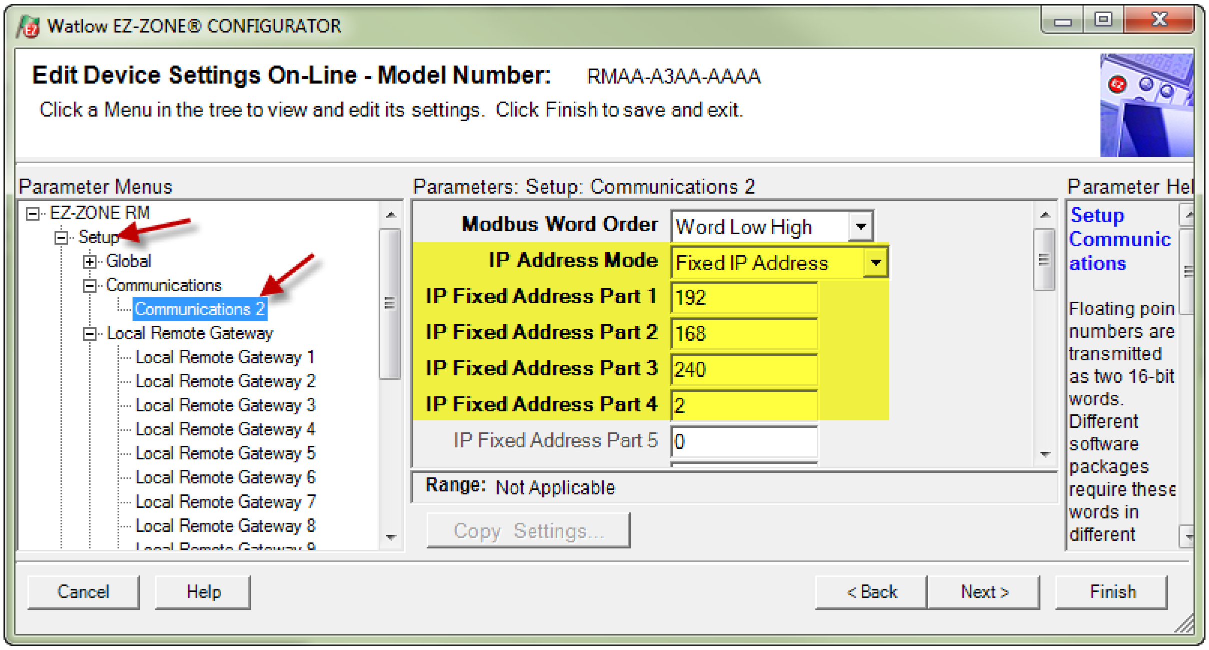
Setting Implicit Message Assembly in EZ-ZONE®

Implicit messaging is the concept of automatic block copying of memory locations between a PLC and another device. The other connected device (EZ-ZONE) will block copy a group of memory locations to the PLC with data values. The PLC will block copy a group of memory locations to the connected device with data values. Once configured and connected, this process is automatic. The data values copied depend on the configuration of the Implicit Assembly. The user must configure both the EZ-ZONE and the PLC to agree on the data pointers and data types as well as the quantities of data values transferred. Implicit means that the values sent between devices are implicitly known so nothing is sent with the data values to indicate what they represent. The maximum number of data values transferred in each direction is 20 or 40 per EZ-ZONE product (model dependent) and 100 (Ethernet) or 200 (DeviceNet) per EZ-ZONE system. An EZ-ZONE system comprises multiple EZ-ZONE products communicating through a RUI/Gateway or Access Module.

The following tasks are required to implement an EZ-ZONE product with a PLC using implicit messaging:

1. Determine what data is to be transferred from the PLC to the EZ-ZONE controller.
2. Determine what data is to be transferred from the EZ-ZONE controller to the PLC.
3. Locate the Parameter ID in the user's manual for each data point to be transferred.
4. Document your choices. *The Excel spreadsheet called "EZ-ZONE Default Implicit Assembly.xlsb" supplied by Watlow may be used to accomplish this task.*
5. Populate the assembly in the EZ-ZONE controller using the procedure detailed later in this document.
6. Determine the total number of values to be sent in each direction for each EZ-ZONE controller.
7. Program the communication's settings using EZ-ZONE Configurator or using the keypad. *Program IP Address Mode and IP address with Subnet Mask. The power must be cycled on the EZ-ZONE product for the new IP address or Node address to take effect.*



Edit Device Settings On-Line - Model Number: RMAA-A3AA-AAAA

Click a Menu in the tree to view and edit its settings. Click Finish to save and exit.

Parameters: Setup: Communications 2

Modbus Word Order	Word Low High
IP Address Mode	Fixed IP Address
IP Fixed Address Part 1	192
IP Fixed Address Part 2	168
IP Fixed Address Part 3	240
IP Fixed Address Part 4	2
IP Fixed Address Part 5	0

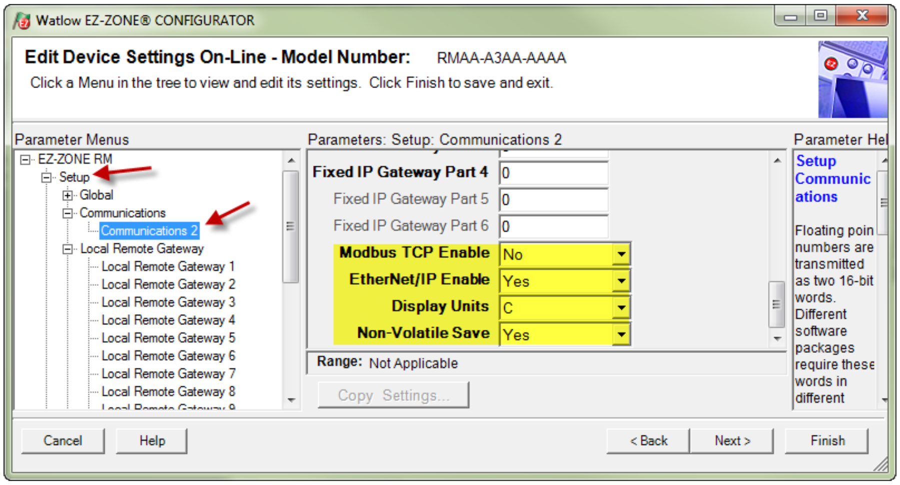
Range: Not Applicable

Parameter Hel

Setup Communications

Floating poin numbers are transmitted as two 16-bit words. Different software packages require these words in different

8. Disable Modbus TCP, Enable EtherNet/IP, set Display Comms units for connection and set Non-Volatile Save to 'Yes'. *Setting protocol enable/disable is not required for DeviceNet versions. Setting Display units to degrees F or C affects units sent via communications. This is independent of units displayed on the controller's display. The controller's display is affected by the setting in the Setup Page, Global Menu.*



Watlow EZ-ZONE® CONFIGURATOR

Edit Device Settings On-Line - Model Number: RMAA-A3AA-AAAA

Click a Menu in the tree to view and edit its settings. Click Finish to save and exit.

Parameters: Setup: Communications 2

Fixed IP Gateway Part 4	0
Fixed IP Gateway Part 5	0
Fixed IP Gateway Part 6	0
Modbus TCP Enable	No
EtherNet/IP Enable	Yes
Display Units	C
Non-Volatile Save	Yes

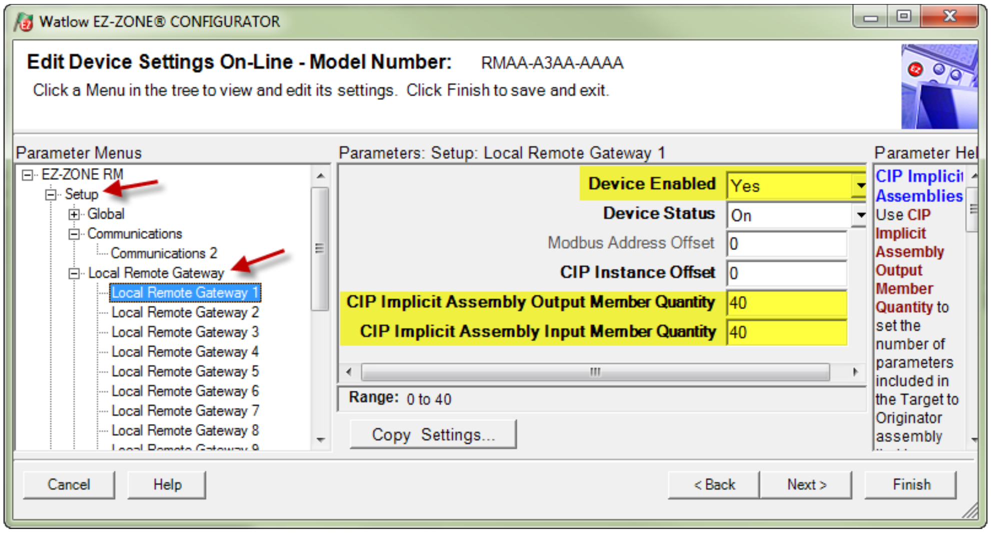
Range: Not Applicable

Parameter Hel

Setup Communic ations

Floating poin numbers are transmitted as two 16-bit words. Different software packages require these words in different

9. Set Device Enabled to 'Yes' for each Local Remote Gateway being used. Ensure all others are set to 'No'. Program each EZ-ZONE controller, RUI/Gateway or Access module for the number of values to be sent via comms in each direction. *When the RUI/Gateway or Access module is used as the communication's connection, the settings are located in the Local Remote Gateway menu of the Setup Page for that device. CIP Implicit Assembly Output Member Quantity refers to the number of values sent from the EZ-ZONE controller to the PLC's Assembly Input. CIP Implicit Assembly Input Member Quantity refers to the number of values sent from the PLC's Assembly Output to the EZ-ZONE controller's Assembly Input. Local Remote Gateway 1 refers to the controller connected as Standard Bus Address 1 where as Local Remote Gateway 2 refers to the controller connected as Standard Bus Address 2 and so on…*



Watlow EZ-ZONE® CONFIGURATOR

Edit Device Settings On-Line - Model Number: RMAA-A3AA-AAAA

Click a Menu in the tree to view and edit its settings. Click Finish to save and exit.

