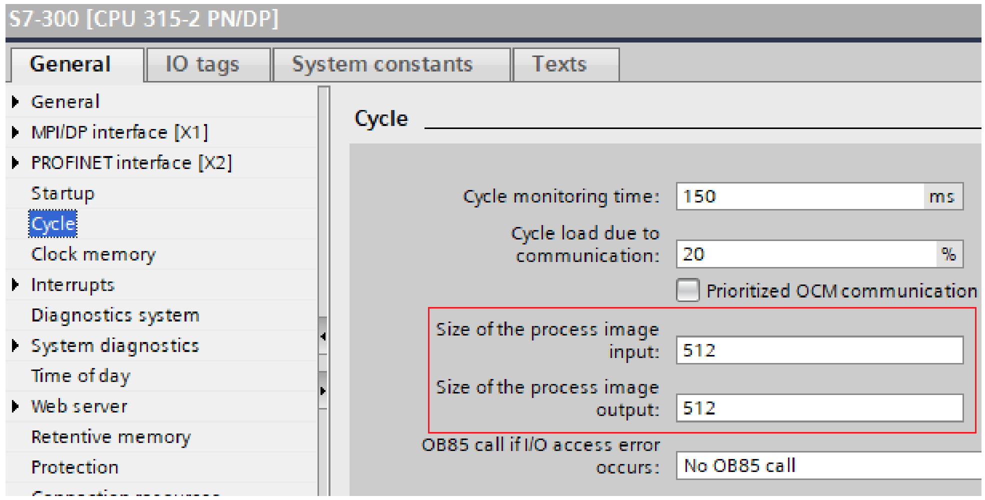


Figure 3-41: Edit the S7-300 PLC Cyclic Data Range

2. Q: How to read the gross, tare or net weight?

   - A: The PLC command to read gross weight is decimal "0" or "1", decimal "2" to read tare weight and decimal "3" to read net weight. Insert one of these decimal command values into the "WeightCmd" input parameter of SAI_Cyclic_WeightProc function block, after a tare or zero command the function block will then return the required weight data accordingly.

3. Q: After I managed to integrate the IND700 to the PLC, why is the IND700's IP address showing 0.0.0.0? A: This is due to the PROFINET protocol. By choosing "Set IP address in the project", while booting up the PLC will assign the IP address to the PROFINET device according to the Device name. Hence with this option, the IND700 will not display its assigned IP address. If the second option "IP address is set directly at the device" is chosen, the PLC will not assign any IP address to the device. With this option, the IND700 will display its own IP address (see below).

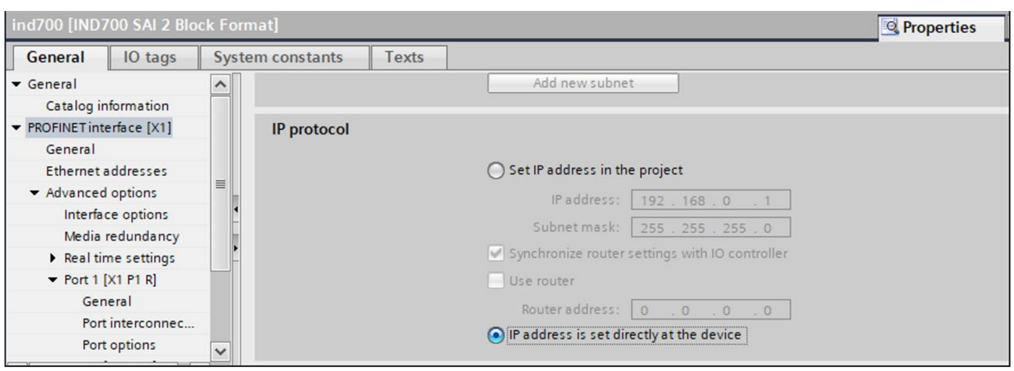


Figure 3-42: The IND700 Displays its own IP Address with this Setting

4. Q: The PLC has activated its LLDP function, but the newly connected IND700 cannot communicate automatically.

   - A: Under Online access, expand the active Ethernet interface, click on Update accessible devices. Look for the newly connected IND700's MAC address, check whether it says "Accessible device [MAC address can be found on the device label]" as shown in Figure 3-28. If the new device has been assigned with Device name and IP address previously, click on Online & diagnostics, then reset the device to factory settings.

