

Application Note

MicroLogix 1100 – MPiec Communication Configuration over EtherNet/IP

Applicable Product: MPiec

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

Table of Contents

1. Application Overview:	3
a. Application Highlights	3
b. Application Challenges	3
c. Products Used:	3
d. System layout	4
e. Application Solution	4
2. Establishing communication	4
3. Using Message Instructions to Communicate.....	7
a. Create five new data files	8
b. Add Message Instructions	14
c. Setting up Message Instructions.....	16
4. Sample Slave program	19
5. Communication Verification	22

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

1. Application Overview:

This document is intended to guide a user through the steps of setting up and configuring an AB MicroLogix 1100 PLC as a scanner to communicate with an MPiec controller which will be an adapter in this set up. Software and hardware requirements, and the procedure to establish connection are detailed in the following sections of this document.

a. Application Highlights

Industry: Motion industry with AB MicroLogix based EtherNet/IP requirements.

Major Features: Yaskawa MPiec controllers have EtherNet/IP assembly groups built in by default when the MPiec is configured as an adapter

b. Application Challenges

- Make sure to work with required software and firmware versions of software and hardware

c. Products Used:

Controller	Yaskawa MPiec controller
PLC	MicroLogix 1100 , Bul 1763, Series B
Software	RSLogix 500 version 8.10.00, MotionWorksIEC 1.5.2, RSLinx v. 2.54
Hardware	Ethernet cables, hub

Only version 8.00.00 and higher can be used for Ethernet IP communication

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

d. System layout

Figure 1 is a schematic that lays out the goal of this project.

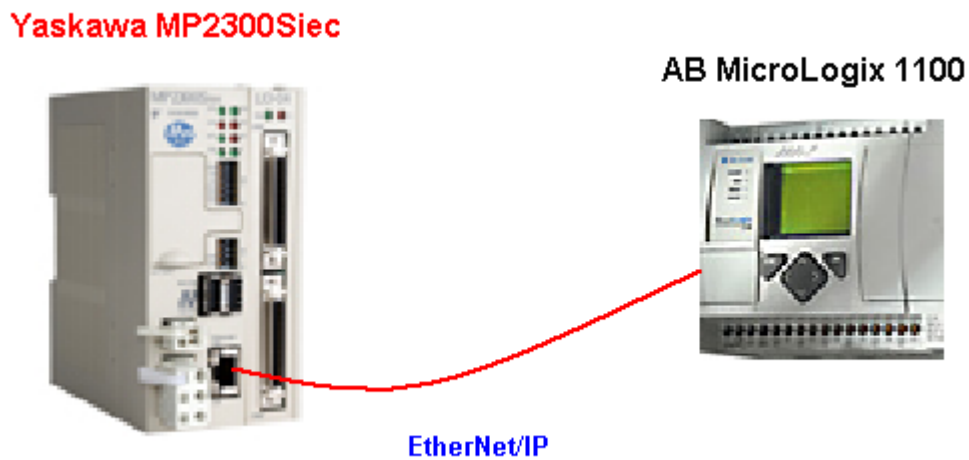


Figure 1: System Layout

e. Application Solution

The steps to be followed are:

- a) Establish Communication.
- b) Use Message Instructions to Communicate.
- c) Create Slave Program.
- d) Verify Communication.

2. Establishing communication

RSLink Classic Lite should be used to install drivers for Ethernet IP communications and establishing connection between the various devices. The version of RSLink that can be used for establishing communication is given below in Figure 2.

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

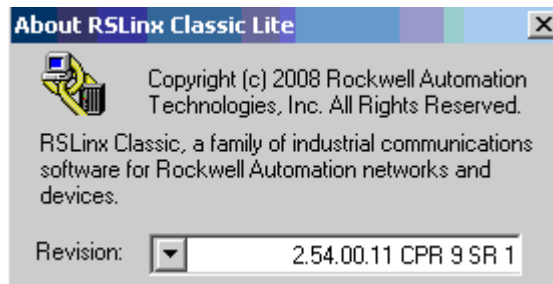


Figure 2: RSLinx version number.

Open RSLinx Classic Lite.

Click Communications > Configure Drivers

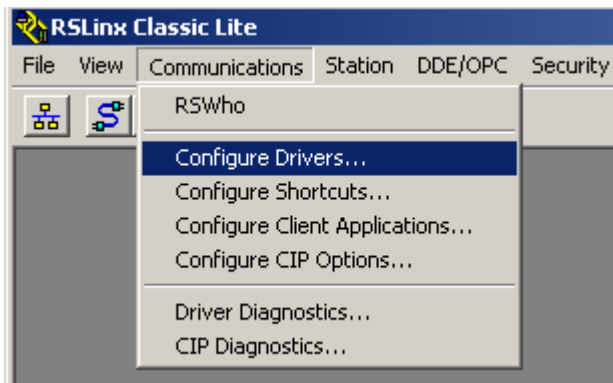


Figure 3: Configuring Ethernet IP driver

From the drop down, select Ethernet IP drivers (Figure 4)

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

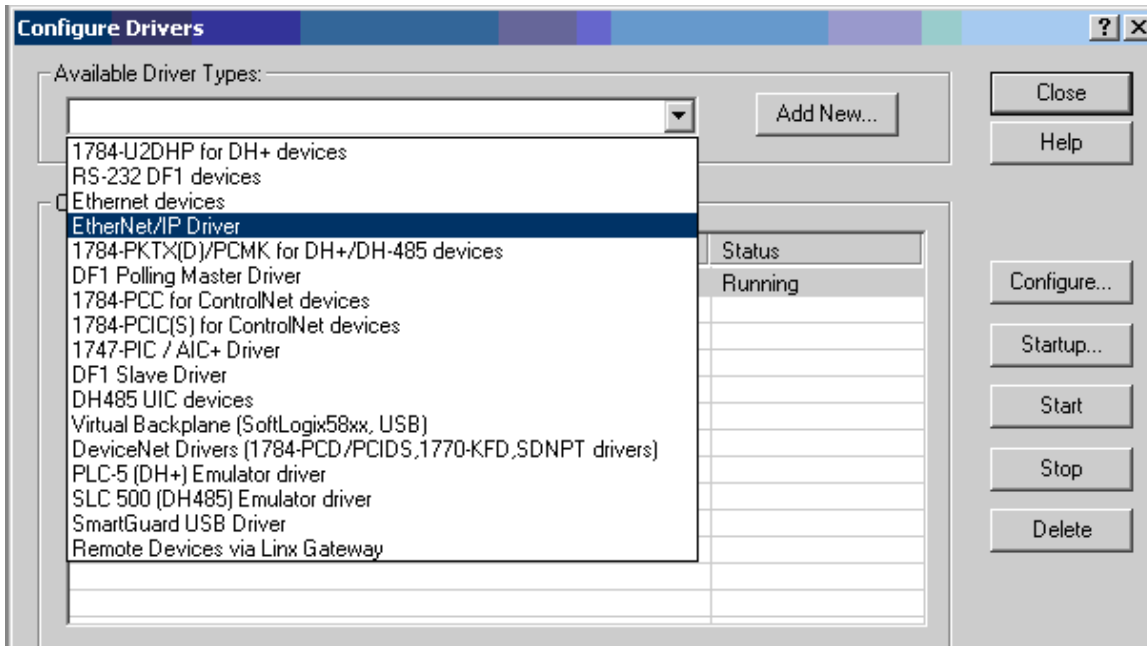


Figure 4: Selecting EtherNet/IP Driver

Click Add new.