Parameters: Setup: Local Remote Gateway 1

Device Enabled	Yes
Device Status	On
Modbus Address Offset	0
CIP Instance Offset	0
CIP Implicit Assembly Output Member Quantity	40
CIP Implicit Assembly Input Member Quantity	40

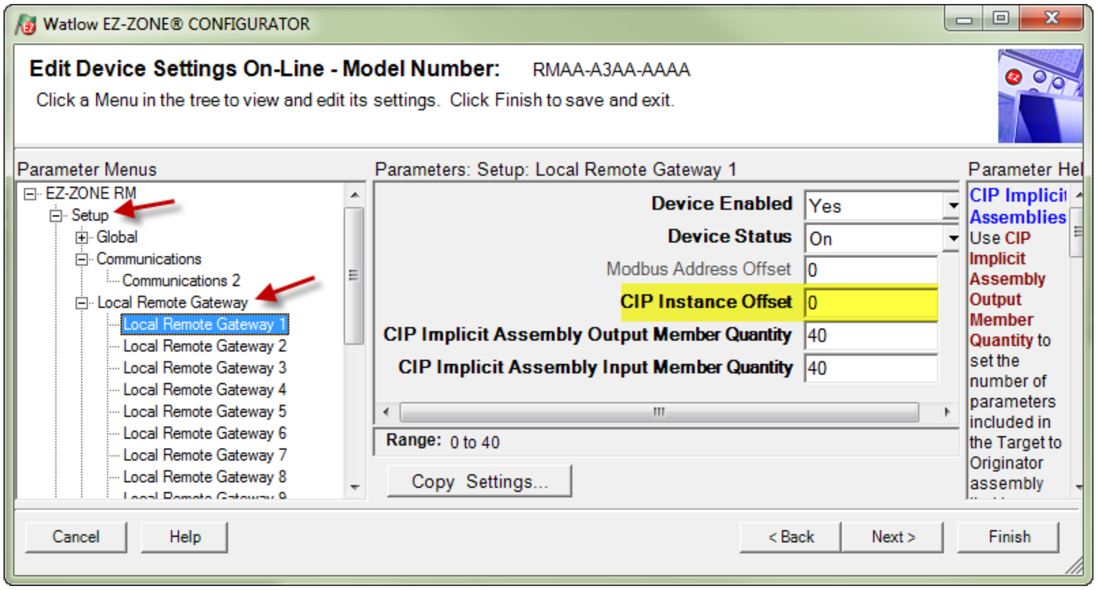
Range: 0 to 40

Parameter Hel

CIP Implicit Assemblies

Use CIP Implicit Assembly Output Member Quantity to set the number of parameters included in the Target to Originator assembly

10. Program the CIP Instance Offset for each Local Remote Gateway if explicit messaging is to be utilized. *We suggest 0 for Local Remote Gateway 1; 40 for Local Remote Gateway 2; 80 for Local Remote Gateway 3 and so on (assuming 40 members per module).*



Watlow EZ-ZONE® CONFIGURATOR

Edit Device Settings On-Line - Model Number: RMAA-A3AA-AAAA

Click a Menu in the tree to view and edit its settings. Click Finish to save and exit.

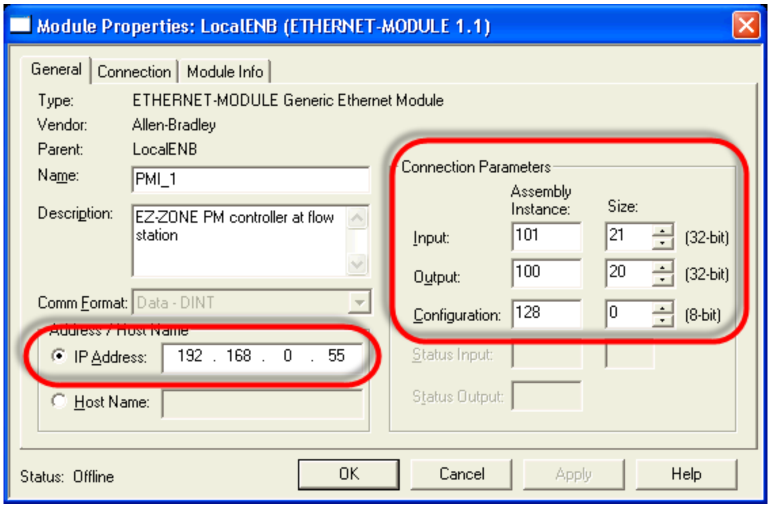
Parameters: Setup: Local Remote Gateway 1

- **Device Enabled**: Yes
- **Device Status**: On
- Modbus Address Offset: 0
- **CIP Instance Offset**: 0
- **CIP Implicit Assembly Output Member Quantity**: 40
- **CIP Implicit Assembly Input Member Quantity**: 40

Range: 0 to 40

Parameter Help: **CIP Implicit Assemblies** Use CIP Implicit Assembly Output Member Quantity to set the number of parameters included in the Target to Originator assembly

11. Specify the IP address or DeviceNet address in the PLC. *Program the PLC implicit assembly in the configuration tool used for your PLC using a Generic Ethernet Module for Ethernet or Generic DeviceNet Module for DeviceNet equipped models.*



Module Properties: LocalENB (ETHERNET-MODULE 1.1)

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: LocalENB
Name: PMI_1
Description: EZ-ZONE PM controller at flow station
Comm Format: Data - DINT

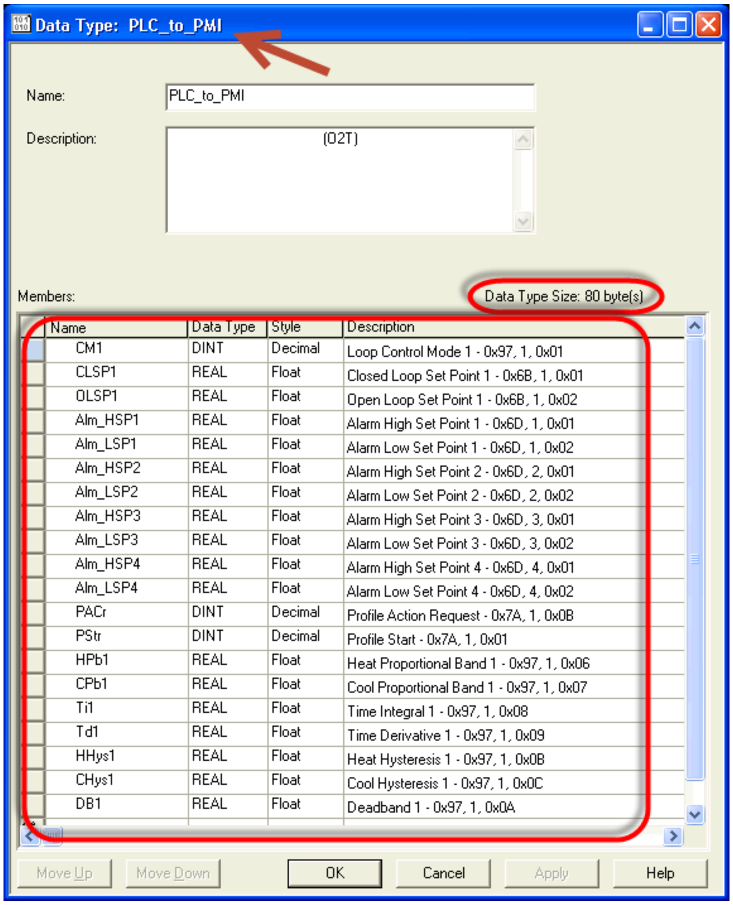
Address / Host Name
● IP Address: 192 . 168 . 0 . 55
○ Host Name:

Connection Parameters

	Assembly Instance:	Size:	
Input:	101	21	(32-bit)
Output:	100	20	(32-bit)
Configuration:	128	0	(8-bit)
Status Input:			
Status Output:			

Status: Offline

12. Define the tags as selected in the Implicit Assembly of the controllers. *The RMA or RUI/Gateway will send one status word of 32-bits for a system of controllers. Therefore the PLC Input size will be one more than the total selected parameters. The Input and Output size must be correct for the connection to be valid. The input size is the sum of all EZ-ZONE outputs for all Local Remote Gateways enabled plus 1 and the output size is the sum of all EZ-ZONE inputs for all Local Remote Gateways enabled. If there is no Local Remote Gateway as in an EZ-ZONE PM Integrated with Ethernet on board, it is the total CIP Implicit Assembly Input and Output Member Quantity specified in the Setup Page, Communications 2 Menu. Configuration Assembly Instance is 128 and Size is always zero for the PLC setup.*



Name: PLC_to_PMI

Description: (O2T)

Members: Data Type Size: 80 byte(s)