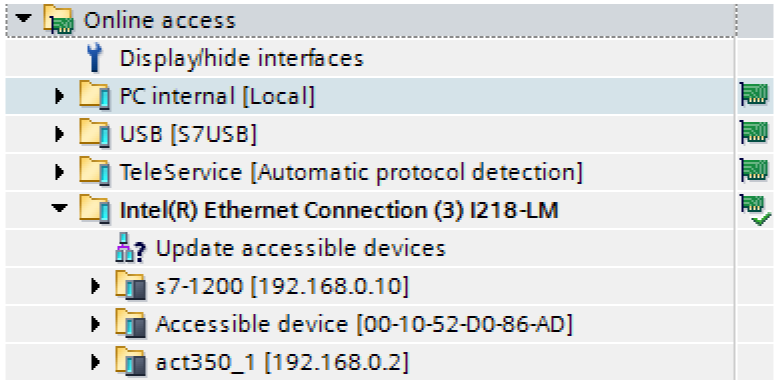


Figure 3-43: The New Device Appears as an Accessible Device

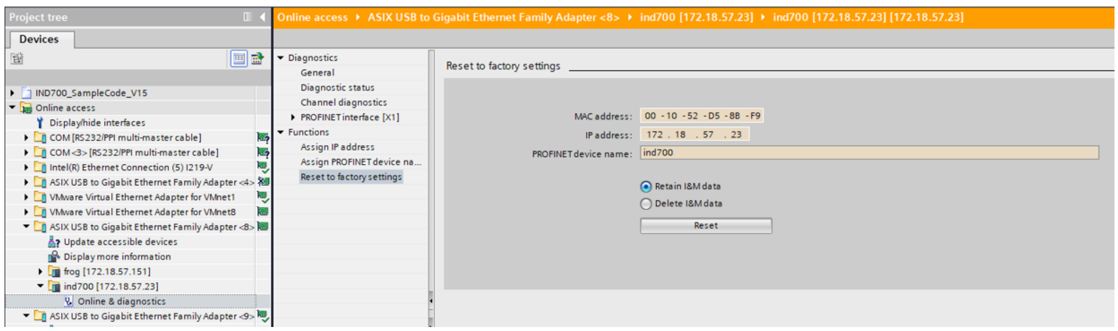


Figure 3-44: Resetting the IND700 to Factory Settings, with no Device Name or IP Address

5. Q: Is the LLDP function still available when a network switch is used?
    - A: A network switch is required to support the LLDP function. After importing the device's GSDML file, configure the Ethernet connection in **Devices & Networks > Topology View**.

A. Integer and Division Formats

When one of these formats is selected, the IND700 will have two 16-bit words for input data and two 16-bit words for output data in each Message Slot. There can be up to four slots and the number of slots is setup at the IND700. The PLC's input data will contain one 16-bit word for the scale's weight information and one 16-bit word for bit encoded status information for each Message Slot. The IND700 will send specific weight data to the PLC input based on the selections the IND700 receives from the PLC's output data. The PLC's output words consist of one 16-bit integer value, which may be used to download a tare or target logic value, and one 16-bit word for bit encoded command information.

The "Select 1, 2 or 3" commands in write word 1 select the type of data that will be returned in the scale data slot. While any type of data can be reported back from any Integer or Division slot, commands such as Tare, Clear and Zero can only be sent to slot 1. (This applies to Integer/Division mode only.)

Table A-1 and Table A-2 provide detailed information on the integer and division data formats. Note that the designation of "Read" or "Write" data is based on the PLC's viewpoint-"Read" data refers to the PLC's input data and "Write" data refers to the PLC's output data.

Table A-1: Discrete Read Integer or Division – IND700 > PLC, per Message Slot

Bit number	First Word	Second Word
0	See Note 1	Target 1[2]
1		Target 2[2]
2		Target 2[2]
3		Comparator 5[3]
4		Comparator 4[3]
5		Comparator 3[3]
6		Comparator 2[3]
7		Comparator 1[3]
8		Enter Key[4]
9		Input 1[5]
10		Input 2[5]
11		Input 3[5]
12		Motion[6]
13		Net Mode[7]
14		Update in Process[8]
15		Data OK[9]

Notes for Table A-1

1 The first word is a 16 bit, signed integer that may represent the indicator's gross weight, net weight, displayed weight, tare weight, or rate. The **bits 0** to **2** in the PLC 2nd output word designate the type of data that is being is being sent by the indicator.

2 The second word **bits 0, 1** and **2** indicate the state of the target comparison logic. When in the material transfer mode; **bit 0** is Feed, **bit 1** is Fast Feed and **bit 2** is Tolerance Ok (within range). When in the over/under mode; **bit 0** is Under, **bit 1** is OK and **bit 2** is Over. An 'ON' condition is indicated by the bit being set to '1'; an 'OFF' condition is indicated by the bit being set to '0'.

3 The second word Comparator bits indicate the state of the associated comparator logic; when the bit is set to '1' the comparator state is 'ON'; when the bit is set to '0' the comparator state is 'OFF'. The setup of each comparator will determine when the state is 'ON' or 'OFF'.

4 The second word **bit 8** is set to '1' when the Enter Key has been pressed on the indicator keypad. The bit can be reset to '0' by changing the state of the second output word **bits 9, 10** and **11**.

5. The second word **bits 9, 10**, and **11** indicate the state of the associated hardware input internal to the indicator; these are 0.1.1, 0.1.2 and 0.1.3. When the input is 'ON" the associated bit is set to '1'.

6 The second word **bit 12**; The motion bit is set to '1' when the scale is in motion (unstable).

7 The second word **bit 13**; The net mode bit is set to '1' when scale is in the net mode (a tare has been taken).

8 The second word **bit 14** (update in process) is set to '1' when the indicator is in process of updating the data to the PLC communications adapter. The PLC should ignore all data while this bit is set to '1'.

9 The second word **bit 15**; The data ok bit is set to '1' when the indicator operating conditions are normal. The bit is set to '0' during power-up, during indicator setup, when the scale is over capacity or under zero, and when in the x10 display mode; additionally, the first word integer value is set to '0'. Note that, when in x10 mode, the data sent is 000000. The PLC should continuously monitor the data ok bit in the IND700 communication and also any PLC data connection fault bit that exists in the PLC (refer to the PLC manufacturer's documentation) to determine the validity of the data in the PLC.