You will see the driver updated and be in run status. You may be asked to map the station of your device. You will have to enter the IP address of the MicroLogix device.

Go to communications menu and click RSWho (figure 5)

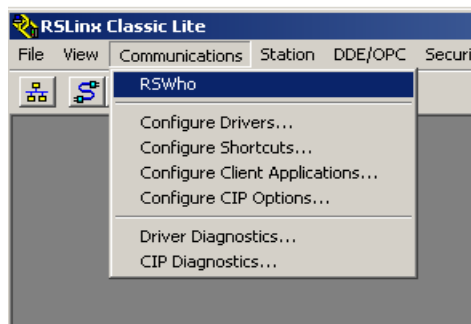


Figure 5: Verifying driver install

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

This opens the page with connections displayed. Click on the installed driver AB_ETHIP-1 and all the devices connected will appear in the right side panel. (Figure 6)

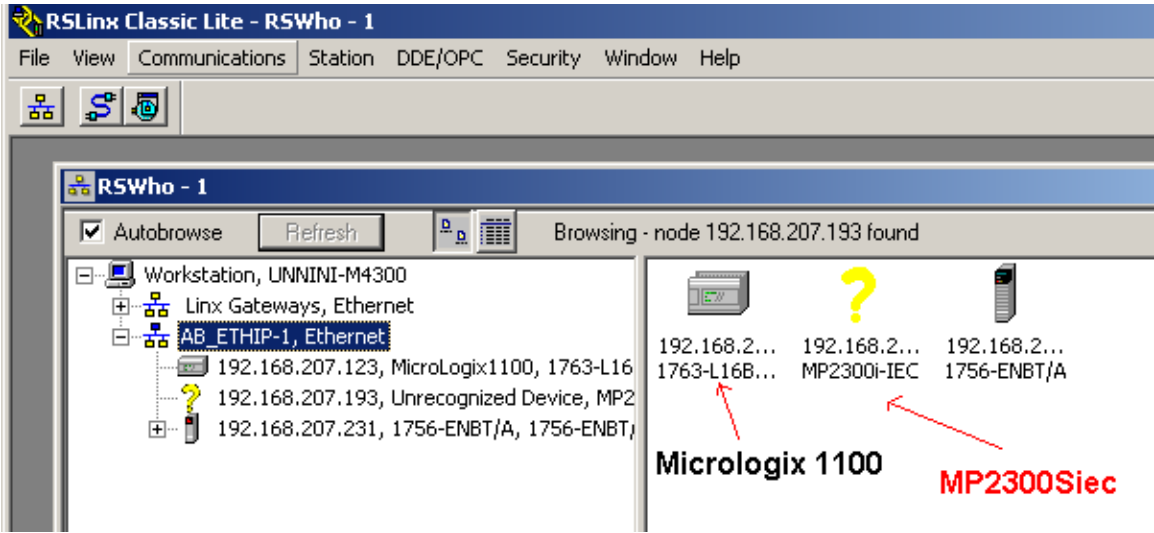


Figure 6: Installed devices

This is a check to see if the devices are visible on the network.

3. Using Message Instructions to Communicate

This document shows how to test communication between the AB MicroLogix PLC and Yaskawa MPiec controller. The MicroLogix controller will be the scanner and the MPiec controller will be the slave in this procedure. Instructions in the ladder program in RSLogix 500 will be used in communication.

Table 1: Reading Data (Data sent from the MPiec to the MicroLogix)

Parameter	Value
Service type	Read Assembly
Service Code	E (hex)
Class	4 (hex)
Instance	101, 103, etc
Attribute	3 (hex)

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

Table 2: Writing Data (Data sent from the MicroLogix to the MPiec)

Parameter	Value
Service type	Write Assembly
Service Code	10(hex)
Class	4 (hex)
Instance	111, 113, etc
Attribute	3 (hex)

NOTE:

Only RSLogix 500 version 8.0 or higher can be used to configure message instructions to communicate with an EtherNet/IP device.

a. Create five new data files

A Message (MG) file, an Extended Routing Information (RIX file), a timer file, an integer file to contain data from the MPiec controller and an integer file to contain data sent to the MPiec controller.

To create a Message data file:

Right Click Data Files. Select new. Create a message file. (Figure 7). MG9 is the message file. (Figure 8)

