

Application Note 008  
Version 003  
20 August 2021

## Configuring an EtherMeter® and PLC for Pulse-Based Flow Metering and Batching Operations.

**(Note: The example in this document uses an Allen Bradley MicroLogix 1400 PLC and the RsLogix programming environment. However, these procedures can be generally adapted to any Allen Bradley or Modbus-based PLC.)**

The purpose of this document is to provide assistance to the PLC user who wishes to use an EtherMeter for pulse-based batching operations. This document assumes that the user is well-versed in PLC's and a ladder-logic programming environment.

In this example, the PLC and EtherMeter are connected via Ethernet. In this example, EtherNet/IP is the connection protocol, although Modbus/TCP could also be used with a compatible PLC.

When creating an EtherNet/IP client/server connection between a MicroLogix PLC (client) and an EtherMeter™ (server), no special setup is generally required within the EtherMeter's Setup Menu. The EtherMeter features an "always-on" EtherNet/IP server on TCP logical port 44818; and it is configured to auto-detect and service incoming client requests from Allen Bradley PLC's. For other PLC brands, the EtherMeter features an "always-on" Modbus/TCP server on TCP logical port 502.

To simplify integration into an Allen Bradley network, the EtherMeter emulates a MicroLogix/SLC5xx.

In this demonstration, the Allen Bradley PLC used was a Model MicroLogix 1400.

### 1. Wiring Configuration.

In this example, the Ethernet ports of the MicroLogix and EtherMeter are wired directly to each other, without an Ethernet switch, using a single Ethernet crossover cable. Alternatively, the Ethernet ports of both the MicroLogix and EtherMeter could be connected to a common Ethernet switch if expanded network connectivity is desired. For reference, the wiring and hardware configuration is illustrated in Figure 1.

### 2. EtherMeter Configuration.

The EtherMeter was configured for two (2) pulse-based meters with the following setup commands:

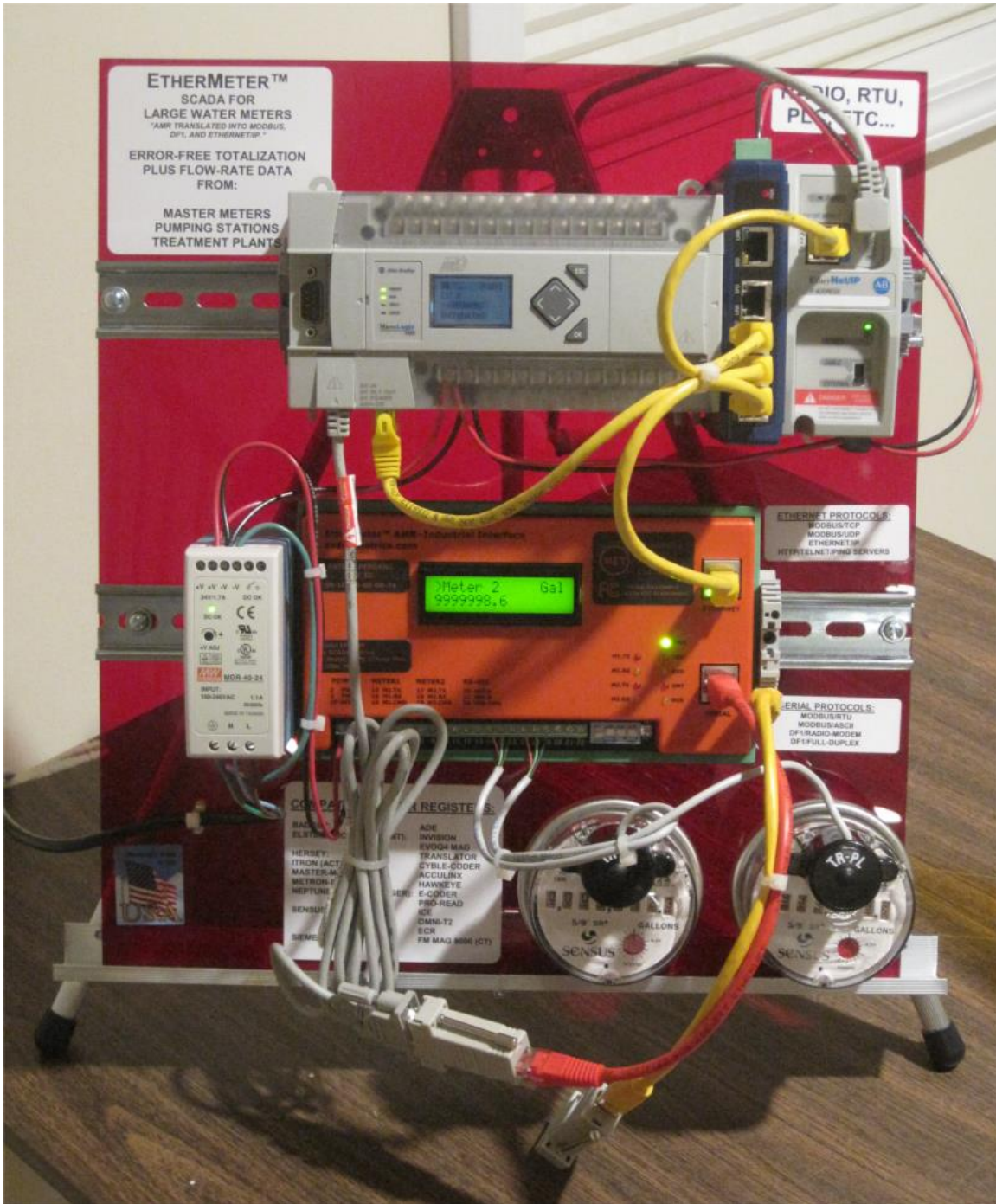
```
SET  TYPE1  PULSE
SET  TYPE2  PULSE
```