Table A-2: Discrete Write Integer or Division –PLC > IND700, per Message Slot

Bit number	First Word	Second Word [Scale Command]
0	See Note 1	Select 1[2]
1		Select 2[2]
2		Select 3[2]
3		Load Tare 1st message slot only[12]
4		Clear Tare[4] 1st message slot only[12]
5		Tare[5] 1st message slot only[12]
6		Print[6] 1st message slot only[12]
7		Zero[7] 1st message slot only[12]

Bit number	First Word	Second Word [Scale Command]
8		Start/Abort Target[8] 1st message slot only[12]
9		Message Display Mode[9] 1st message slot only
10		Message Display Mode[9] 1st message slot only[12]
11		Message Display Mode[9] 1st message slot only[12]
12		Output 1[10] 1st message slot only[12]
13		Output 2[10] 1st message slot only[12]
14		Output 3[10] 1st message slot only[12]
15		Load Target[11] 1st message slot only[12]

Notes for Table A-2

1 First word is a 16 bit, signed integer that represents a value to be downloaded to the indicator. The value represents a tare or target value. When using the divisions format, the data set must be in the number of divisions, not an integer weight value. A value must be loaded in this word before setting the **bits 3** or **15** in the second word. To load the target value ,first enter the value into the first word and then set bit 15 (Load Target) "On"

2 The select bits change the type of data being sent from the indicator in the first word. Use a decimal value in binary format within **bits 0, 1**, and **2** to change the data reported by the indicator. '0' = gross weight, '1' = net weight, '2' = displayed weight, '3' = tare weight, '4' = target, '5' = rate; any value above 5 will equal gross weight.

3 A transition from '0' to '1' will cause the value in the first word to be loaded into the tare register of the indicator and set the indicator into the net mode. Set this bit to '1' only **after** the first word has been loaded with the required value.

4 A transition from '0' to '1' will cause the indicator tare register to be set to '0' and the indicator will be set to the gross weight mode.

5 A transition from '0' to '1' will cause the weight on the scale to be used as the tare value and set the indicator to the net mode (equivalent to a tare command). Note that the scale will not tare while the scale is "In Motion". If the indicator has not tared within 3 seconds, the command must be resent. A good practice is to check for no motion –bit 12 of input word 1-"Off"

6 A transition from '0' to '1' will issue a print command.

7 A transition from '0' to '1' will cause the scale to re-zero, but only within the ranges established in scale setup.

8 A transition from '0' to '1' will cause the target logic to start. A transition from '1' to '0' will cause the target logic to abort. The use of the PLC in conjunction with the indicator console keypad and/or a remote input is not advised, as unexpected results may occur.

9 The message display mode bits will cause messages to be displayed on the indicator display above the soft key prompts; messages are limited to 20 characters. The use of the display mode bits will clear the Enter Key bit in the second word of the indicator output data. The message display mode bits cause a value to be written to shared data pd0119, which is available for use by Task Expert applications. The transition from ‘0’ to a decimal value in binary form to the second word **bits 9, 10** and **11** will initiate the message events.

Setting the message display bits to a value of ‘1’ will cause the characters in shared data aw0101 to be displayed and pd0119 will be set to ‘1’.

Setting to ‘2’ = display aw0102 and pd0119 = ‘2’.
Setting to ‘3’ = display aw0103 and pd0119 = ‘3’.
Setting to ‘4’ display aw0104 and pd0119 = ‘4’.
Setting to ‘5’ = display aw0105 and pd0119 = ‘5’.
Setting to ‘6’ = start Prompt sequence, pd0119 = ‘6’ and xc0134 = ‘1’.
Setting to ‘7’ = display pd0118 and pd0119 = ‘7’.

The message display mode bits must return to ‘0’ before a new message can be displayed.

10 The output bits will cause the associated hardware output to be turned ‘ON’ and ‘OFF’. This is the indicator internal outputs only; 0.1.1, 0.1.2 and 0.1.3. The output bits will not override the hardware outputs being used by the indicator logic as setup within the indicator. Setting a bit to ‘1’ will cause the output to turn ‘ON’; setting the bit to ‘0’ will cause the output to turn ‘OFF’.

11 A transition from ‘0’ to ‘1’ will cause the value in the first word to be loaded into the target register of the indicator and will be used the next time the target logic is started. Set this bit to ‘1’ only **after** the first word has been loaded with the required value.

12 These are bit commands to the indicator that function only in the first message slot.

B. Floating Point Format

B.1. Operational Overview

The IND700 uses integer commands from the PLC to select the floating point weight input data. The IND700 recognizes a command when it sees a new value in the Message Slot command word. If the command has an associated floating point value (for example: loading a target value), it must be loaded into the floating point value words before the command is issued. Once the IND700 recognizes a command, it acknowledges the command by setting a new value in the command acknowledge bits of the scale's command response word. The IND700 also tells the PLC what floating point value is being sent (via the floating point input indicator bits of the command response word). The PLC should wait until it receives the command acknowledgment from the IND700 before sending another command.

The IND700 can report two types of values to the PLC: real-time and static. When the PLC requests a real-time value, the IND700 acknowledges the command from the PLC once but sends and updates the value at every interface update cycle. If the PLC requests a static value, the IND700 acknowledges the command from the PLC once and updates the value once. The IND700 will continue to send this value until it receives a new command from the PLC. Gross weight and net weight are examples of real-time data. Tare weight, target, feed, and tolerance values are examples of static data.

The IND700 can send a rotation of up to nine different real-time values. The PLC sends commands to the IND700 to add a value to the rotation. Once the rotation is established, the PLC must instruct the IND700 to begin its rotation automatically, or the PLC may control the pace of rotation by instructing the IND700 to advance to the next value. If the IND700 is asked to automatically alternate its output data, it will switch to the next value in its rotation at the next interface update cycle. (The interface update cycle has an update rate of up to 20 Hz or 58 milliseconds.)

The PLC may control the rotation by sending alternate report next field commands (1 and 2). When the PLC changes to the next command, the IND700 switches to the next value in the rotation order. The IND700 stores the rotation in its shared data so the rotation does not have to be re-initialized after each power cycle. When the PLC does not set up an input rotation, the default input rotation consists of gross weight only. See the floating-point command examples in Table B-5 through Table B-8 for additional information. The method of handling string and floating point data varies between PLC types. The IND700 provides floating point data in the order entered in Data Format setup.