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

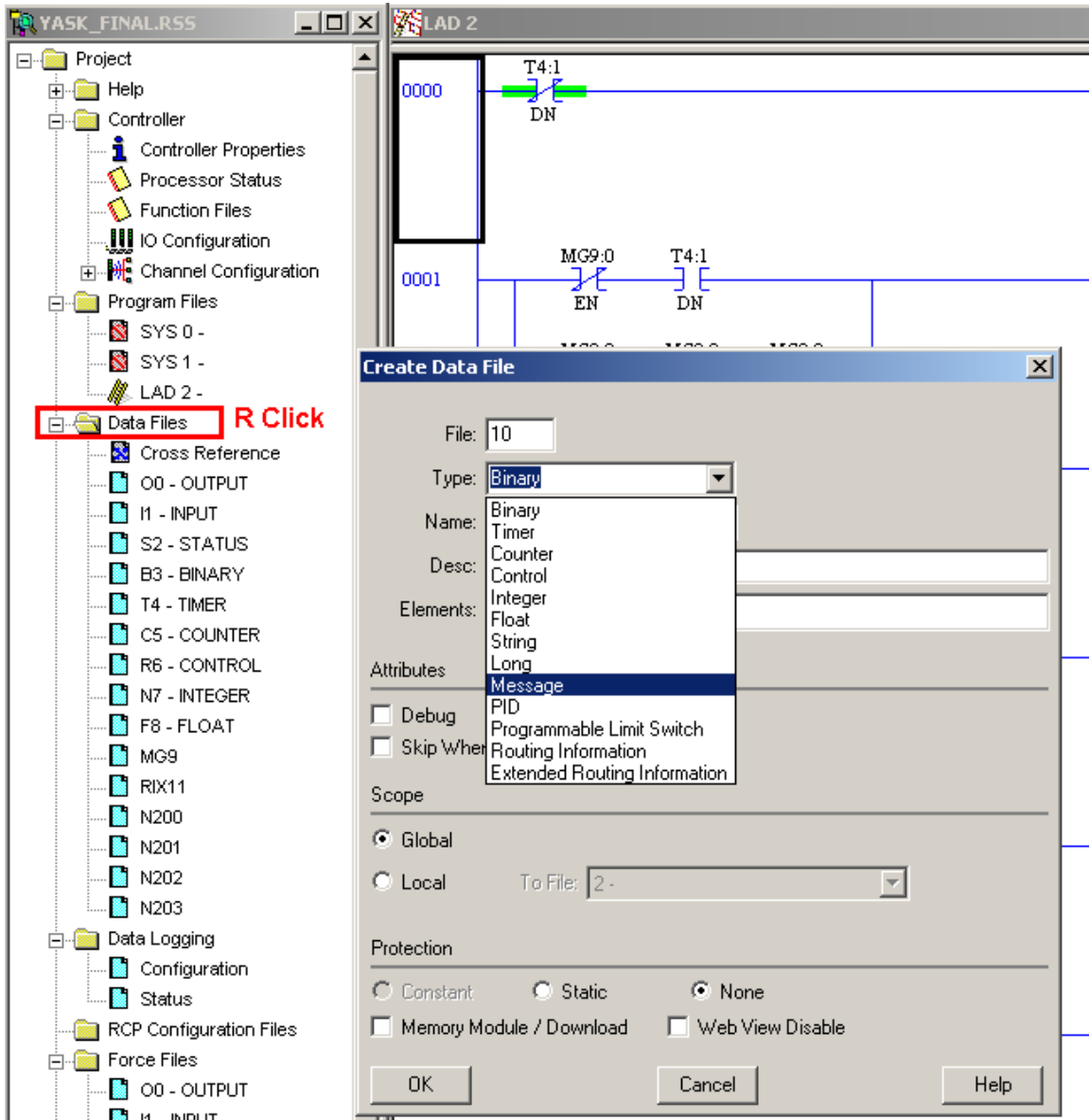


Figure 7: Creating data files in RSLogix 500

Properties of the MG file are given below. This file must have at least two elements, one to control the Read Operation and one to control the Write Operation

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

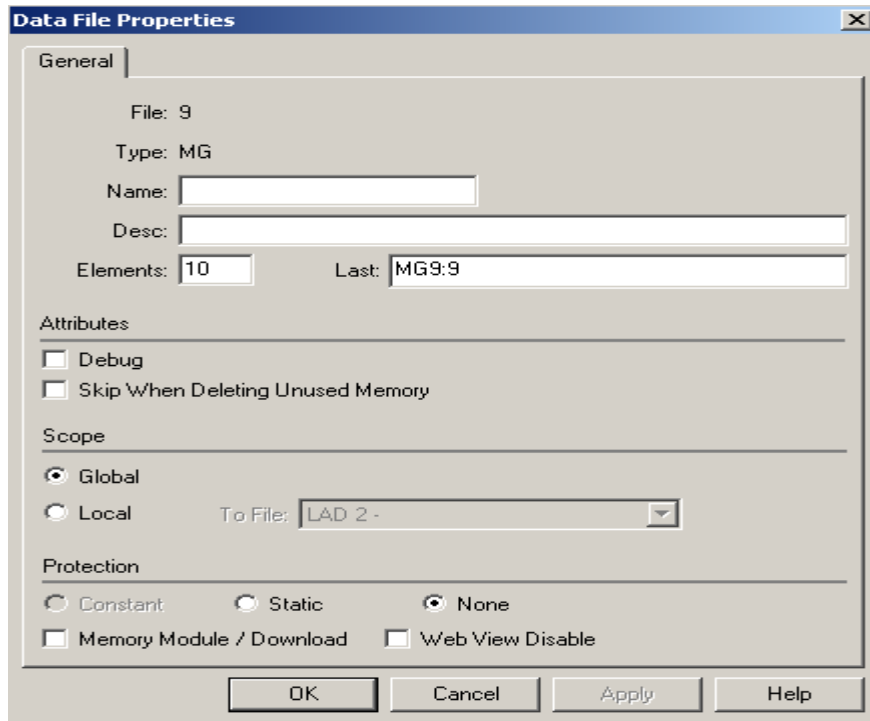


Figure 8: Creation of a Message file (MG9)

To create an Extended Routing Information data file:

Right Click Data Files. Select new. Create an Extended Routing Information file. RIX11 is the routing information file. (Figure 9)

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

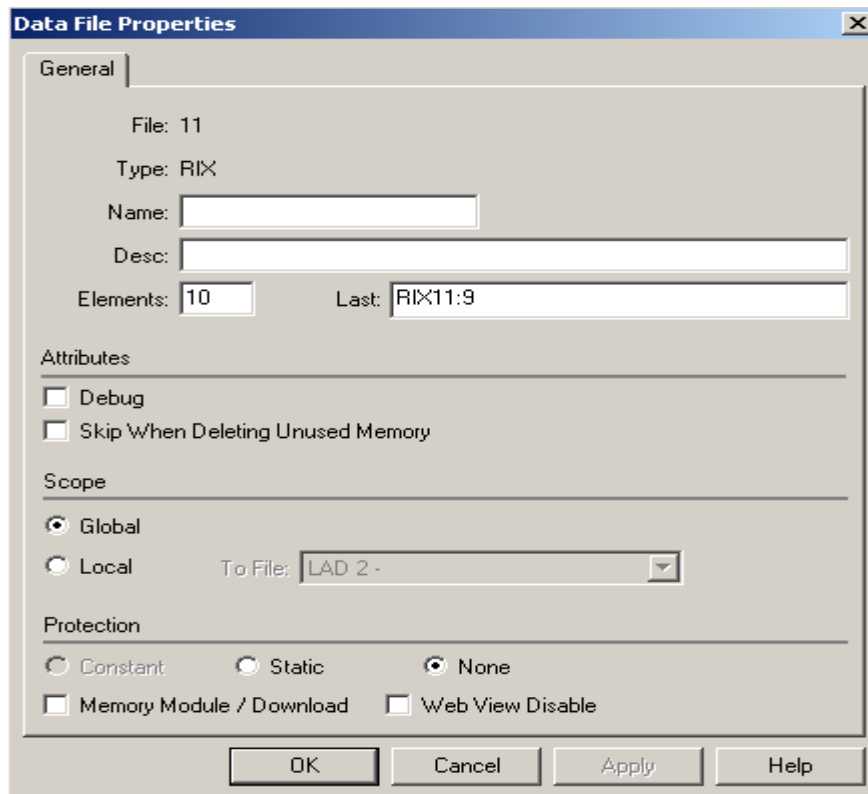


Figure 9: Properties of RIX file

To create integer data files:

Right Click Data Files. Select new. Create two Integer files. N202 is the integer file to write data into the MPiec. (Figure 10). N203 is the integer file to read data from the MPiec. (Figure 11)