**...or for legacy firmwares dated \*before 08 Nov 2020 ...**

```
SET  PWR1   0
SET  PWR2   0
```

### 3. Meter Registers.

The Sensus encoder-type "ICE" registers shown in Figure 1 were replaced with two (2) pulse-based meters.



**Figure 1. MicroLogix EtherNet/IP Communication Demonstration Panel. MicroLogix 1400 PLC / EtherMeter Hardware and Wiring Configuration.**

**Note(s):**

- 1. The MicroLogix and EtherMeter are both powered by a single AC/DC Converter (24VDC output).**

4. RsLogix Message Instruction Configuration: Read Meter Totalizations and Flow Rates

In the ladder logic fragment illustrated in Figure 2, a timer-initiated message instruction is performed to read meter totalization and flow rate data from two (2) attached meters. When using EtherNet/IP, the data is read from the EtherMeter's N7:0...7 registers; and the data is later transferred into the PLC's long-integer registers for future mathematical manipulations. (For Modbus/TCP, the data would be read from the EtherMeter's 40001...40008 registers.) The setup of the EtherNet/IP message instruction is illustrated in Figures 3a and 3b.

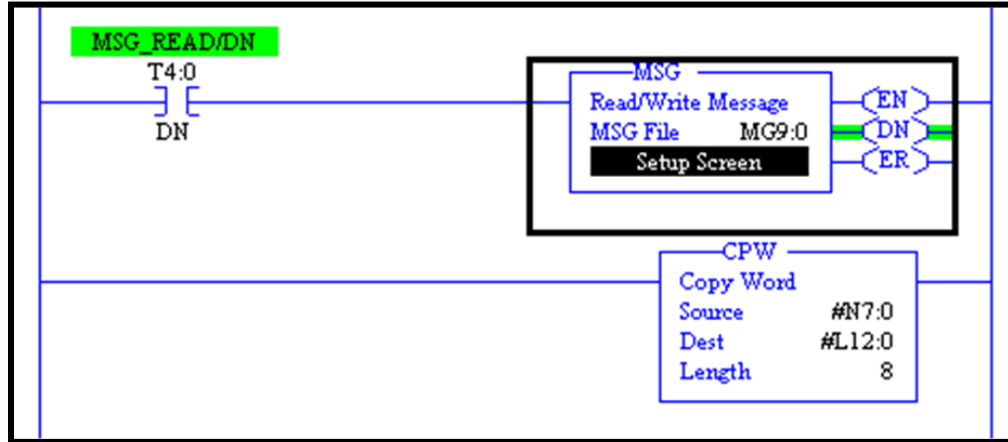


Figure 2. Ladder Logic Fragment: Polling An EtherMeter For Data.