B.2. Floating Point Data Format and Compatibility

In Floating Point Data Format, the PLC and IND700 terminal exchange weight, target, and tare data in single-precision floating-point format. The IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Standard 754-1985, specifies the format for single-precision floating point numbers. It is a 32-bit number that has a 1-bit sign, an 8-bit signed exponent, and a 23-bit mantissa. The 8-bit signed exponent provides scaling of weight data. The 23-bit mantissa allows representation of 8 million unique counts.

Although the single-precision floating point number provides greater numerical precision and flexibility than integer weight representations, it has limitations. The weight representation may not be exact, particularly for the extended-resolution weight fields for high-precision bases.

There are two data integrity bits that the IND700 uses to maintain data integrity when communicating with the PLC. One bit is in the beginning word of the data; the second is in the ending byte of the data for a scale slot. The PLC program must verify that both data integrity bits have the same polarity for the data in the scale slot to be valid. There is a possibility that the PLC program will see several consecutive invalid reads when the terminal is freely sending weigh updates to the PLC, if the PLC program detects this condition, it should send a new command to the terminal.

The method of handling string and floating point data varies between Allen-Bradley PLC generations.

B.2.1. Notes: Floating Point Numbers in Various PLCs

The Simatic TI505 PLCs support the IEEE Standard floating point numbers. According to the **Simatic TI505 Programming Reference Manual**, real numbers are stored in the single-precision 32-bit format, according to ANSI/IEEE Standard 754-1985, in the range 5.42101070 E-20 to 9.22337177 E18.

Siemens S5 PLCs do not inherently support the IEEE-format floating point numbers. S5 PLCs do support floating point numbers in their own unique format. S software "function block" can be implemented in the S5 PLC, to convert between S5 floating point numbers and IEEE Standard floating point numbers.

B.3. Floating Point Data Format Definitions

The following tables provide detailed information on the floating-point data format. Read data refers to the PLC's input data and write data refers to the PLC's output data.

Table B-1: Discrete Read Floating Point – IND700 > PLC Input, per Message Slot

Bit number	1st Word Command Response	2nd Word FP value	3rd Word FP value	4th Word Scale Status
0	RESERVED	See Note 4	See Note 4	Target 1[5]
1				Comparator 1[6]
2				Target 2[5]
3				Comparator 2[6]
4				Target 3[5]
5				Always = 1

Bit number	1st Word Command Response	2nd Word FP value	3rd Word FP value	4th Word Scale Status
6				TE bit 1[7]
7				TE bit 2[7]
8	FP Input Indicator 1[1]			Enter Key[8]
9	FP Input Indicator 2[1]			Input 1[9]
10	FP Input Indicator 3[1]			Input 2[9]
11	FP Input Indicator 4[1]			Input 3[9]
12	FP Input Indicator 5[1]			Motion[10]
13	Data integrity1[2]			Net Mode[11]
14	Command Ack 1[3]			Data Integrity 2[2]
15	Command Ack 2[3]			Data OK[12]

Notes for Table B-1

1 The Floating Point Indicator bits (1st word bits 8-12) are used to determine what type of floating or other data is being sent in the second and third words. See the Floating Point Indicator Table for the information from these bits in decimal format.

2 The Data Integrity bits (1st word **bit 13** and 4th word **bit 14**) should be used to assure that communication is still valid and that data are valid. Both of these bits are set to '1' for one update from the indicator, then are set to '0' for the next update from the indicator and this change of state is on every update and is constant as long as the communications link is not disrupted.

3 The first word Command Acknowledge bits (**bits 14** and **15**) are used by the indicator to inform the PLC that a new command was received. The decimal values of these bits will rotate sequentially from 1 to 3 as long as a command other than '0' is being sent (3rd output word). The decimal value of these bits will be '0' when the 3rd output word (PLC output command word) is decimal '0'.

4 The second and third words are 32 bit, single precision floating point data. The data may represent the various scale weight data or setup configuration data. The PLC output command word determines what data will be sent.

5 The fourth word, **bits 0, 2** and **4** indicate the state of the Target comparison logic. When in the material transfer mode; **bit 0** is Feed, **bit 2** is Fast Feed and **bit 4** is Tolerance Ok (within range). When in the over/under mode; **bit 0** is Under, **bit 2** is OK and **bit** 4 is Over. An 'ON' condition is indicated by the bit being set to '1'; an 'OFF' condition is indicated by the bit being set to '0'.

6 The fourth word; Comparator bits indicate the state of the associated comparator logic; when the bit is set to '1' the comparator state is 'ON'; when it is set to '0' the comparator state is 'OFF'. The setup on each comparator will determine when the state is 'ON' or 'OFF'.

7 The fourth word; TE **bit 1** is the state of shared data variable ac0101. TE **bit 2** is the state of shared data variable ac0102. A Task Expert (TE) application may use these bits to instruct the PLC to perform a procedure or function.

8 The fourth word **bit 8** is set to '1' when the Enter Key has been pressed on the keypad of the indicator. The bit can be reset to '0' by sending the command 75 (decimal) in the PLC output command word.

9 The fourth word **bits 9, 10**, and **11** indicate the state of the associated hardware input internal to the indicator; these are 0.1.1, 0.1.2 and 0.1.3. When the input is 'ON' the associated bit is set to '1'.

10 The fourth word **bit 12**; The motion bit is set to '1' when the scale is in motion.

11 The fourth word **bit 13**; The net mode bit is set to '1' when scale is in the net mode (a tare has been taken).

12 The fourth word **bit 15**; The data ok bit is set to '1' when the indicator operating conditions are normal, and when in the x10 display mode. Note that when in x10 mode, the data sent is in the higher resolution. The bit is set to '0' during power-up, during indicator setup, and when the scale is over capacity or under zero. The PLC should continuously monitor the data ok bit in the IND700 communication and also any PLC data connection fault bit that exists in the PLC (refer to the PLC manufacturer documentation) to determine the validity of the data in the PLC.