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

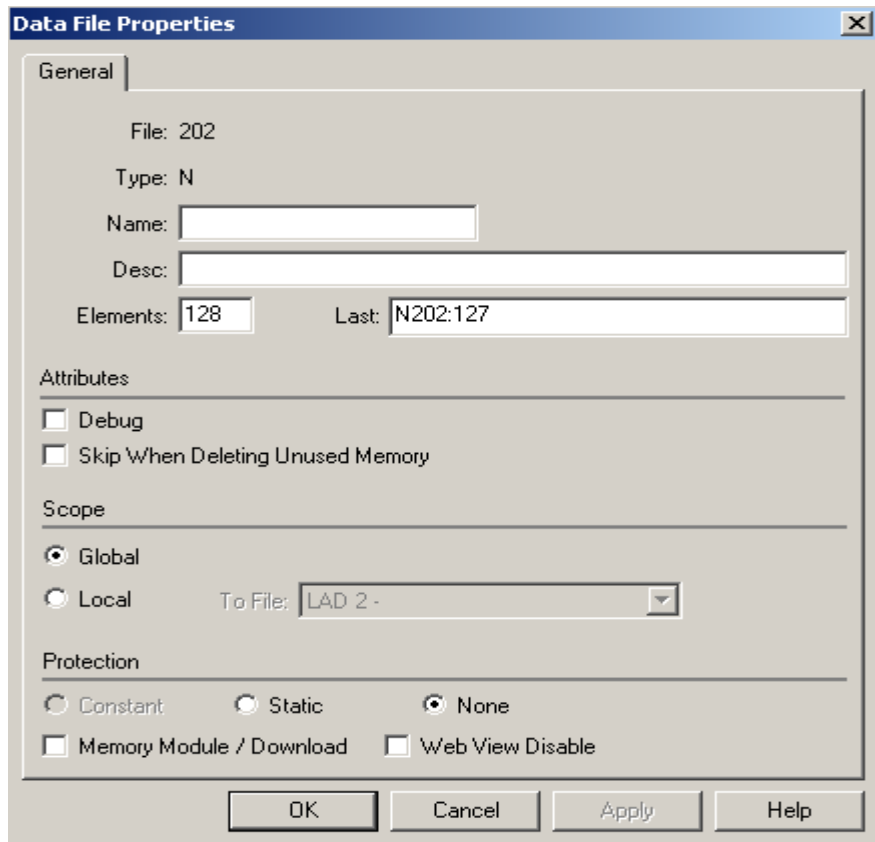


Figure 10: New data file for writing data to the MPiec

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

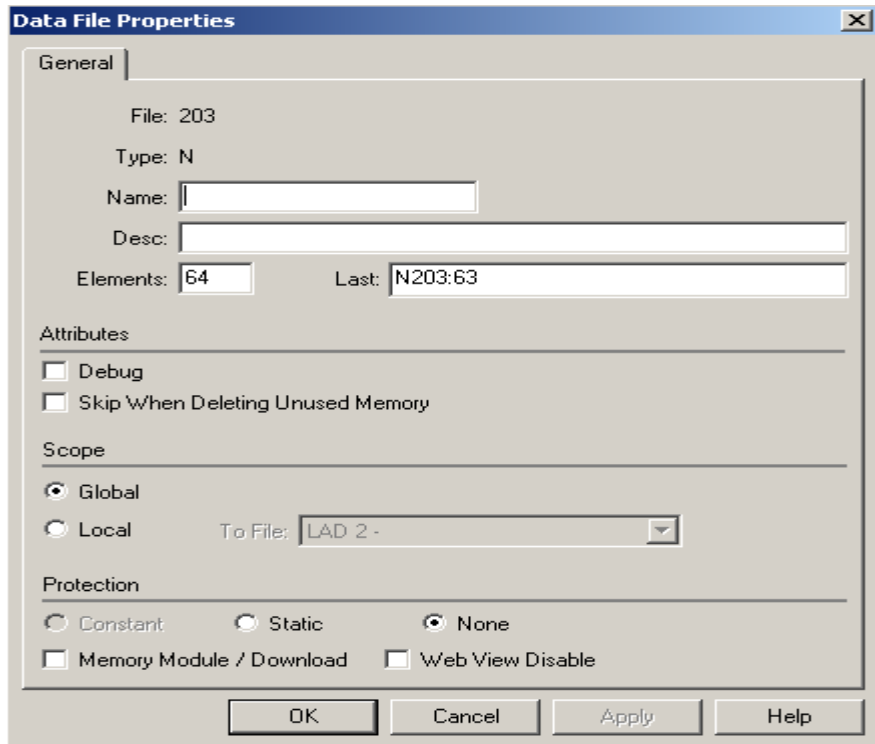


Figure 11: New data file for reading data from the MPiec

To use a timer file:

The timer file properties are as below (Figure 12).

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

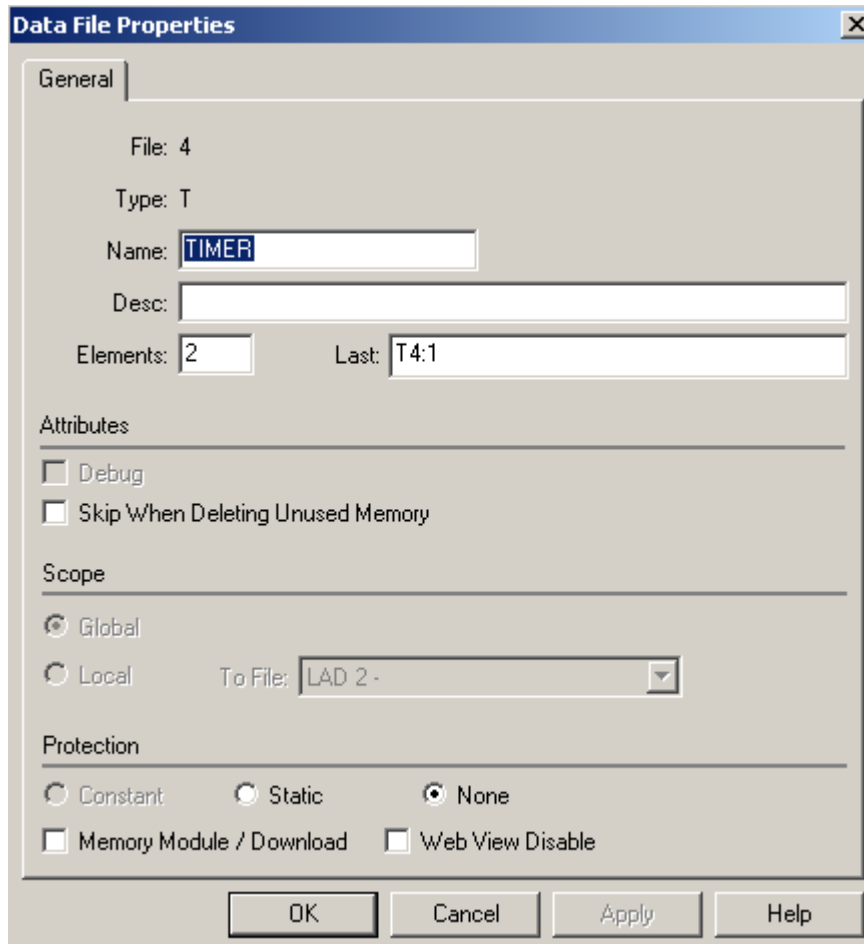


Figure 12: Timer file properties

b. Add Message Instructions

The following rungs show how data can be read and written to the MPiec unit.