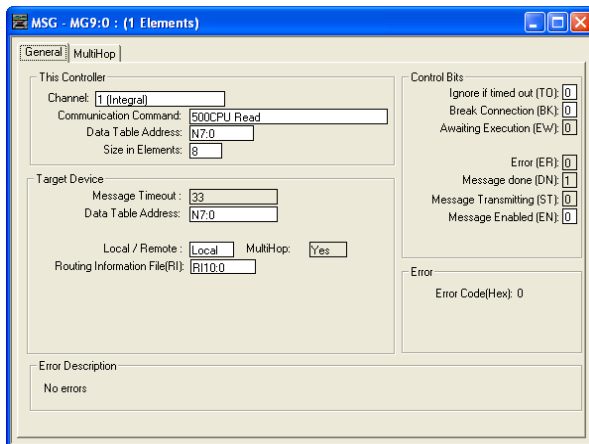


Figure 3a. Message Instruction Details: EtherMeter Totalization & Flow Rate Data

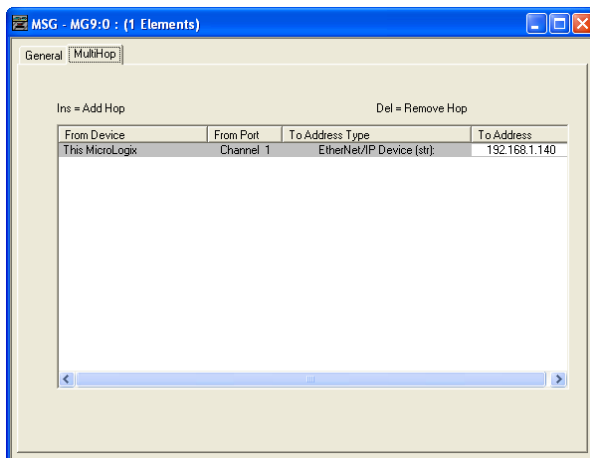


Figure 3b. Message Instruction Details: EtherMeter Totalization & Flow Rate Data

5. RsLogix Message Instruction Configuration: Reset Pulse-Based Totalization

In the ladder logic fragment illustrated in Figure 4, a timer-initiated message instruction is performed to write a 16-bit word to the attached EtherMeter's B10:0 file. (For Modbus, the "Write Single/Multiple Coil" instruction should be used.) The source word on the MicroLogix in this example is N18:0. The setup details of the message instruction are illustrated in Figures 5a and 5b.