Table B-2: Floating Point Input Indication

Dec	Data	Dec	Data	Dec	Data
0	Gross Weight [1]	11	Low-pass filter frequency	22	Weigh-in +tolerance value [3]
1	Net Weight [1]	12	Notch filter frequency	23	Weigh-in -tolerance value [3]
2	Tare Weight [1]	13	Target value [3]	24	Weigh-out target value [3]
3	Fine Gross Weight [1]	14	+ Tolerance value [3]	25	Weigh-out fine feed value [3]
4	Fine Net Weight [1]	15	Fine feed value [3]	26	Weigh-out spill value [3]
5	Fine Tare Weight [1]	16	- Tolerance value [3]	27	Weigh-out +tolerance value [3]
6	Rate[1]	17	Spill value [3]	28	Weigh-out -tolerance value [3]
7	Custom field #1	18	Primary units, low increment size	29	Last indicator error code
8	Custom field #2 [2]	19	Weigh-in target value [3]	30	Command received successfully, no response
9	Custom field #3	20	Weigh-in fine feed value [3]	31	Invalid Command
10	Custom field #4 [2]	21	Weigh-in spill value [3]		

Notes for Table B-2

1 Data is refreshed on every indicator update

2 Data is ASCII characters and is limited to the first 4 characters

3 Value that is in the Target registers, may not be the active Target value

Table B-3: Discrete Write Floating Point – PLC >> IND700, per Message Slot

Bit Number	1st Word [Scale command]	2nd Word	3rd Word
0	See Note 1	See Notes 2 and 3	
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			

Notes for Table B-3

1. The first word is a 16 bit integer and is used to send commands to the indicator. The commands are used to:
    - instruct the indicator to report a specific type of data in words 2 and 3. Examples are Gross Weight, Net Weight, + Tolerance Value, etc.
    - instruct the indicator to load the floating point data in the second and third words for a tare value, target value; or other value
    - instruct the indicator to turn on internal outputs or perform a functions, such as Clear Tare, Print, Tare, Start Weigh, etc
2. The second and third words represent a 32 bit single precision floating point value that will be used for downloading a tare, target or other value to the indicator.
3. Not all commands require a floating point value in the second and third words.

Table B-4: PLC Output Command Table (Floating Point Only)

Dec	Hex	Command	SDName
0	0	Report next rotation field @ next A/D update [1]	
1	1	Report next rotation field [1,3]	
2	2	Report next rotation field [1,3]	
3	3	Reset (cancel) rotation	
10	A	Report gross weight [2]	
11	B	Report net weight [2]	
12	C	Report tare weight [2]	
13	D	Report fine gross weight [2]	
14	E	Report fine net weight [2]	
15	F	Report fine tare weight [2]	
16	10	Report Rate[2]	
17	11	Report custom float value #1 [2,5]	aj0101
18	12	Report custom string value #2 [2,4,5]	ak0101
19	13	Report low-pass filter frequency [2,5]	
20	14	Report notch filter frequency [2]	
21	15	Report target value [2,5]	
22	16	Report (+) tolerance value [2,5]	
23	17	Report fine feed [2,5]	
24	18	Report (-) tolerance value [2,5]	
25	19	Report spill value[5]	
27	1B	Report custom float value #3 [5]	aj0102
28	1C	Report custom string value #4 [5]	ak0102
30	1E	Report primary units [5]	
40	28	Add gross weight to rotation [7]	
41	29	Add net weight to rotation [7]	
42	2A	Add tare weight to rotation [7]	
43	2B	Add fine gross weight to rotation [7]	
44	2C	Add fine net weight to rotation [7]	
45	2D	Add fine tare weight to rotation [7]	
46	2E	Add rate to rotation [7]	
47	2F	Add custom value #1 to rotation [7]	aj0101
48	30	Add custom value #2 to rotation [7]	ak0101
60	3C	Load programmable tare value [6]	
61	3D	Pushbutton tare command [7]	
62	3E	Clear command [7]	
63	3F	Print command [7]	
64	40	Zero command [7]	
68	44	Trigger 1 command [7]	
69	45	Trigger 2 command [7]	
70	46	Trigger 3 command [7]	
71	47	Trigger 4 command [7]	
72	48	Trigger 5 command [7]	
73	49	Set low-pass filter frequency [6]	
74	4A	Set notch filter frequency [6]	
75	4B	Reset (clear) ENTER key [7]	
80	50	Clear display message [7,8]	
81	51	Display Message 1 [7,8]	
82	52	Display Message 2 [7,8]	
83	53	Display Message 3 [7,8]	
84	54	Display Message 4 [7,8]	
85	55	Display Message 5 [7,8]	