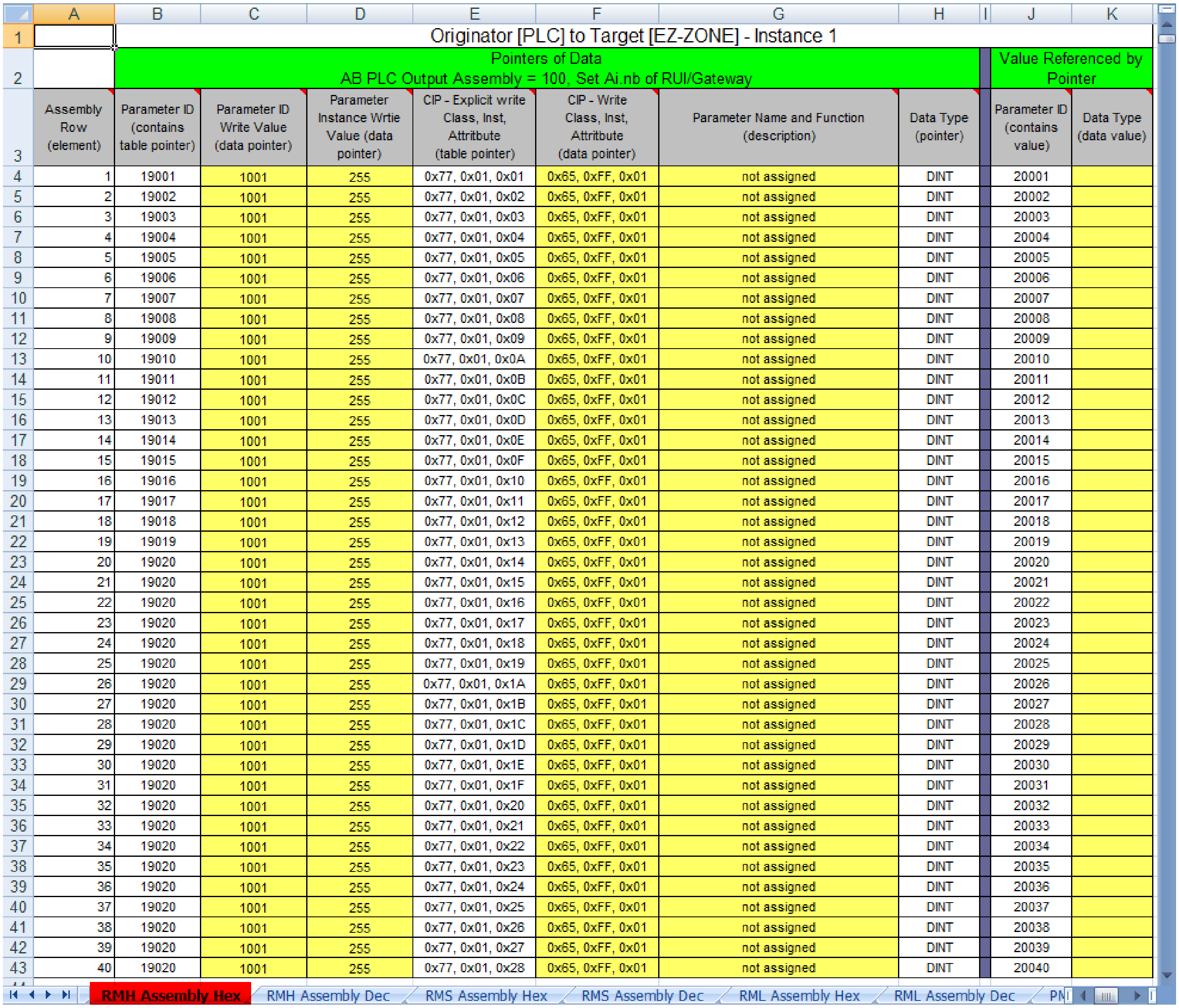
Name	Data Type	Style	Description
CM1	DINT	Decimal	Loop Control Mode 1 - 0x97, 1, 0x01
CLSP1	REAL	Float	Closed Loop Set Point 1 - 0x6B, 1, 0x01
OLSP1	REAL	Float	Open Loop Set Point 1 - 0x6B, 1, 0x02
Alm_HSP1	REAL	Float	Alarm High Set Point 1 - 0x6D, 1, 0x01
Alm_LSP1	REAL	Float	Alarm Low Set Point 1 - 0x6D, 1, 0x02
Alm_HSP2	REAL	Float	Alarm High Set Point 2 - 0x6D, 2, 0x01
Alm_LSP2	REAL	Float	Alarm Low Set Point 2 - 0x6D, 2, 0x02
Alm_HSP3	REAL	Float	Alarm High Set Point 3 - 0x6D, 3, 0x01
Alm_LSP3	REAL	Float	Alarm Low Set Point 3 - 0x6D, 3, 0x02
Alm_HSP4	REAL	Float	Alarm High Set Point 4 - 0x6D, 4, 0x01
Alm_LSP4	REAL	Float	Alarm Low Set Point 4 - 0x6D, 4, 0x02
PACr	DINT	Decimal	Profile Action Request - 0x7A, 1, 0x0B
PStr	DINT	Decimal	Profile Start - 0x7A, 1, 0x01
HPb1	REAL	Float	Heat Proportional Band 1 - 0x97, 1, 0x06
CPb1	REAL	Float	Cool Proportional Band 1 - 0x97, 1, 0x07
Ti1	REAL	Float	Time Integral 1 - 0x97, 1, 0x08
Td1	REAL	Float	Time Derivative 1 - 0x97, 1, 0x09
HHys1	REAL	Float	Heat Hysteresis 1 - 0x97, 1, 0x0B
CHys1	REAL	Float	Cool Hysteresis 1 - 0x97, 1, 0x0C
DB1	REAL	Float	Deadband 1 - 0x97, 1, 0x0A

13. Connect the PLC to the controller via EtherNet/IP rated switch or through DeviceNet as applicable. For DeviceNet equipped devices, set the Node address and baud rate.
14. The controller and PLC should now exchange data.

This completes the basic steps required to connect the EZ-ZONE product to a PLC for Implicit Messaging. The following procedure documents changing the Implicit Assembly data pointers to match your requirements using a software program called EZ-ZONE Assembly Programmer available from Watlow.

The Implicit Assembly data pointers may also be changed using explicit messaging but this method will not be utilized in this procedure.

1. Open the Excel file called "EZ-ZONE Default Implicit Assemblies.xlsb". Resave and re-name this file for your application.
2. Go to the tab in this file using either Decimal or Hexadecimal format for the module or controller type utilized in your application. In this example we are using an RMH module and working in hexadecimal.