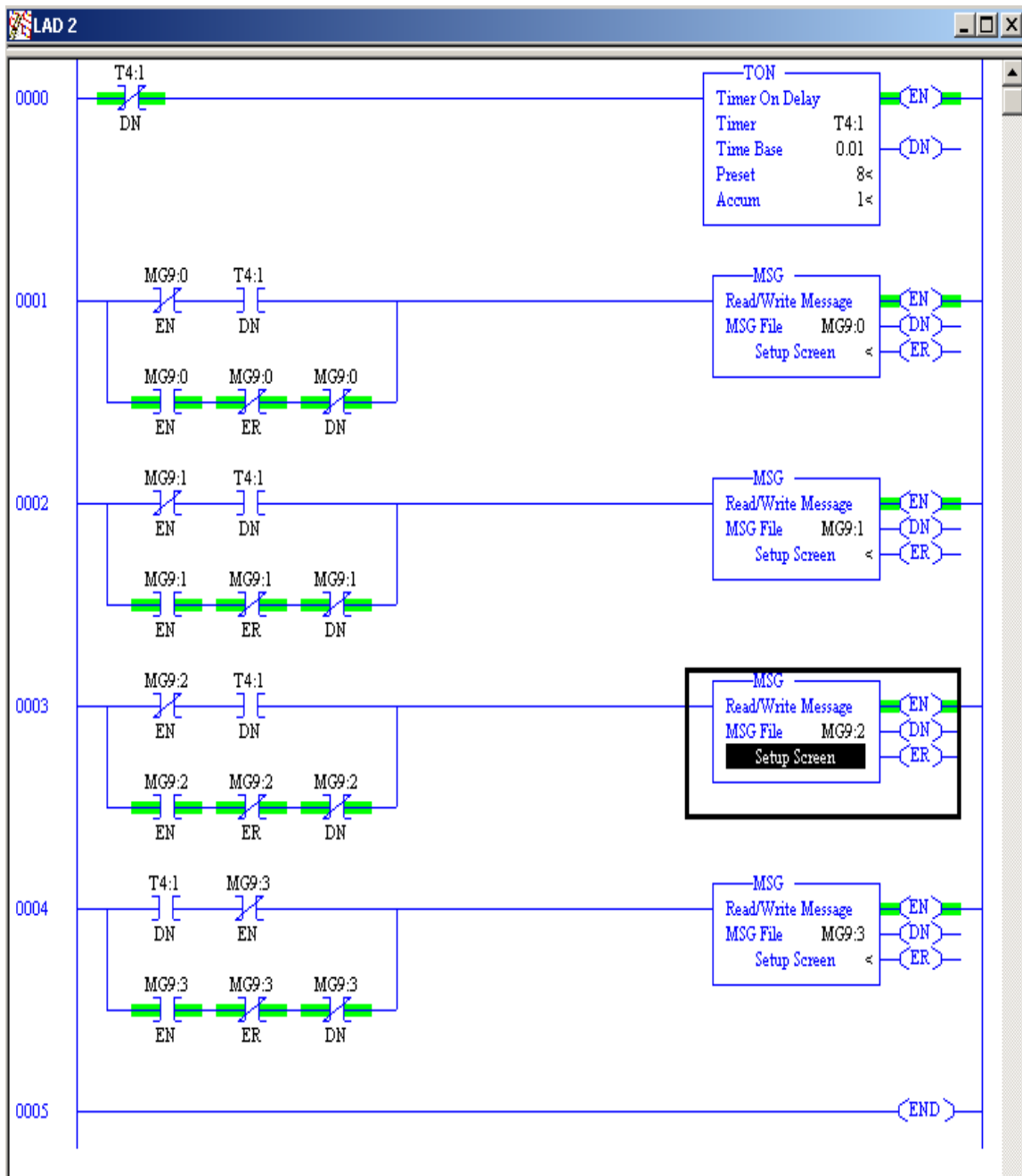


Figure 13: Rungs for Message Instructions

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

c. Setting up Message Instructions

Double click on the Setup Screen to open the MSG file configuration page. Follow the procedure given below to set up the message file to send and receive messages.

- a. Double Click in the *Channel* field, select *1 (Integral)*, from the drop down menu and press Enter. (Figure 14). Make sure that channel 1 has been configured with the correct IP settings. This can be verified by double clicking on *channel configuration* in the project tree.
- b. Double Click in the *Communication Command* field, select *CIP Generic* and press Enter.
- c. If the Message Instruction is being used to write data to the MPiec unit, enter the integer file where the data will be placed in the *Data Table Address (Send)* field and press enter. If the Message Instruction is being used to read data to the MPiec unit, enter the integer file where the data will be located in the *Data Table Address (Receive)* field and press Enter. (Figures 14 and 16)
- d. Verify the instance size on the slave side. There are pre-defined instance groups created in every new MotionWorksIEC new program if the MPiec controller is being used as a slave. (When a user chooses to open a new program in MotionWorksIEC, the new program has EtherNet/IP instances already created and ready for use. The slave output instances are 101 to 106 and input instances are 111 to 116. The user will only have to create variables in these pre defined groups). Enter the data size in RSLogix 500 based on what instance is going to be used. (Figure 18)
- e. Enter a RIX address in the *Extended Routing Info* field. Please note that each Message Instruction must have its own RIX address.
- f. Double Click in the *Service* field and select *Read Assembly* for a Message Instruction that is being used to read data from the MPiec unit or *Write Assembly* for a Message Instructions that is being used to send data to the MPiec unit, and press Enter.
- g. For read operations, the *Service Code* field will change to E (hex). For write operations, the *ServiceCode* field will change to 10 (hex). For both read and write operations, the *Class* field will change to 4 (hex), and the *Attribute* field will change to 3 (hex).
- h. For read operations, enter a value of 101, 103, 105 etc in the *Instance* field. For write operations, enter a value of 111, 113, 115 etc in the *Instance* field.

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

- i. Click on the MultiHop tab and enter the IP address of the MPiec controller to which data will be sent if data is being written or from where data will be received if it is a receive instruction. (Figure 15)
- j. The images below show a typical configuration for Message Instructions being used to both read data from and write data to an MPiec unit.