Dec	Hex	Command	SDName
86	56	Display Message 6 [7,8]	
87	57	Display Message 7 [7,8]	
88	58	Disable weight display [7]	
89	59	Enable weight display [7]	
90	5A	Set discrete output 0.1.1 "ON" [7]	di0105
91	5B	Set discrete output 0.1.2 "ON" [7]	di0106
92	5C	Set discrete output 0.1.3 "ON" [7]	di0107
93	5D	Set discrete output 0.1.4 "ON" [7]	di0108
100	64	Set discrete output 0.1.1 "OFF" [7]	di0105
101	65	Set discrete output 0.1.2 "OFF" [7]	di0106
102	66	Set discrete output 0.1.3 "OFF" [7]	di0107
103	67	Set discrete output 0.1.4 "OFF" [7]	di0108
110	6E	Set target value [6,13]	
111	6F	Set target fine feed value [6,13]	
112	70	Set - tolerance value [6,13]	
114	72	Start target comparison [7, 12,13]	
115	73	Abort target comparison [7, 12,13]	
116	74	Target use gross weight [7]	
117	75	Target use net weight [7]	
119	77	Weigh-In Start [7,10]	
120	78	Weigh-Out Start [7,10]	
121	79	Enable target latching [7]	
122	7A	Disable target latching [7]	
123	7B	Reset target latch [7]	
124	7C	Set Spill Value [6,13]	
131	83	Set (+) tolerance value [6,13]	
160	A0	Apply scale setup (reinitialize) [7, 9]	
162	A2	Disable indicator tare (IDNet only) [7]	
163	A3	Enable indicator tare (IDNet only) [7]	
170	AA	Set weigh-in target value [6,10,11]	af0161
171	AB	Set weigh-in fine feed value [6,10,11]	af0163
172	AC	Set weigh-in spill value [6,10, 11]	af0162
173	AD	Set weigh-in +tolerance value [6,10,11]	af0164
174	AE	Set weigh-in -tolerance value [6,10,11]	af0165
175	AF	Set weigh-out target value [6,10,11]	af0151
176	B0	Set weigh-out fine feed value [6,10,11]	af0153
177	B1	Set weigh-out spill value [6,10,11]	af0152
178	B2	Set weigh-out +tolerance value [6,10,11]	af0154
179	B3	Set weigh-out -tolerance value [6,10,11]	af0155
180	B4	Report weigh-in target value [6,10]	
181	B5	Report weigh-in fine feed value [6,10]	
182	B6	Report weigh-in spill value [6,10]	
183	B7	Report weigh-in +tolerance value [6,10]	
184	B8	Report weigh-in -tolerance value [6,10]	
185	B9	Report weigh-out target value [6,10]	
186	BA	Report weigh-out fine feed value [6,10]	
187	BB	Report weigh-out spill value [6,10]	
188	BC	Report weigh-out +tolerance value [6,10]	
189	BD	Report weigh-out -tolerance value [6,10]	
190	BE	Not used	
191	BF	Not used	
192	C0	Trigger OK key [7]	ac0109
193	C1	Trigger ENTER key [7]	xc0130
194	C2	Trigger weigh-in pause [7,10]	

Dec	Hex	Command	SDName
195	C3	Trigger weigh-in resume	ac0101
196	C4	Trigger weigh-in abort [7,10]	
197	C5	Trigger weigh-out pause [7,10]	
198	C6	Trigger weigh-out resume [7,10]	ac0102
199	C7	Trigger weigh-out abort [7,10]	
210	D2	Set Comparator 1 limit	
211	D3	Set Comparator 1 high limit	
212	D4	Set Comparator 2 limit	
213	D5	Set Comparator 2 high limit	
214	D6	Set Comparator 3 limit	
215	D7	Set Comparator 3 high limit	
216	D8	Set Comparator 4 limit	
217	D9	Set Comparator 4 high limit	
218	DA	Set Comparator 5 limit	
219	DB	Set Comparator 5 high limit	
220	DC	Disable Keypad	
221	DD	Enable Keypad	
222	DE	Report Comparator 1 limit	
223	DF	Report Comparator 1 high limit	
224	E0	Report Comparator 2 limit	
225	E1	Report Comparator 2 high limit	
226	E2	Report Comparator 3 limit	
227	E3	Report Comparator 3 high limit	
228	E4	Report Comparator 4 limit	
229	E5	Report Comparator 4 high limit	
230	E6	Report Comparator 5 limit	
231	E7	Report Comparator 5 high limit	

Dec	Hex	Command	SDName
232	E8	Apply Comparator Values	
233		Set weigh-in +tolerance value for tolerance type "% of target"[10]	af0166
234		Set weigh-in -tolerance value for tolerance type "% of target"[10]	af0167
235		Set weigh-out +tolerance value for tolerance type "% of target"[10]	af0156
236		Set weigh-out -tolerance value for tolerance type "% of target"[10]	af0157
237		Report weigh-in +tolerance value for tolerance type "% of target"[10]	
238		Report weigh-in -tolerance value for tolerance type "% of target"[10]	
239		Report weigh-out +tolerance value for tolerance type "% of target"[10]	
240		Report weigh-out -tolerance value for tolerance type "% of target"[10]	

Notes for Table B-4

1 Rotation is set up by commands 40 to 48 (dec). On each indicator update the next field of the rotation setup is reported in the second and third words of the floating point output from the indicator. The floating point indication date reports what the field data represents. To keep up with the rotation changes, the PLC program scan time should be 30 milliseconds or less. A command of '0' without rotation setup will report the scale gross weight. The commands acknowledge bits are set to the value of '0'.

2 A command that requests data that is refreshed on every indicator update.

3 Toggling between commands 1 and 2 will allow the PLC to control the rotation field change.

4 Only 4 characters of a string field are reported; the PLC must process the data as a string value.

5 A command that request a specific value; as long as the request is in the command word to the indicator no other data will be reported by the indicator.

6 A command that requires a floating point value be in the second and third word when the command is sent to the indicator. If the command is successful the returned floating point value will equal the value sent to the indicator.

7 A command that will not report back a value; the floating point data from the indicator will be zero.

8 The message display commands will cause messages to be displayed on the indicator display above the soft key prompts; this is limited to 20 characters. The message display commands cause a value to be written to shared data PD0119; PD0119 values can be use by Task Expert applications. The command 81 to 87 (dec) will initiate the message events. Command 81will cause the characters in shared data AW0101 to be displayed and PD0119 will be set to '1'. Command 82 = display AW0102 and PD0119 = '2'. Command 83' = display AW0103 and PD0119 = '3'. Command 84 display AW0104 and PD0119 = '4'.