Originator [PLC] to Target [EZ-ZONE] - Instance 1

Pointers of Data
AB PLC Output Assembly = 100, Set Ai.nb of RUI/Gateway

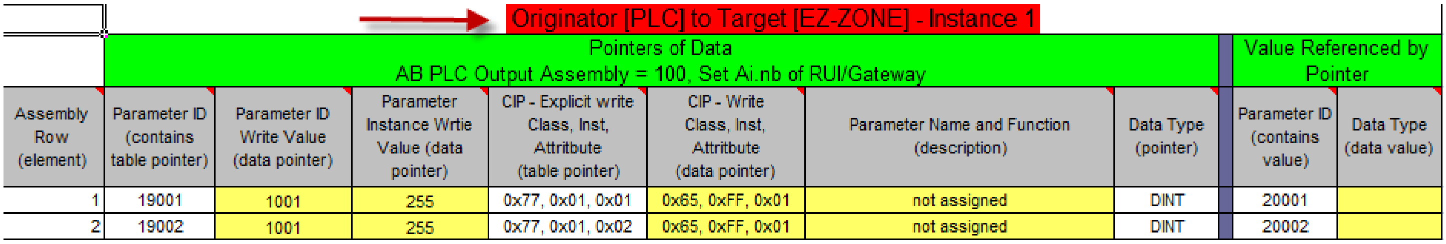
Value Referenced by Pointer

	A	B	C	D	E	F	G	H	J	K
3	Assembly Row (element)	Parameter ID (contains table pointer)	Parameter ID Write Value (data pointer)	Parameter Instance Wrtie Value (data pointer)	CIP - Explicit write Class, Inst, Attritbute (table pointer)	CIP - Write Class, Inst, Attritbute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)	Parameter ID (contains value)	Data Type (data value)
4	1	19001	1001	255	0x77, 0x01, 0x01	0x65, 0xFF, 0x01	not assigned	DINT	20001	
5	2	19002	1001	255	0x77, 0x01, 0x02	0x65, 0xFF, 0x01	not assigned	DINT	20002	
6	3	19003	1001	255	0x77, 0x01, 0x03	0x65, 0xFF, 0x01	not assigned	DINT	20003	
7	4	19004	1001	255	0x77, 0x01, 0x04	0x65, 0xFF, 0x01	not assigned	DINT	20004	
8	5	19005	1001	255	0x77, 0x01, 0x05	0x65, 0xFF, 0x01	not assigned	DINT	20005	
9	6	19006	1001	255	0x77, 0x01, 0x06	0x65, 0xFF, 0x01	not assigned	DINT	20006	
10	7	19007	1001	255	0x77, 0x01, 0x07	0x65, 0xFF, 0x01	not assigned	DINT	20007	
11	8	19008	1001	255	0x77, 0x01, 0x08	0x65, 0xFF, 0x01	not assigned	DINT	20008	
12	9	19009	1001	255	0x77, 0x01, 0x09	0x65, 0xFF, 0x01	not assigned	DINT	20009	
13	10	19010	1001	255	0x77, 0x01, 0x0A	0x65, 0xFF, 0x01	not assigned	DINT	20010	
14	11	19011	1001	255	0x77, 0x01, 0x0B	0x65, 0xFF, 0x01	not assigned	DINT	20011	
15	12	19012	1001	255	0x77, 0x01, 0x0C	0x65, 0xFF, 0x01	not assigned	DINT	20012	
16	13	19013	1001	255	0x77, 0x01, 0x0D	0x65, 0xFF, 0x01	not assigned	DINT	20013	
17	14	19014	1001	255	0x77, 0x01, 0x0E	0x65, 0xFF, 0x01	not assigned	DINT	20014	
18	15	19015	1001	255	0x77, 0x01, 0x0F	0x65, 0xFF, 0x01	not assigned	DINT	20015	
19	16	19016	1001	255	0x77, 0x01, 0x10	0x65, 0xFF, 0x01	not assigned	DINT	20016	
20	17	19017	1001	255	0x77, 0x01, 0x11	0x65, 0xFF, 0x01	not assigned	DINT	20017	
21	18	19018	1001	255	0x77, 0x01, 0x12	0x65, 0xFF, 0x01	not assigned	DINT	20018	
22	19	19019	1001	255	0x77, 0x01, 0x13	0x65, 0xFF, 0x01	not assigned	DINT	20019	
23	20	19020	1001	255	0x77, 0x01, 0x14	0x65, 0xFF, 0x01	not assigned	DINT	20020	
24	21	19020	1001	255	0x77, 0x01, 0x15	0x65, 0xFF, 0x01	not assigned	DINT	20021	
25	22	19020	1001	255	0x77, 0x01, 0x16	0x65, 0xFF, 0x01	not assigned	DINT	20022	
26	23	19020	1001	255	0x77, 0x01, 0x17	0x65, 0xFF, 0x01	not assigned	DINT	20023	
27	24	19020	1001	255	0x77, 0x01, 0x18	0x65, 0xFF, 0x01	not assigned	DINT	20024	
28	25	19020	1001	255	0x77, 0x01, 0x19	0x65, 0xFF, 0x01	not assigned	DINT	20025	
29	26	19020	1001	255	0x77, 0x01, 0x1A	0x65, 0xFF, 0x01	not assigned	DINT	20026	
30	27	19020	1001	255	0x77, 0x01, 0x1B	0x65, 0xFF, 0x01	not assigned	DINT	20027	
31	28	19020	1001	255	0x77, 0x01, 0x1C	0x65, 0xFF, 0x01	not assigned	DINT	20028	
32	29	19020	1001	255	0x77, 0x01, 0x1D	0x65, 0xFF, 0x01	not assigned	DINT	20029	
33	30	19020	1001	255	0x77, 0x01, 0x1E	0x65, 0xFF, 0x01	not assigned	DINT	20030	
34	31	19020	1001	255	0x77, 0x01, 0x1F	0x65, 0xFF, 0x01	not assigned	DINT	20031	
35	32	19020	1001	255	0x77, 0x01, 0x20	0x65, 0xFF, 0x01	not assigned	DINT	20032	
36	33	19020	1001	255	0x77, 0x01, 0x21	0x65, 0xFF, 0x01	not assigned	DINT	20033	
37	34	19020	1001	255	0x77, 0x01, 0x22	0x65, 0xFF, 0x01	not assigned	DINT	20034	
38	35	19020	1001	255	0x77, 0x01, 0x23	0x65, 0xFF, 0x01	not assigned	DINT	20035	
39	36	19020	1001	255	0x77, 0x01, 0x24	0x65, 0xFF, 0x01	not assigned	DINT	20036	
40	37	19020	1001	255	0x77, 0x01, 0x25	0x65, 0xFF, 0x01	not assigned	DINT	20037	
41	38	19020	1001	255	0x77, 0x01, 0x26	0x65, 0xFF, 0x01	not assigned	DINT	20038	
42	39	19020	1001	255	0x77, 0x01, 0x27	0x65, 0xFF, 0x01	not assigned	DINT	20039	
43	40	19020	1001	255	0x77, 0x01, 0x28	0x65, 0xFF, 0x01	not assigned	DINT	20040	

3. Column A lists the Assembly Row or element number from 1 to 40. The column is used to reference the order in which items are organized in the table.
4. Column B lists the Parameter ID (memory location contains table pointer) that will be written into with the EZ-ZONE Assembly Programmer application. This location will be written into with a data pointer

(another Parameter ID number) that is to be referenced such as Closed Loop 1, Closed Loop Set Point which has a Parameter ID associated.

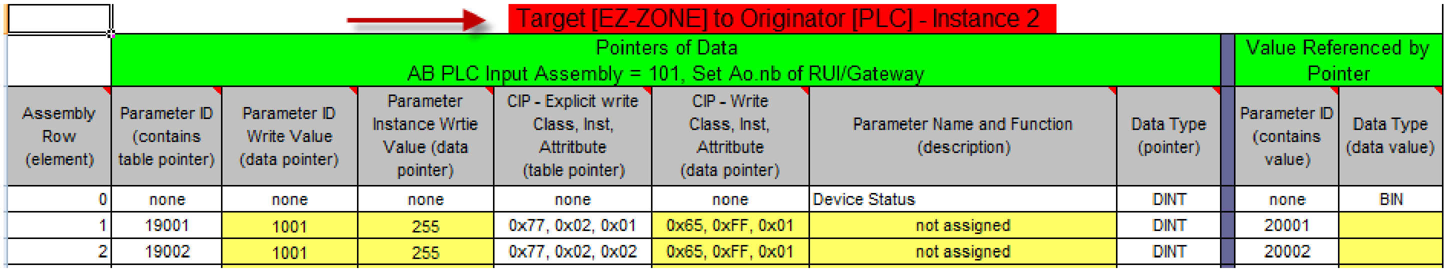
5. Column C lists the Parameter ID (data pointer) that we wish to place into the above location.
6. Column D list the Parameter Instance of the Parameter ID in column C.
7. *Column E lists the CIP (Class, Instance and Attribute) table pointer. This memory location provides another way to write a pointer of the data in column F that is to be referenced such as Closed Loop 1, Closed Loop Set Point which has a CIP associated. This is the method used when the PLC writes an explicit message to change the data pointer. This method will not be used in the document.*
8. *Column F lists the CIP (Class, Instance and Attribute) of the data pointer. This is the method used when the PLC writes an explicit message to change the data pointer. This method will not be used in the document.*
9. Column G provides a location in the spreadsheet to document the Parameter Name and Function. *In other words, a description of the pointer you are populating in the table.*
10. Column H displays that the data type for the pointer is <u>always</u> of DINT (Double Integer containing 32 bits).
11. Column J contains a Parameter ID that is used by the EZ-ZONE Assembly Programmer application if you want to test the results of the pointer you placed in the table. *When you click read on the column to the right of Actual Value that is the location being read.*
12. Finally, column K provides a location in the spreadsheet to document the data type being referenced by the data pointer. *Data types will be 32-bit floating point called real or 32-bit integer called DINT.*
13. There are two sections to the assembly documented in this spreadsheet.
    a. The top section defines Originator [PLC] to Target [EZ-ZONE] – Instance 1. This is information that the PLC writes to the controller on each scan.



Originator [PLC] to Target [EZ-ZONE] - Instance 1

	Pointers of Data AB PLC Output Assembly = 100, Set Ai.nb of RUI/Gateway							Value Referenced by Pointer	
Assembly Row (element)	Parameter ID (contains table pointer)	Parameter ID Write Value (data pointer)	Parameter Instance Wrtie Value (data pointer)	CIP - Explicit write Class, Inst, Attritbute (table pointer)	CIP - Write Class, Inst, Attritbute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)	Parameter ID (contains value)	Data Type (data value)
1	19001	1001	255	0x77, 0x01, 0x01	0x65, 0xFF, 0x01	not assigned	DINT	20001	
2	19002	1001	255	0x77, 0x01, 0x02	0x65, 0xFF, 0x01	not assigned	DINT	20002	

    b. The bottom section defines Target [EZ-ZONE] to Originator [PLC] – Instance 2. This is information that the controller writes to the PLC on each scan.