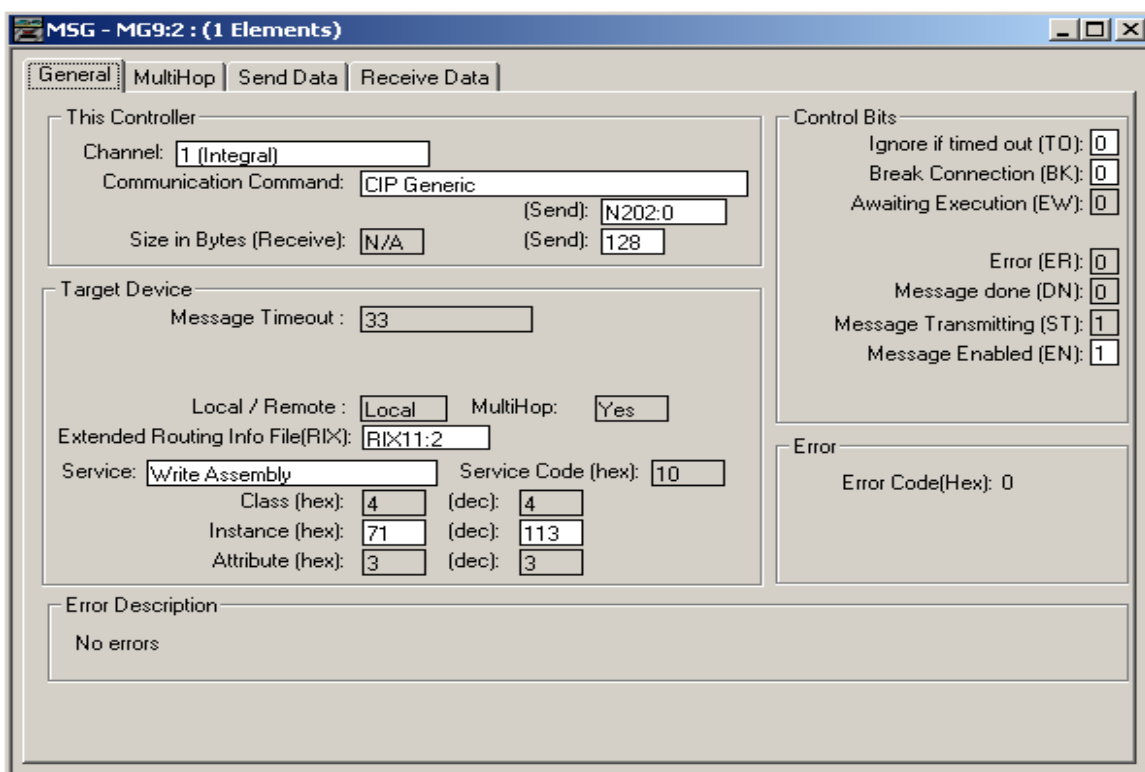


Figure 14: Configuring a write instruction

MultiHop data for message send address specification.

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

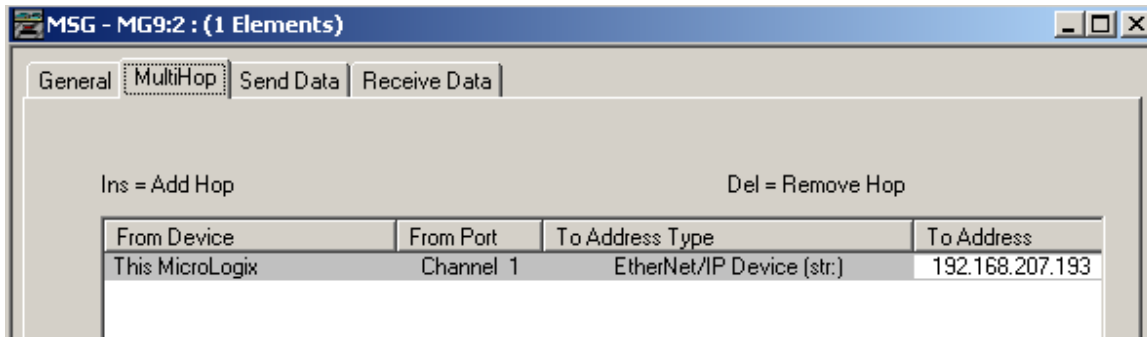


Figure 15: Configuring MultiHop for send instruction

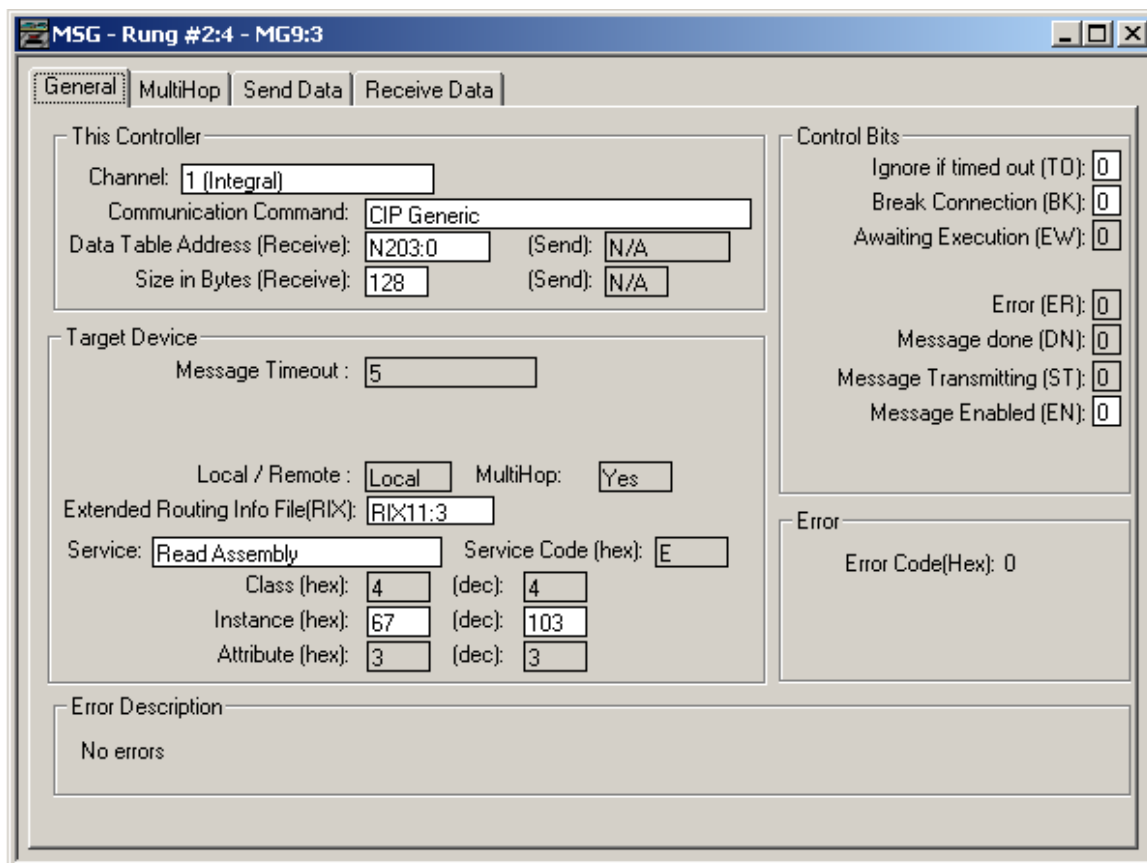


Figure 16: Configuring a read instruction

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

MultiHop data for message receive address specification.

