



## FieldServer Driver - Serial FS-8700-94 Digital microWATT Plus™<sup>1</sup>

### Description

The Digital microWATT Plus™ (DMWPL) Serial Driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 using Digital microWATT Plus™ (DMWPL) protocol. The FieldServer can emulate a Server or Client although Server functionality is limited. The following is a summary of the DMWPL driver's abilities.

This driver is capable of operating as a

- Client – Standard Operation
- Client – Application Specific Operation
- Server – Standard Operation

### Standard Operation:

Operates as a standard FieldServer Driver with customizable read/write capability. The customer can determine what data is read/written and where the data is stored. Using the configuration file the customer can map data between the DMWPL device and other vendor's remote devices connected using a different protocol

### Application Specific Operation (ASO):

The driver requires minimal configuration. The customer specifies connections and nodes and the driver configures itself, automatically mapping data between the ASO driver and another protocol according to a prearranged mapping

### Operation as a Standard Client

The driver can request "Status" and "Power" information from a remote DMWPL device, as well as send several commands including control mode commands and set points.

Under the "flat" data option, the FieldServer requests data and records the response messages byte-for-byte into FieldServer data arrays. Under the "collated" data option, the data messages returned by the DMWPL devices are recorded in a Present Value array and an Out of Service array. The Present Value array contains summary data about the status of DMWPL devices, and the Out of Service array contains information about DMWPL device and server function availability.

---

<sup>1</sup> Digital microWATT / Digital microWATT Plus / Lutron are all trademarks of Lutron Electronics Co., Inc



The collated data is presented in the FieldServer's Data Arrays at a fixed location which is determined by the unit address and the data being stored. A table in the Driver Manual lists the array location of each data value collated by the FieldServer.

The driver can poll, on each serial port, for any number of DMWPL devices connected to any number of routers provided that the unit and router ID's are legal in terms of the addressing rules of the DMWPL.

Scaling the incoming data before it is stored in the Data Arrays is not available for this driver. An object oriented BACnet driver enhancement may include scaling.

When a node is offline the driver will set the Out of Service state to true for all elements of the Out of Service data array.

**Application Specific Operation (ASO) as a Client**

Application Specific Operation (ASO) of this driver maps data read by this driver acting as a client to a series of complex data objects which are presented as BACnet (IP or Ethernet) objects using the FieldServer's server capabilities.

A BACnet (IP or Ethernet) client can browse the FieldServer for BACnet Objects which contain the data read from the DMWPL and can set objects Out of Service, write set points and activate modes on the DMWPL devices by writing to these BACnet objects.

The driver can be configured to create a single BACnet node to serve data for all the DMWPL nodes or to create a virtual BACnet node for each DMWPL node.

**Operation as a Standard Server**

The driver can emulate any number of legally addressed units. The emulated DMWPL units can be connected to a specific port on the FieldServer or can be generalized across any port.

Commands are used to change operating states; changed set points have the effect of changing data in the FieldServer's Data Arrays but have no operational/physical effect. No validation is performed on the values/set points sent by remote devices.

<b>Fieldserver Mode</b>	<b>Nodes</b>	<b>Comments</b>
Client	1	Only 1 client node allowed on Multidrop systems
Server	63	Limit is per port.

**Formal Driver Type**

Serial  
Client or Server



## Protocol Driver



A Sierra Monitor Company

### Connection Information

**Connection type:** RS-232 (with converter) or RS-485  
The driver supports all standard baud rates: 110; 300; 600; 1200; 2400; 4800; 9600; 19200; 28800; 38400; 57600; 115200 Baud

**Baud Rates:** The vendor equipment supports baud rates of 38400 from FieldServer to router; and 9600 from FieldServer directly to DMWPL device