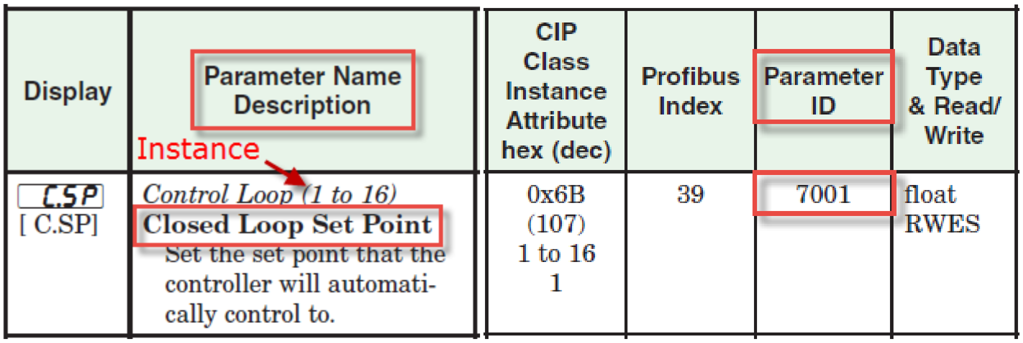


Target [EZ-ZONE] to Originator [PLC] - Instance 2

	Pointers of Data AB PLC Input Assembly = 101, Set Ao.nb of RUI/Gateway							Value Referenced by Pointer	
Assembly Row (element)	Parameter ID (contains table pointer)	Parameter ID Write Value (data pointer)	Parameter Instance Wrtie Value (data pointer)	CIP - Explicit write Class, Inst, Attritbute (table pointer)	CIP - Write Class, Inst, Attritbute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)	Parameter ID (contains value)	Data Type (data value)
0	none	none	none	none	none	Device Status	DINT	none	BIN
1	19001	1001	255	0x77, 0x02, 0x01	0x65, 0xFF, 0x01	not assigned	DINT	20001	
2	19002	1001	255	0x77, 0x02, 0x02	0x65, 0xFF, 0x01	not assigned	DINT	20002	

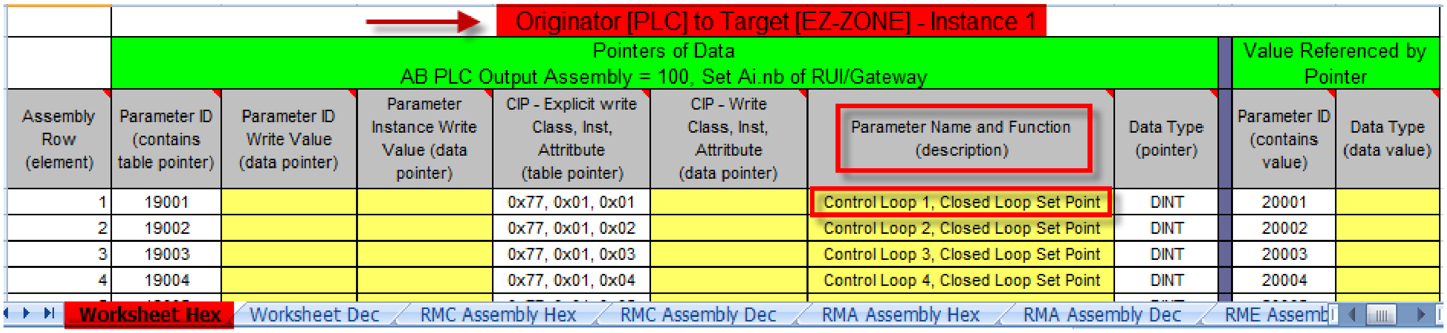
Depending on the model of EZ-ZONE product, there may be 20 or 40 elements in each section. The EZ-ZONE product allows the user to set the number of elements sent in each direction using the Setup Page, Local Remote Gateway Menu. This was discussed earlier in this document.

14. Referring to the User's Manual, identify all parameters that the PLC will write to the controller. *Example: the PLC will write the Control Loop 1, Closed Loop Set Point. The Parameter ID = 7001.*



Display	Parameter Name Description	CIP Class Instance Attribute hex (dec)	Profibus Index	Parameter ID	Data Type & Read/ Write
C.SP [C.SP]	*Control Loop (1 to 16)* **Closed Loop Set Point** Set the set point that the controller will automatically control to.	0x6B (107) 1 to 16 1	39	7001	float RWES

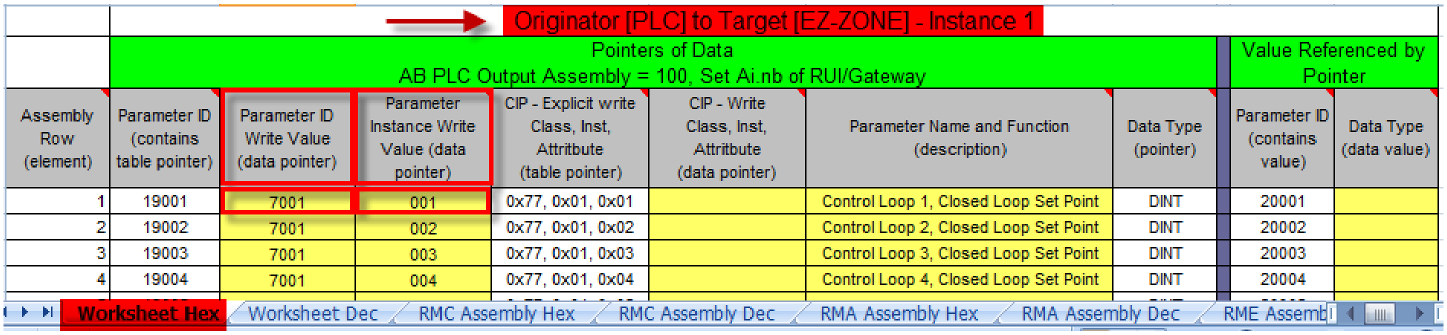
15. Document in the "EZ-ZONE Default Implicit Assemblies.xlsb" file in Worksheet Hex, the Parameter Name and Function starting at Assembly Row 1 each parameter that will be used. *This can be obtained from the User's Manual. This spreadsheet will document one controller or module on the network. Duplicate the Worksheet as often as required and name each sheet. Note to record this in instance 1 called Originator [PLC] to Target [EZ-ZONE] as the closed loop set point is to be written from the PLC.*



Originator [PLC] to Target [EZ-ZONE] - Instance 1

	Pointers of Data AB PLC Output Assembly = 100, Set Ai.nb of RUI/Gateway							Value Referenced by Pointer	
Assembly Row (element)	Parameter ID (contains table pointer)	Parameter ID Write Value (data pointer)	Parameter Instance Write Value (data pointer)	CIP - Explicit write Class, Inst, Attritbute (table pointer)	CIP - Write Class, Inst, Attritbute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)	Parameter ID (contains value)	Data Type (data value)
1	19001			0x77, 0x01, 0x01		Control Loop 1, Closed Loop Set Point	DINT	20001	
2	19002			0x77, 0x01, 0x02		Control Loop 2, Closed Loop Set Point	DINT	20002	
3	19003			0x77, 0x01, 0x03		Control Loop 3, Closed Loop Set Point	DINT	20003	
4	19004			0x77, 0x01, 0x04		Control Loop 4, Closed Loop Set Point	DINT	20004	

16. Therefore document in Worksheet the Parameter ID write Value and Parameter Instance:

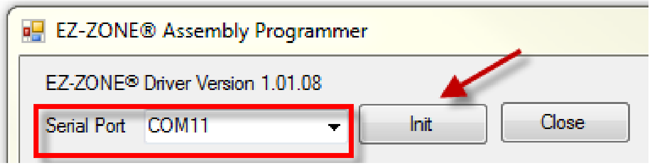


Originator [PLC] to Target [EZ-ZONE] - Instance 1

	Pointers of Data AB PLC Output Assembly = 100, Set Ai.nb of RUI/Gateway							Value Referenced by Pointer	
Assembly Row (element)	Parameter ID (contains table pointer)	Parameter ID Write Value (data pointer)	Parameter Instance Write Value (data pointer)	CIP - Explicit write Class, Inst, Attritbute (table pointer)	CIP - Write Class, Inst, Attritbute (data pointer)	Parameter Name and Function (description)	Data Type (pointer)	Parameter ID (contains value)	Data Type (data value)
1	19001	7001	001	0x77, 0x01, 0x01		Control Loop 1, Closed Loop Set Point	DINT	20001	
2	19002	7001	002	0x77, 0x01, 0x02		Control Loop 2, Closed Loop Set Point	DINT	20002	
3	19003	7001	003	0x77, 0x01, 0x03		Control Loop 3, Closed Loop Set Point	DINT	20003	
4	19004	7001	004	0x77, 0x01, 0x04		Control Loop 4, Closed Loop Set Point	DINT	20004	

17. Continue filling each parameter name, parameter ID and parameter instance for Originator to Target and Target to Originator. *Do this for each PM, ST or RM controller on the network, creating a* *spreadsheet for each. Be sure to label each spreadsheet with which Zone (Standard Bus Address) applies to each.*

18. Connect a PC using EIA-485 to the Standard Bus port of the controller or module. *The terminals are labeled CD, CE and CF on the EZ-ZONE product.*
19. Execute EZ-ZONE Assembly Programmer application.
20. Specify the Serial Port of the PC connected to the EZ-ZONE product. Then Click 'Init' to open the port.