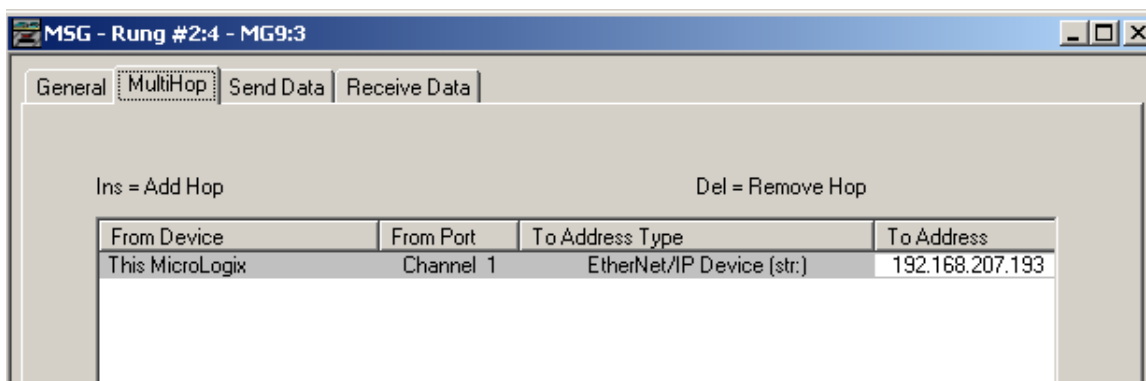


Figure 17: Configuring MultiHop for receive instruction

After the read and write message instructions have been placed in the program, save and download the program to the PLC. Put the PLC in Run mode and communication will start immediately.

4. Sample Slave program

EtherNet/IP groups (when the MPiec controller is a slave) are already built in to every new program created in MotionWorksIEC. Create new variables in the groups (instances) which will be used for communication. In this example FromMicro2 is a variable that will have data sent from the MicroLogix PLC and ToMicro2 is a variable that will be used to send data to the PLC from the MPiec. Addressing is important as the address of these variables will be based on data type of the variable being used. IW and QW (Input Word and Output Word) are used for the above mentioned variables because they are of INT type and use 16 bits. (Figure 18)

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

Name	Online value	Type	Usage	Description	Address
System					
EIP Input Instance #111, Qty: 128 Bytes, Address Range: %IB0 - %IB127					
FromMicro1	2	DINT	VAR_GLOBAL		%M384
EIP Input Instance #112, Qty: 256 Bytes, Address Range: %IB128 - %IB383					
EIP Input Instance #113, Qty: 128 Bytes, Address Range: %IB384 - %IB511					
FromMicro2	100	INT	VAR_GLOBAL		%M384
EIP Input Instance #114, Qty: 256 Bytes, Address Range: %IB512 - %IB767					
EIP Input Instance #115, Qty: 128 Bytes, Address Range: %IB768 - %IB895					
EIP Input Instance #116, Qty: 256 Bytes, Address Range: %IB896 - %IB1151					
EIP Output Instance #101, Qty: 128 Bytes, Address Range: %QB0 - %QB127					
ToMicro1	0	DINT	VAR_GLOBAL		%QD0
EIP Output Instance #102, Qty: 256 Bytes, Address Range: %QB128 - %QB383					
EIP Output Instance #103, Qty: 128 Bytes, Address Range: %QB384 - %QB511					
ToMicro2	101	INT	VAR_GLOBAL		%QW384
EIP Output Instance #104, Qty: 256 Bytes, Address Range: %QB512 - %QB767					
EIP Output Instance #105, Qty: 128 Bytes, Address Range: %QB768 - %QB895					

Figure 18: Global variable list in the MPiec program

The exercise carried out in the test project is as follows

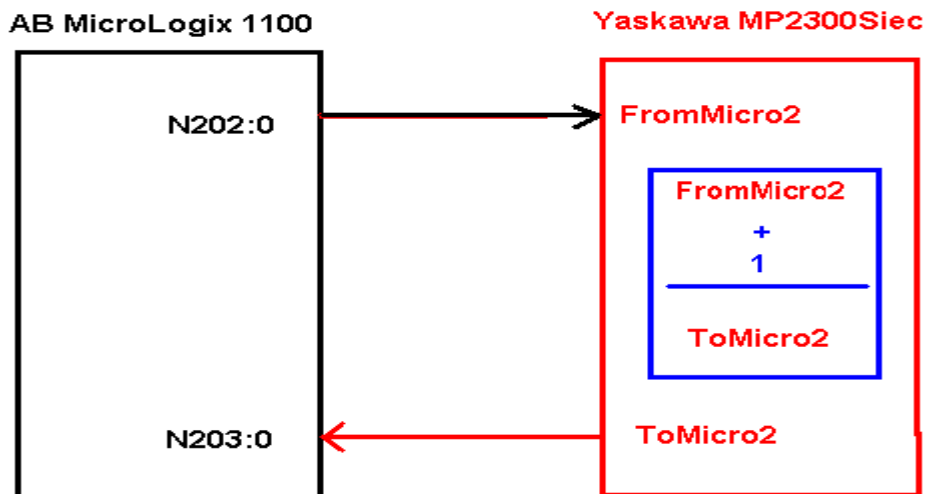


Figure 19: Test exercise

Create the following program on the slave MPiec controller. Download the program and run the controller.