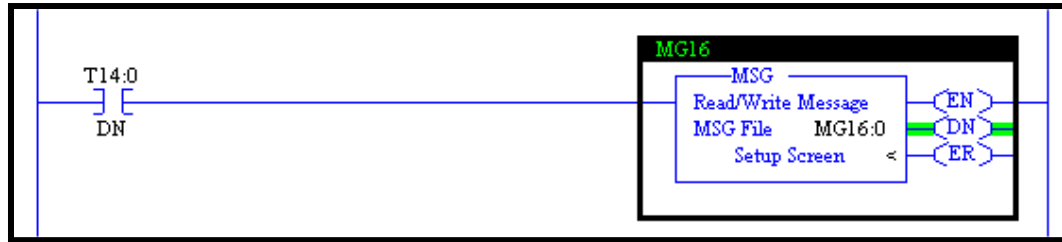


Figure 4. Ladder Logic Fragment: Writing To Coils On An EtherMeter

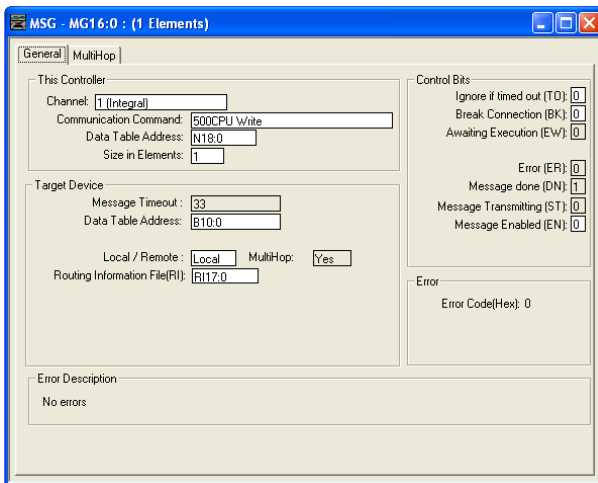


Figure 5a. Message Instruction Details: Writing To Coils On An EtherMeter

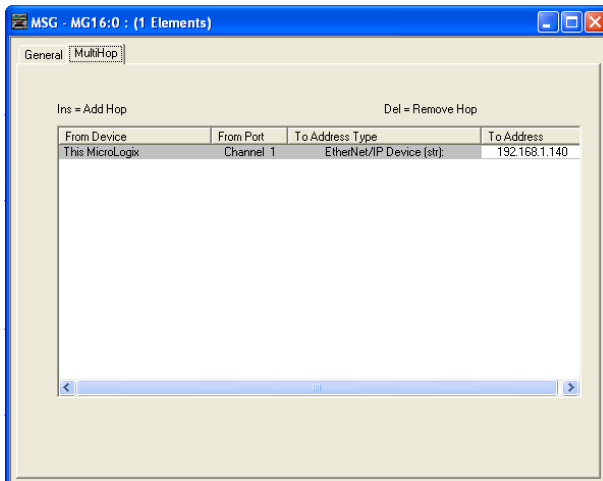


Figure 5b. Message Instruction Details: Writing To Coils On An EtherMeter

6. Writing To The EtherMeter Coils.

In this example, the N18:0 word within the MicroLogix was manually edited to achieve the desired results:

MicroLogix Memory	EtherMeter Memory	Result
N18:0/0 = 1 N18:0/0 = 0	B10:0/0 = 1 B10:0/0 = 0	Aux Digital Output 1 = ON = OFF
N18:0/1 = 1 N18:0/1 = 0	B10:0/1 = 1 B10:0/1 = 0	Aux Digital Output 2 = ON = OFF
N18:0/8 = 1	B10:0/8 = 1	Reset Pulse-Based CNT1 to ZERO
N18:0/9 = 1	B10:0/9 = 1	Reset Pulse-Based CNT2 to ZERO

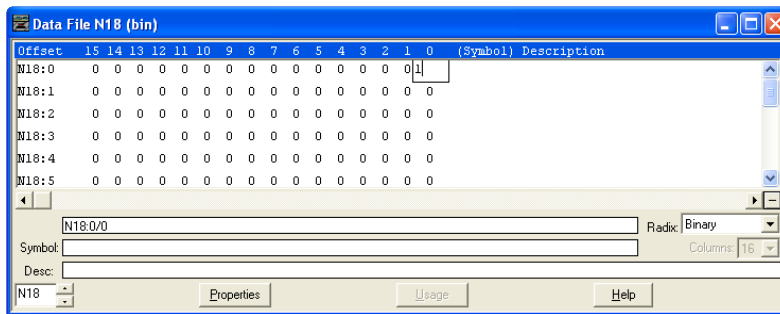


Figure 6. Write a '1' to B10:0/0:

EtherMeter Aux Digital Output 1 = ON.

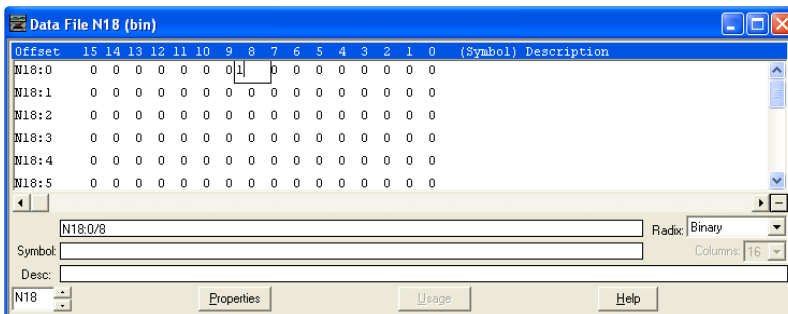


Figure 7. Write a '1' to B10:0/8:

Reset EtherMeter CNT1 to ZERO.

Alternatively, from a Modbus/TCP PLC, the "Write Single/Multiple Coil" instruction should be used:

EtherMeter Coil	Result
00001 = 1 00001 = 0	Aux Digital Output 1 = ON = OFF
00002 = 1 00002 = 0	Aux Digital Output 2 = ON = OFF
00009 = 1	Reset Pulse-Based CNT1 to ZERO
00010 = 1	Reset Pulse-Based CNT2 to ZERO