Command 85 = display AW0105 and PD0119 = '5'. Command 86 = start Prompt sequence, PD0119 = '6' and XC0134 = '1'. Command = display PD0118 and PD0119 = '7'. Command 80 (dec) will remove the message display.

9 If shared data classes pl, ds, ll, nt, ce, zr, ct, cm, xs, cs, dp, wk, ao, rp, or dc are changed by the PLC this command (160 dec) will trigger the changes into effect. Shared data is not available with the AB-RIO, DeviceNet and Modbus TCP.

10 A command that can only be used with the IND700 Fill. When Fill-700 is installed, the following commands for standard target control cannot be used: 110-115, 124, 131

11 If Fill-700 is not installed in the terminal, this command can be used to access the corresponding Shared Data field incorporated into a custom TaskExpert program.

12 In the basic terminal (without Fill-700 installed), target control can be paused and resumed using the Abort and Start commands. Note that any changes made to the target values since the original START command was given will be loaded before target control is resumed.

13 This command does not function when Fill-700 is installed. Commands with a "10" footnote should be used for Fill-700 target control.

B.4. Floating Point Command Examples

Table B-5: Data Requirement: Only Net Weight Sent (continuously) for Scale 1

Step #	Scale Command (From PLC)	Scale Floating Point Value	Command Response From Terminal	Floating Point Value
1 (PLC sends command to IND700 terminal to report net weight)	11 (dec) loaded into command word 0	none required		
2 (IND700 terminal sees new command)			Command ack. =1 F.P. ind. = 1 (net)	Net weight in floating point

As long as the PLC leaves the 11 (dec) in the command word, the IND700 terminal will update the net value every interface update cycle.

Table B-6: Data Requirement: Load Target Value = 21.75 for Scale 1

Step #	Scale command (from PLC)	Scale Floating Point Value	Command response from terminal	Floating Point Value
1 (PLC loads floating point value first)		floating point value = 21.75		
2 (PLC sends command to set target 1 cutoff value)	110 (dec) loaded into command word 0	floating point value = 21.75		
3 (IND700 terminal sees new command, loads the value into the target and ends a return message to indicate the new target value)			Command ack. = 1 F.P. ind = 30	Floating point value = 21.75
4 (PLC instructs IND700 terminal to start "using" new target value)	114 (dec) loaded into command word 0			
5 (IND700 terminal sees new command)			Command ack. = 2 F.P. ind = 30	0.0

The PLC should always wait to receive a command acknowledgment before sending the next command to the IND700 terminal. After the PLC finishes loading its target value, it can resume monitoring the weight information required by sending a command to report some type of weight or set up a rotation of reported data.

Table B-7: Data Requirement: Rotation of Gross Weight and Rate Updated on Interface Update Cycle

Step #	Scale Command (from PLC)	Scale Floating Point Value	Command Response from Terminal	Floating Point Value
1 (PLC clears out any previous rotation with reset)	3 (dec) loaded into command word 0			
2 (IND700 terminal sees new command)			Command ack.= 1 F.P. ind = 30	0.0
3 (PLC adds gross weight to rotation)	40 (dec) loaded into command word 0	(null value)		
4 (IND700 terminal sees new command)			Command ack. = 2 F.P. ind = 30	0.0
5 (PLC adds rate to the rotation)	46 (dec) loaded into command word 0			
6 (IND700 terminal sees new command)			Command ack. = 3 F.P. ind = 30	0.0
At this point, the rotation has been set up. Now the PLC needs to command the IND700 terminal to begin the rotation.				
7 (PLC sends the command to begin the rotation at interface update cycle)	0 (dec) loaded into command word 0			
8 (IND700 terminal sends gross weight at interface update cycle ~ 60 msec)			Command ack. = 0 F.P. ind = 0	Floating point value = gross wt.
9 (PLC leaves 0 in its command word and the IND700 terminal sends the rate value at the next interface update cycle)	0 (dec) loaded into command word 0		Command ack. = 0 F.P. ind = 6	Floating point value = rate
10 (PLC leaves 0 in its command word and IND700 terminal sends the gross value at next interface update cycle)	0 (dec) loaded into command word 0		Command ack. = 0 F.P. ind = 0	Floating point value = gross wt.
11 (PLC leaves 0 in command word and IND700 terminal sends the rate value at the next interface update cycle)	0 (dec) loaded into command word 0	RESERVED for Future Use	Command ack. = 0 F.P. ind = 6	Floating point value = rate

This rotation continues until the PLC sends a different command. At approximately every 60 msec the IND700 terminal updates its data with the next field in its rotation. The PLC must check the floating point indication bits to determine which data is in the floating point value.

Table B-8: Data Requirement: Rotation of Net Weight and Rate Updated on PLC Command

Step #	Scale command (from PLC)	Scale Floating Point Value	Command response from terminal	Floating Point Value
1 (PLC clears out any previous rotation with reset)	3 (dec) loaded into command word 0			
2 (IND700 terminal sees new command)			Command ack.= 1 F.P. ind = 30	0.0
3 (PLC adds net weight to rotation)	41 (dec) loaded into command word 0	(null value)		
4 (IND700 terminal sees new command)			Command ack. = 2 F.P. ind = 30	0.0
5 (PLC adds rate to the rotation)	46 (dec) loaded into command word 0	RESERVED for Future Use		
6 (IND700 terminal sees new command)			Command ack. = 3 F.P. ind = 30	0.0
At this point, the rotation has been set up. Now the PLC needs to send commands to the IND700 terminal to begin the rotation and advance to the next value when required.				
7 (PLC sends the command to report the first field in the rotation.)	1 (dec) loaded into command word 0			
8 (IND700 terminal acknowledges the command and sends net weight at every interface update cycle until the PLC gives the command to report the next rotation field.)			Command ack. = 1 F.P. ind = 1	Floating point value = net weight
9 (PLC sends the command to report the next field.) Note: if the PLC leaves the 1 (dec) in the command, the IND700 terminal does NOT see this as another command to report the next rotation field.	2 (dec) loaded into command word 0			
10 (IND700 terminal acknowledges the command and sends rate at every interface update cycle until the PLC gives the command to report the next rotation field.)		RESERVED for Future Use	Command ack. = 2 F.P. ind = 6	Floating point value = rate