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

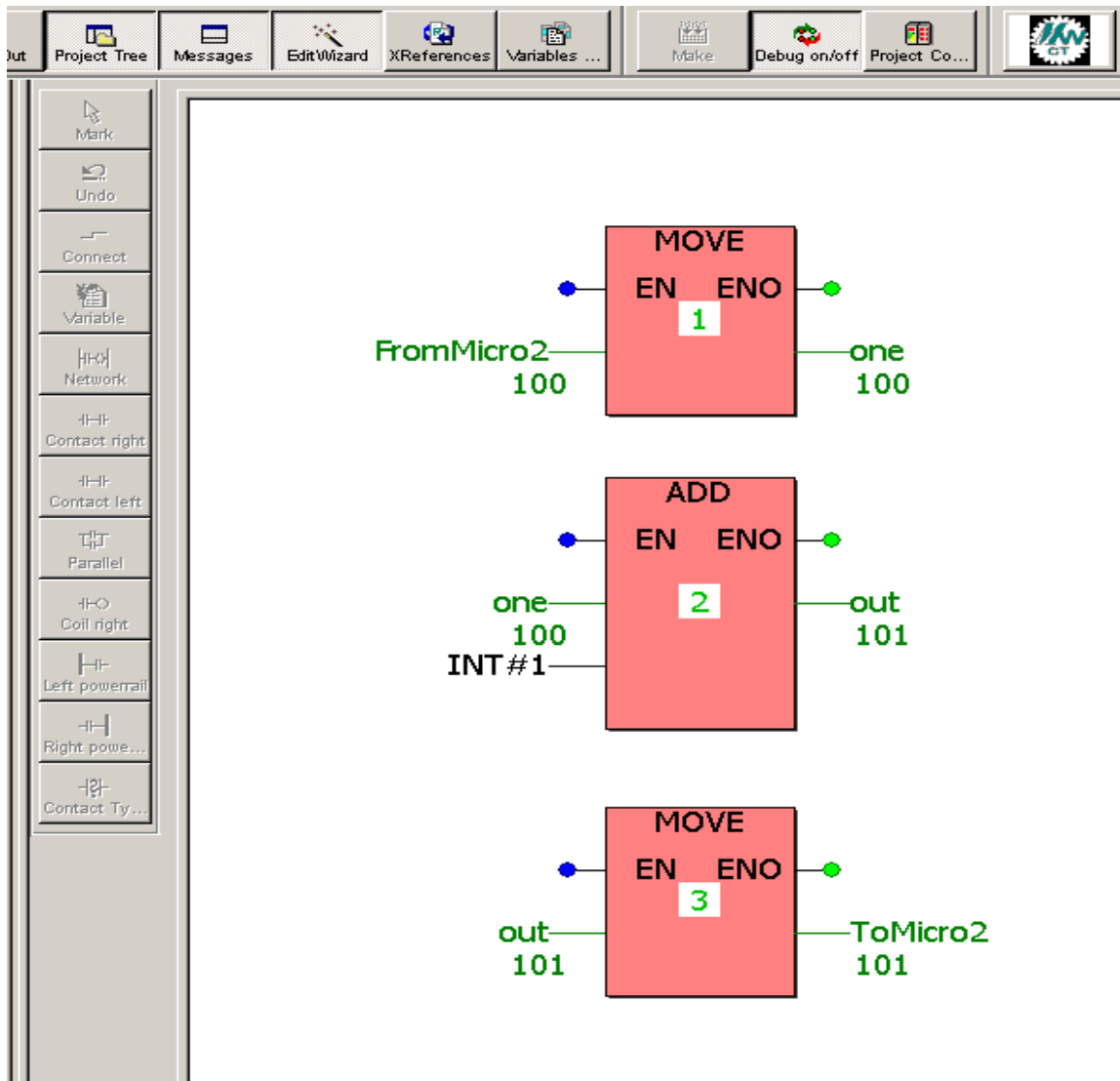


Figure 20: Program in MPiec

On running the MPiec controller in Debug mode, the data from the MicroLogix controller will be seen in the MPiec variable FromMicro2. ToMicro2 will be sent out to the MicroLogix controller.

Subject: Application Note	Product: MPiec	Doc#: AN.MCD.08.108
Title: MicroLogix – MPiec Communication Configuration over EtherNet/IP		

5. Communication Verification

Open up the data file N202. Input data in N202:0. Open N203. Based on the logic in the MPiec, N203:0 = N202:0 + 1 . This can be verified by opening up the data file N203 simultaneously. (Figure 21)

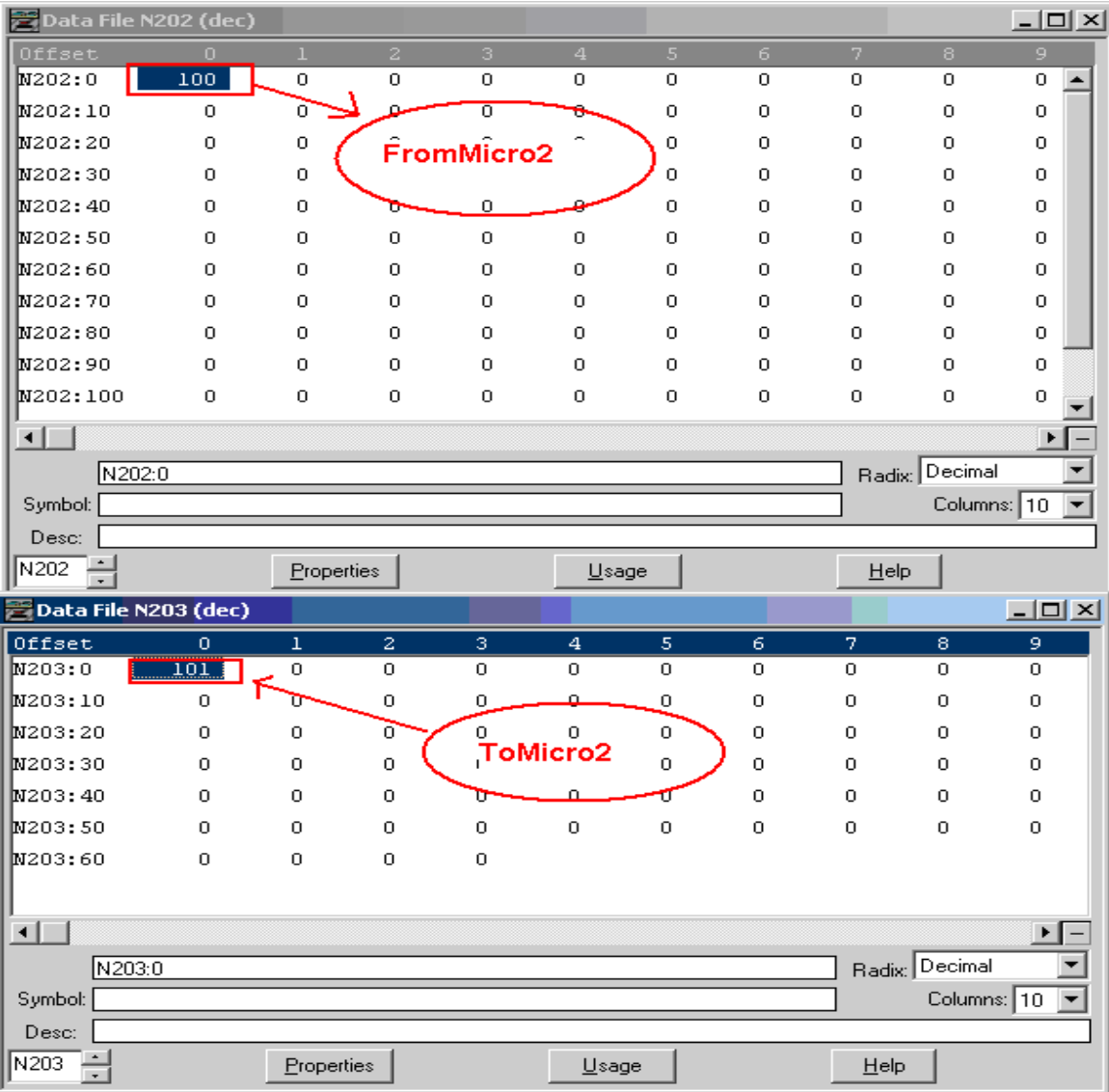


Figure 21: Verification of write and read message over EtherNet/IP