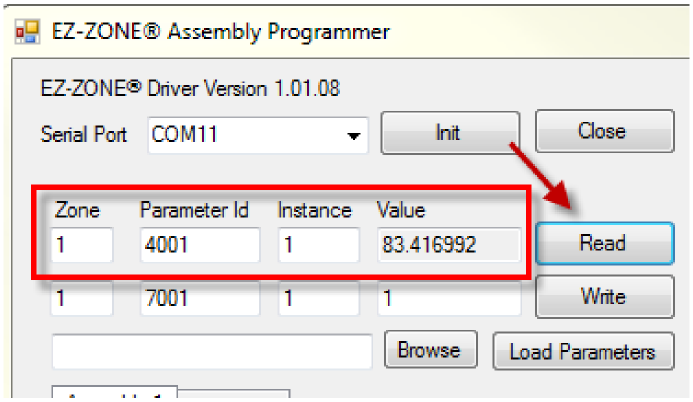


EZ-ZONE® Assembly Programmer

EZ-ZONE® Driver Version 1.01.08

Serial Port COM11

21. Read a Parameter ID to verify communication. Enter the Zone, Parameter Id 4001, instance 1 for analog input 1 and click the 'Read' button. *The example below reads Zone 1, the Parameter ID 4001 which is Analog Input, Instance 1 which shows a value of 83.4 degrees. Zone refers to the Standard Bus address shown on the controller. The captured screen shows where you can either read or write to a Parameter Id. Do not randomly write to Parameter Id as unexpected results will occur. It is important to read a parameter first to ensure you selected the correct one before a write to that location. This step validates that you are communicating correctly to the selected zone.*



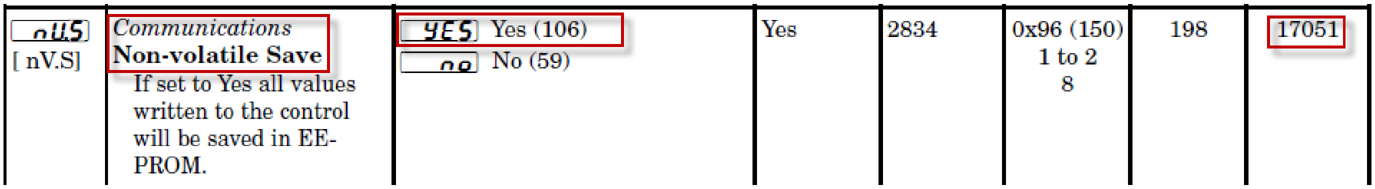
EZ-ZONE® Assembly Programmer

EZ-ZONE® Driver Version 1.01.08

Serial Port COM11

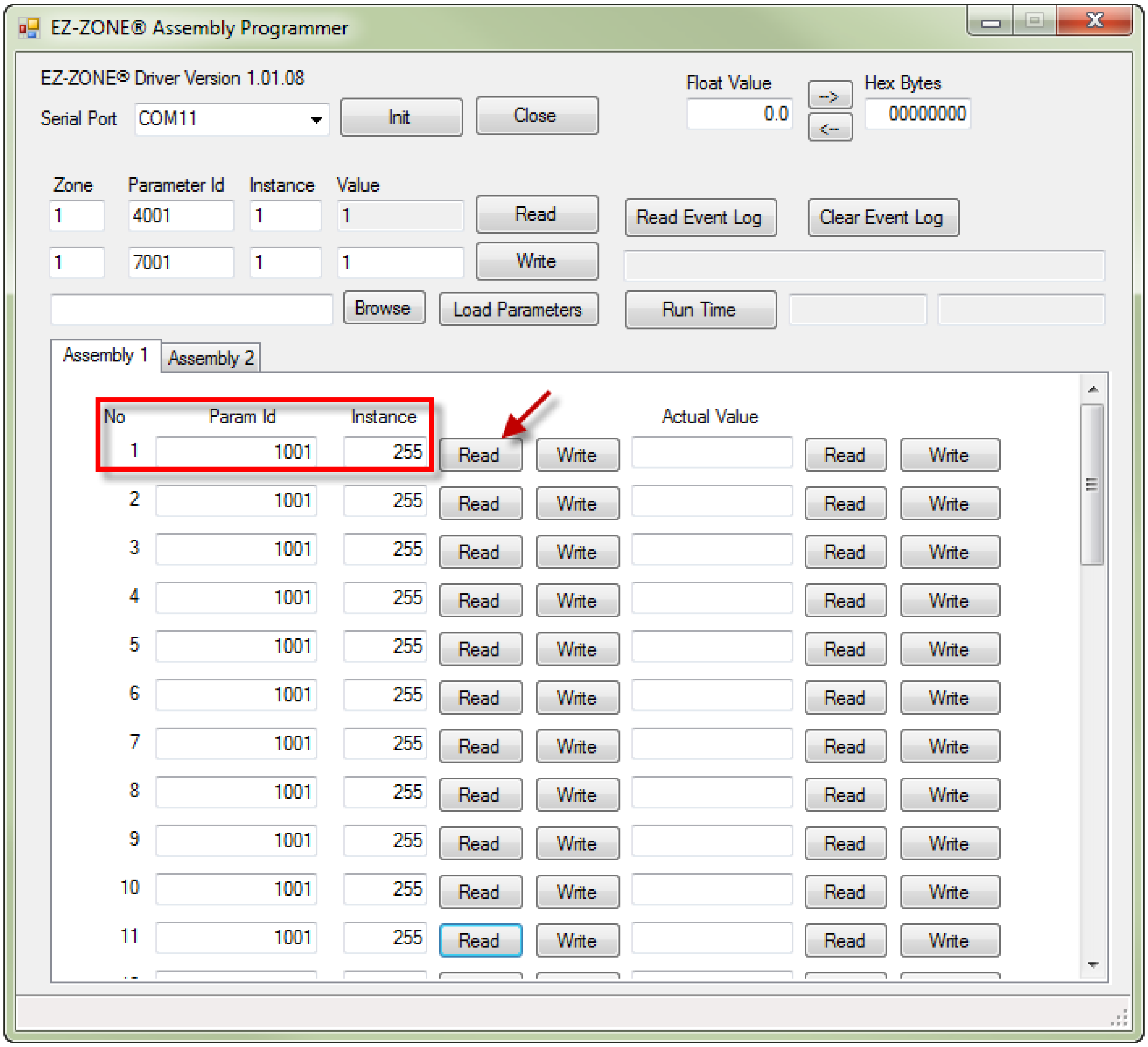
Zone	Parameter Id	Instance	Value
1	4001	1	83.416992
1	7001	1	1

22. Ensure that Non-volatile Save is set to 'Yes' for each controller or module on the network. *This ensures that all changes to the implicit assembly are retained after a power cycle. You can read Parameter Id 17051 for each module to validate. The response should be 106 for 'Yes'.*



nUS [nV.S]	*Communications* **Non-volatile Save** If set to Yes all values written to the control will be saved in EEPROM.	YES Yes (106) no No (59)	Yes	2834	0x96 (150) 1 to 2 8	198	17051

23. Now read each assembly, elements 1 to 40 (or less if using fewer members). *Below, the default for Assembly 1, Element 1; the EZ-ZONE RMH module is 1001 for Parameter Id and 255 for Instance. These indicate that they have never been programmed. Recall that Assembly 1 is for Originator [PLC] to Target [EZ-ZONE]. Assembly 2 is for Target [EZ-ZONE] to Originator [PLC]. Your results may be different than shown here. The idea is to get these to match the spreadsheet created earlier.*



EZ-ZONE® Assembly Programmer

EZ-ZONE® Driver Version 1.01.08

Serial Port: COM11

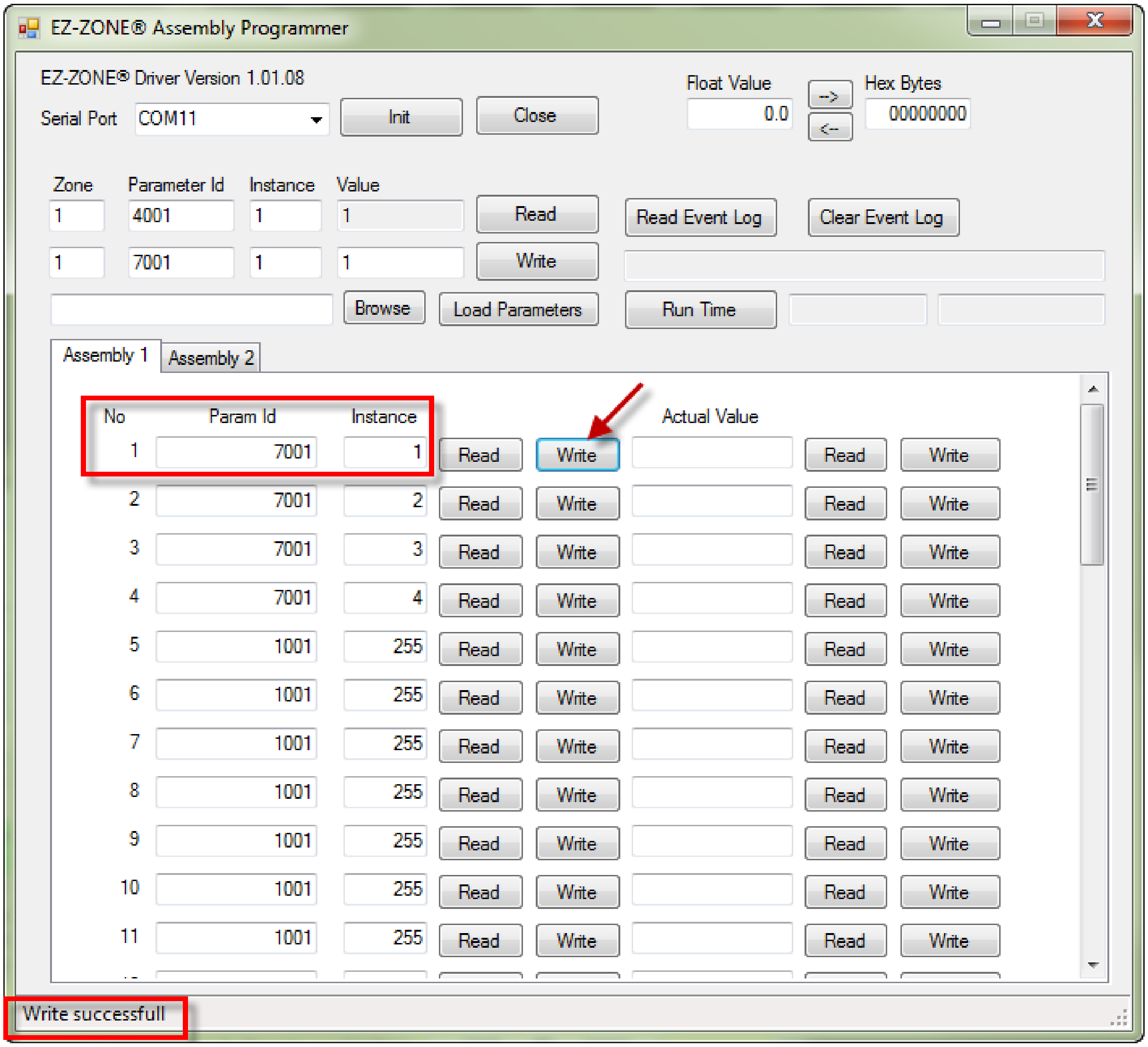
Float Value: 0.0

Hex Bytes: 00000000

Zone	Parameter Id	Instance	Value
1	4001	1	1
1	7001	1	1

No	Param Id	Instance	Actual Value
1	1001	255	
2	1001	255	
3	1001	255	
4	1001	255	
5	1001	255	
6	1001	255	
7	1001	255	
8	1001	255	
9	1001	255	
10	1001	255	
11	1001	255	