Step #	Scale command (from PLC)	Scale Floating Point Value	Command response from terminal	Floating Point Value
11 (PLC sends the command to report the next field in the rotation.)	1 (dec) loaded into command word 0			
12 (IND700 terminal acknowledges the command and sends net weight at every interface update cycle until the PLC gives the command to report the next rotation field.)			Command ack. = 1 F.P. ind = 1	Floating point value = net wt.
13 (PLC sends the command to report the next field.)	2 (dec) loaded into command word 0			
14 (IND700 terminal acknowledges the command and sends rate at every interface update cycle until the PLC gives the command to report the next rotation field.)		RESERVED for Future Use	Command ack. = 2 F.P. ind = 6	Floating point value = rate

At approximately every 60 msec the IND700 terminal updates its data with new data, but it does not advance to the next field in the rotation until the PLC sends it the command to report the next field. The PLC should check the floating point indication bits to determine which data is in the floating point value

C. Common Data Features

C.1. Data Formats

C.1.1. Discrete Data

Three data formats are available: Integer (the default), Divisions and Floating Point.

Integer Reports scale weight as a signed 16 bit integer (± 32767).

Divisions Reports scale weight in display divisions (± 32767). The PLC multiplies the reported divisions by the increment size to calculate the weight in display units.

Floating Point Displays weight in floating point data format

The data format of discrete data will affect the data size required in the configuration of the PLC. The IND700 console PLC message slot setup screen provides data size requirements in bytes.

Selection of the appropriate format depends on issues such as the range or capacity of the scale used in the application. The integer format can represent a numerical value up to 32,767. The division format can represent a value up to 32,767 scale divisions or increments. The floating-point format can represent a value encoded in IEEE 754, single precision floating point format.

Floating point is the only data format that includes decimal point information. Integer and division formats ignore decimal points. Accommodation of decimal point location must occur in the PLC logic, when it is needed with these formats.

C.1.1.1. Examples

250 x .01 scale					**50,000 x 10 scale**				
IND700 Displays:	0	2.00	51.67	250.00	IND700 Displays:	0	200	5160	50000
Format sent:					Format sent:				
Integer	0	200	5167	25000	Integer	0	200	5160	–(xxxxx)
Division	0	200	5167	25000	Division	0	20	516	5000
Floating Point	0	2.00	51.67	250.00	Floating Point	0	200	5160	50000
Any of the formats could be used in this case.					The integer format could not be used because it would send a negative or invalid value once the weight exceeded 32,760.				

150 x .001 scale				
IND700 Displays:	0	2.100	51.607	150.000
Format sent:				
Integer	0	2100	–(xxxxx)	–(xxxxx)
Division	0	2100	–(xxxxx)	–(xxxxx)
Floating Point	0	2.100	51.607	150.000

The integer and division formats could not be used because they would send a negative value once the weight exceeded 32.767.

Please refer to Appendix A and Appendix B for each format's detailed description of data available to determine which is most suitable.

C.2. Byte Order

The byte order parameter sets the order in which the data bytes and words will be presented in the PLC data format. Available Byte Order selections are:

Word Swap	Makes the data format compatible with RSLogix 5000 processors.
Byte Swap	Makes the data format compatible with S7 Profibus.
Standard	Makes the data format compatible with PLC 5
Double Word Swap	Makes the data format compatible with the Modicon Quantum PLC for Modbus TCP networks.

Table C-1 provides examples of the various byte ordering.

Table C-1: PLC Data Byte Ordering

		Word Swap	Byte Swap	Double Word Swap	Standard
Terminal Weight Value		1355			
	PLC	**15 Bit # 0**	**15 Bit # 0**	**15 Bit # 0**	**15 Bit # 0**
Integer	Weight value word	0x054B Hex	0x4B05 Hex	0x4B05 Hex	0x054B Hex
Floating Point	1st Weight value word	0x6000 Hex	0xA944 Hex	0x0060 Hex	0x44A9 Hex
	2nd Weight value word	0x44A9 Hex	0x0060 Hex	0xA944 Hex	0x6000 Hex

Please refer to Appendix A and Appendix B for each format's detailed description of data available to determine which is most suitable.

C.3. Controlling Discrete I/O Using a PLC Interface

The IND700 terminal provides the ability to directly control some of its discrete outputs and read some of its discrete inputs via the (digital) PLC interface options. System integrators should be aware that the terminal's discrete I/O updates are synchronized with the terminal's A/D rate and not with the PLC I/O scan rate. This may cause a noticeable delay in reading inputs or updating outputs as observed from the PLC to real world signals.

Consult the **IND700 Terminal Installation Manual** for discrete I/O wiring. Also note that the outputs must be unassigned in the IND700 terminal at **Setup > Application > Discrete I/O** in order to be controlled by the PLC.

To protect your product's future:

METTLER TOLEDO Service assures the quality, measuring accuracy and preservation of value of this product for years to come.

Please request full details about our attractive terms of service.

www.mt.com/service

www.mt.com

For more information

Mettler-Toledo, (Changzhou) Measurement Technology Co., Ltd
111 Taihu West Road
Xinbei District
Changzhou, Jiangsu
China, 213125
www.mt.com/contacts

30753886 Rev. A, 03/2024

30753886