24. Write each of the data pointers identified and recorded earlier. *This example places Control Loop 1, Closed Loop Set Point into the implicit table of the Originator [PLC] to Target [EZ-ZONE] – Instance 1 as the No 1 Assembly Row (element). Enter Parameter Id 7001, Instance 1, click on write button. The dialog box indicates a successful write.*



EZ-ZONE® Driver Version 1.01.08

Serial Port COM11

Float Value 0.0

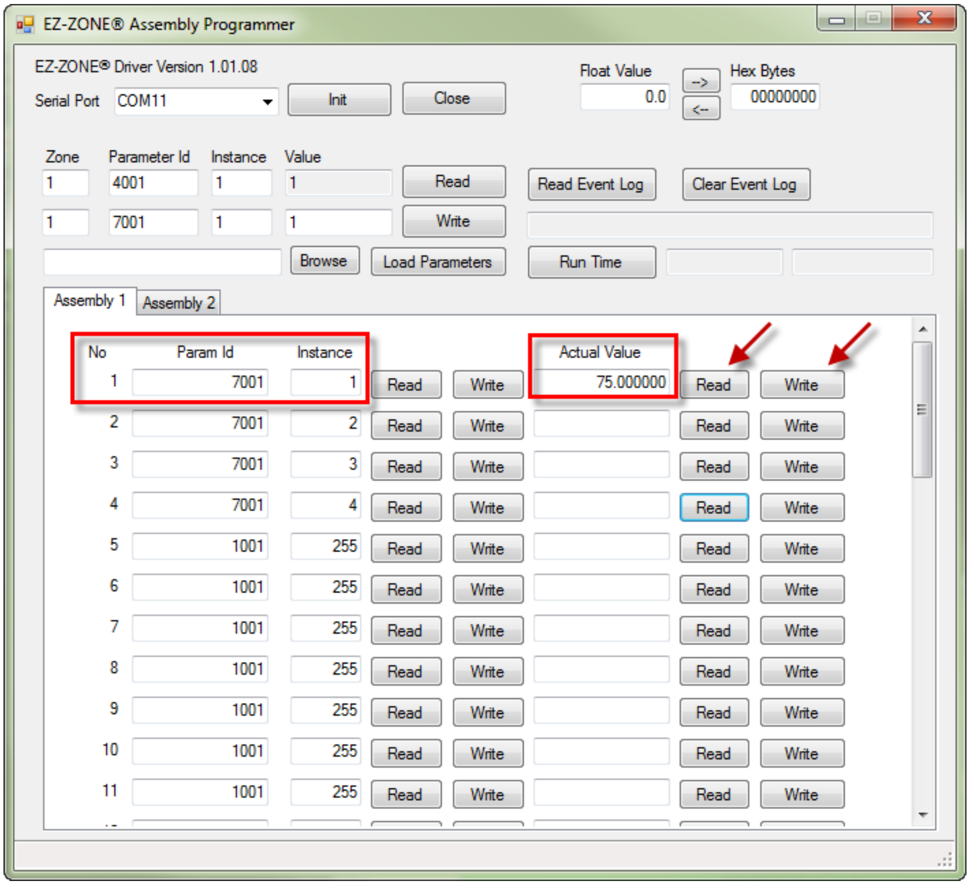
Hex Bytes 00000000

Zone	Parameter Id	Instance	Value
1	4001	1	1
1	7001	1	1

No	Param Id	Instance	Actual Value
1	7001	1	
2	7001	2	
3	7001	3	
4	7001	4	
5	1001	255	
6	1001	255	
7	1001	255	
8	1001	255	
9	1001	255	
10	1001	255	
11	1001	255	

Write successfull

25. Read each of the Values Referenced by Pointer to validate that this is the correct data pointer. *This example reads the value of the Closed Loop 1, Closed Loop Set Point that was placed into the implicit table of the Originator [PLC] to Target [EZ-ZONE] – Instance 1 as the first Assembly Row (element). You may wish to close Parameter Access Tool and execute EZ-ZONE Configurator to validate some values. The close Configurator and re-execute Parameter Access Tool. For controllers that have displays such as the EZ-ZONE PM or systems with RUI/Gateways, you can use the display for validation. At this point you could change the actual value and click on the write button next to the read button just executed to test the ability to affect this parameter.*



EZ-ZONE® Assembly Programmer

EZ-ZONE® Driver Version 1.01.08

Serial Port: COM11

Float Value: 0.0

Hex Bytes: 00000000

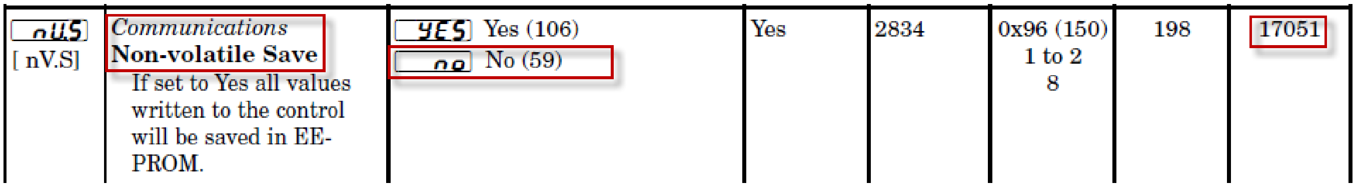
Zone	Parameter Id	Instance	Value
1	4001	1	1
1	7001	1	1

Assembly 1 | Assembly 2

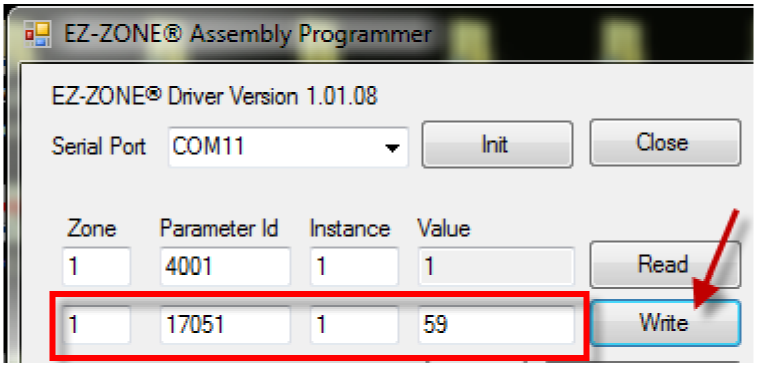
No	Param Id	Instance			Actual Value		
1	7001	1	Read	Write	75.000000	Read	Write
2	7001	2	Read	Write		Read	Write
3	7001	3	Read	Write		Read	Write
4	7001	4	Read	Write		Read	Write
5	1001	255	Read	Write		Read	Write
6	1001	255	Read	Write		Read	Write
7	1001	255	Read	Write		Read	Write
8	1001	255	Read	Write		Read	Write
9	1001	255	Read	Write		Read	Write
10	1001	255	Read	Write		Read	Write
11	1001	255	Read	Write		Read	Write

26. Continue this process for all Assembly Rows (elements) of the implicit table. *Remember that Instance 1 is the PLC sending data to the EZ-ZONE so these must be writable values. Instance 2 is the EZ-ZONE sending data to the PLC so these must be readable values.*

After all configuration and testing is complete, you may wish to disable saving of values to EEPROM of the EZ-ZONE products to prevent early failures from excessive writes. The EZ-ZONE will save any changed values sent to the product every 5 seconds. To accomplish changing Non-volatile Save in each controller or module to No (don't save), write to Parameter ID 17051, Instance 1 a value of 59 for each controller or module on the network connected via RUI/Gateway or Access module. For EZ-ZONE PM controllers that have Ethernet or DeviceNet directly in controller, write Parameter ID 17051, Instance 2 a value of 59 for each controller.



nUS [nV.S]	*Communications* **Non-volatile Save** If set to Yes all values written to the control will be saved in EE-PROM.	YES Yes (106) no No (59)	Yes	2834	0x96 (150) 1 to 2 8	198	17051



EZ-ZONE® Assembly Programmer

EZ-ZONE® Driver Version 1.01.08

Serial Port: COM11

Zone	Parameter Id	Instance	Value
1	4001	1	1
1	17051	1	59

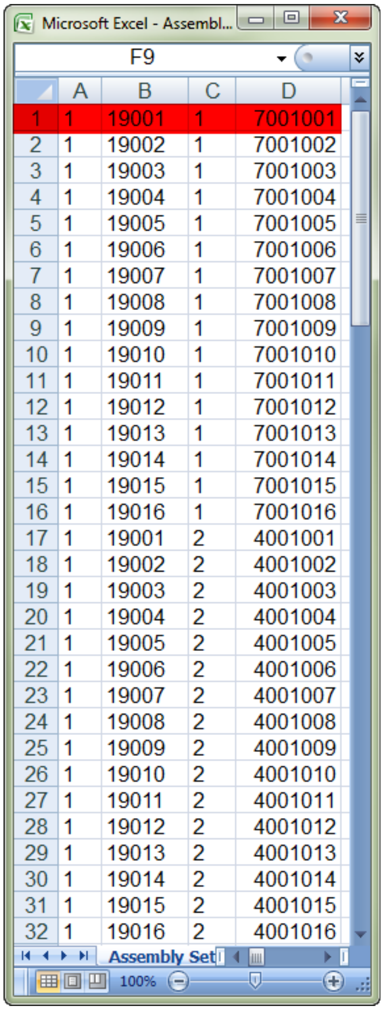
That completes the topic of individually programming the implicit assembly.

Now suppose that you want to repeat programming other systems with the same implicit assemblies in a quicker method. The EZ-ZONE Assembly Programmer Application allows you to create a special CSV text file containing all of the assembly setups for all of the controllers on the same standard bus network.

Here is how to accomplish that task.

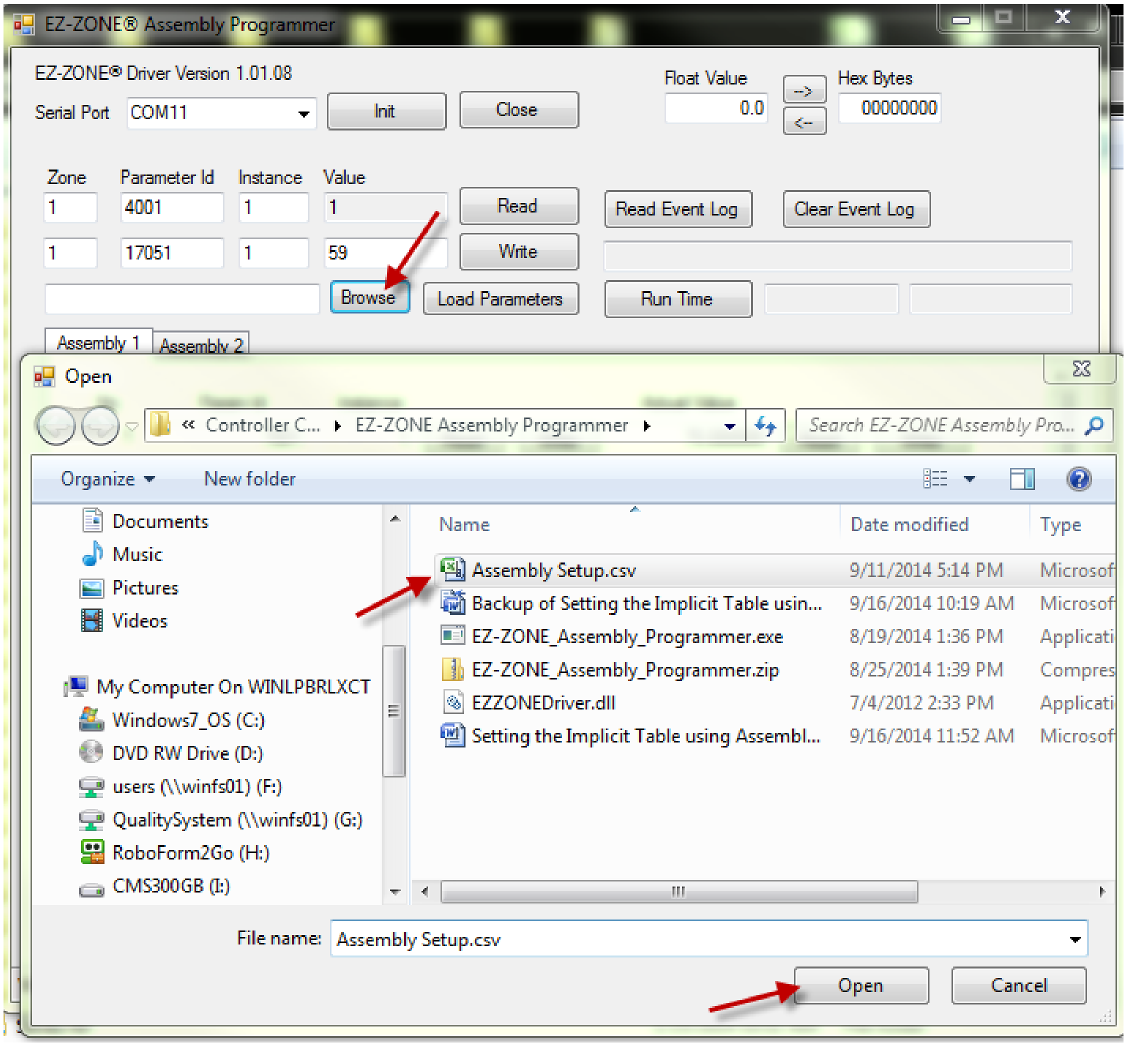
1) Create a text file in comma separated format (CSV) with the format of Zone, table row Parameter ID, Assembly Instance (1 or 2) and Parameter ID with Instance of Write Value. *Below we show the zone in Column A to be 1. Column B is the parameter id of the table row (element) starting at 19001 to 19016 representing rows (elements) 1 to 16. This example has only 16 elements being defined for each assembly. Column C is the assembly instance of 1 for Originator (PLC) to Target (EZ-ZONE) or assembly instance of 2 for Target (EZ-ZONE) to Originator (PLC). Column D is the Parameter ID with the instance appended. Instance 1 is shown as 001, 2 as 002 and so on. The range of instances is 001 to 255.*

In row 1 of this example spreadsheet, Zone 1 is programming element 1 of assembly 1 to 7001001 which is Parameter 7001 (Control Loop, Closed Loop Set Point) and 001 is instance 1 meaning Control Loop 1.



	A	B	C	D
1	1	19001	1	7001001
2	1	19002	1	7001002
3	1	19003	1	7001003
4	1	19004	1	7001004
5	1	19005	1	7001005
6	1	19006	1	7001006
7	1	19007	1	7001007
8	1	19008	1	7001008
9	1	19009	1	7001009
10	1	19010	1	7001010
11	1	19011	1	7001011
12	1	19012	1	7001012
13	1	19013	1	7001013
14	1	19014	1	7001014
15	1	19015	1	7001015
16	1	19016	1	7001016
17	1	19001	2	4001001
18	1	19002	2	4001002
19	1	19003	2	4001003
20	1	19004	2	4001004
21	1	19005	2	4001005
22	1	19006	2	4001006
23	1	19007	2	4001007
24	1	19008	2	4001008
25	1	19009	2	4001009
26	1	19010	2	4001010
27	1	19011	2	4001011
28	1	19012	2	4001012
29	1	19013	2	4001013
30	1	19014	2	4001014
31	1	19015	2	4001015
32	1	19016	2	4001016

2) After the complete CSV file is created for all zones with each zone assembly as desired, open the file. Click on ‘Browse’. Select the file and select ‘Open’.



EZ-ZONE® Assembly Programmer

EZ-ZONE® Driver Version 1.01.08

Serial Port: COM11

Float Value: 0.0

Hex Bytes: 00000000

Zone	Parameter Id	Instance	Value
1	4001	1	1
1	17051	1	59

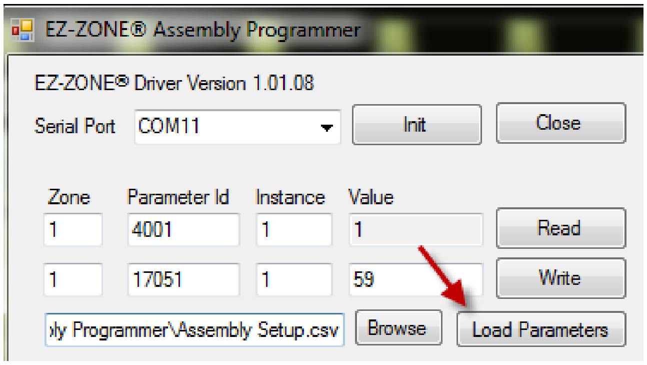
Open

« Controller C... ▸ EZ-ZONE Assembly Programmer ▸

Name	Date modified	Type
Assembly Setup.csv	9/11/2014 5:14 PM	Microsof
Backup of Setting the Implicit Table usin...	9/16/2014 10:19 AM	Microsof
EZ-ZONE_Assembly_Programmer.exe	8/19/2014 1:36 PM	Applicati
EZ-ZONE_Assembly_Programmer.zip	8/25/2014 1:39 PM	Compres
EZZONEDriver.dll	7/4/2012 2:33 PM	Applicati
Setting the Implicit Table using Assembl...	9/16/2014 11:52 AM	Microsof

File name: Assembly Setup.csv

3) Click on ‘Load Parameters’ to send file to controllers.



EZ-ZONE® Assembly Programmer

EZ-ZONE® Driver Version 1.01.08

Serial Port: COM11

Zone	Parameter Id	Instance	Value
1	4001	1	1
1	17051	1	59

ıly Programmer\Assembly Setup.csv

This completes the instructions on the use of this application!