**Data Bits:** Driver Supports: 7,8  
Vendor Equipment Supports: 8

**Stop Bits:** Driver Supports: 1,2  
Vendor Equipment Supports: 1

**Parity:** Driver Supports: Odd, Even, None  
Vendor Equipment Supports: None

**Multidrop Capability** Yes

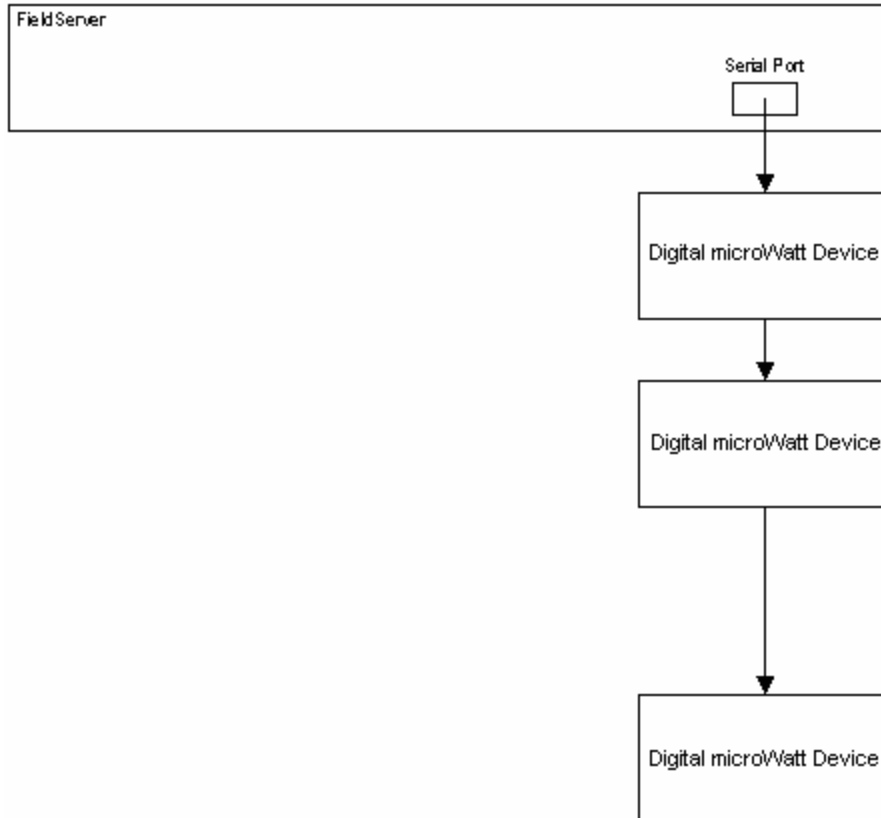
### Devices tested

Device	Tested (FACTORY, SITE)
DMW-RP3	-Site



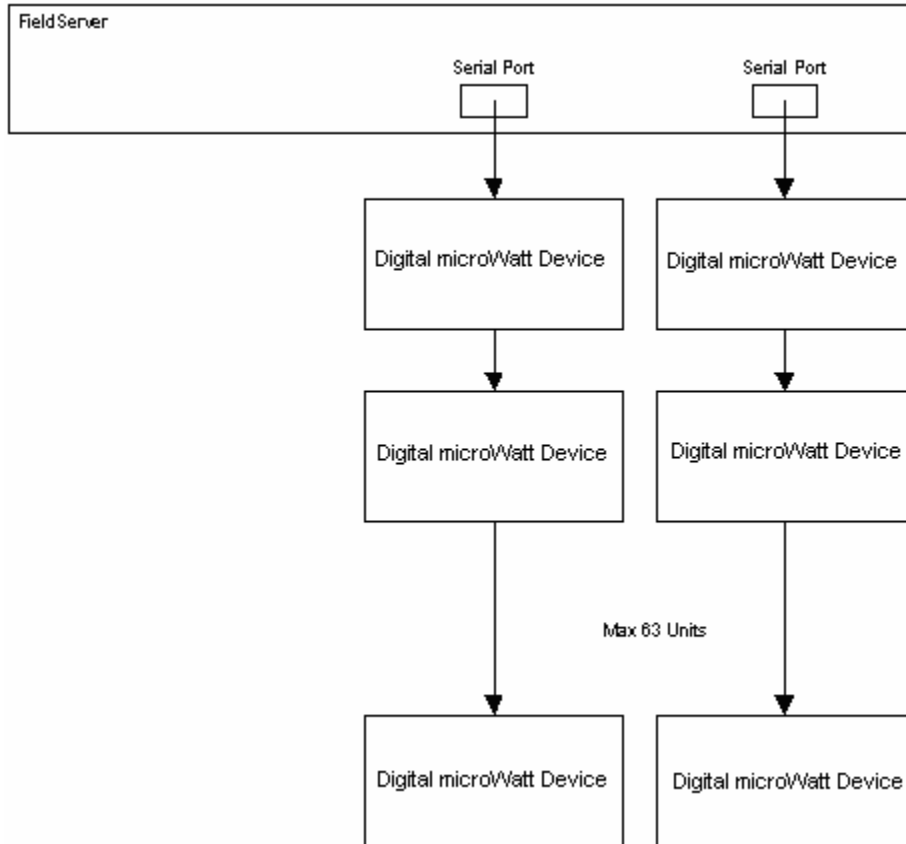
**Connection configurations**

Connection Style 1 – Use a single serial port on the FieldServer to poll a max of 63 DMWPL devices connected in series. Uses router address zero.



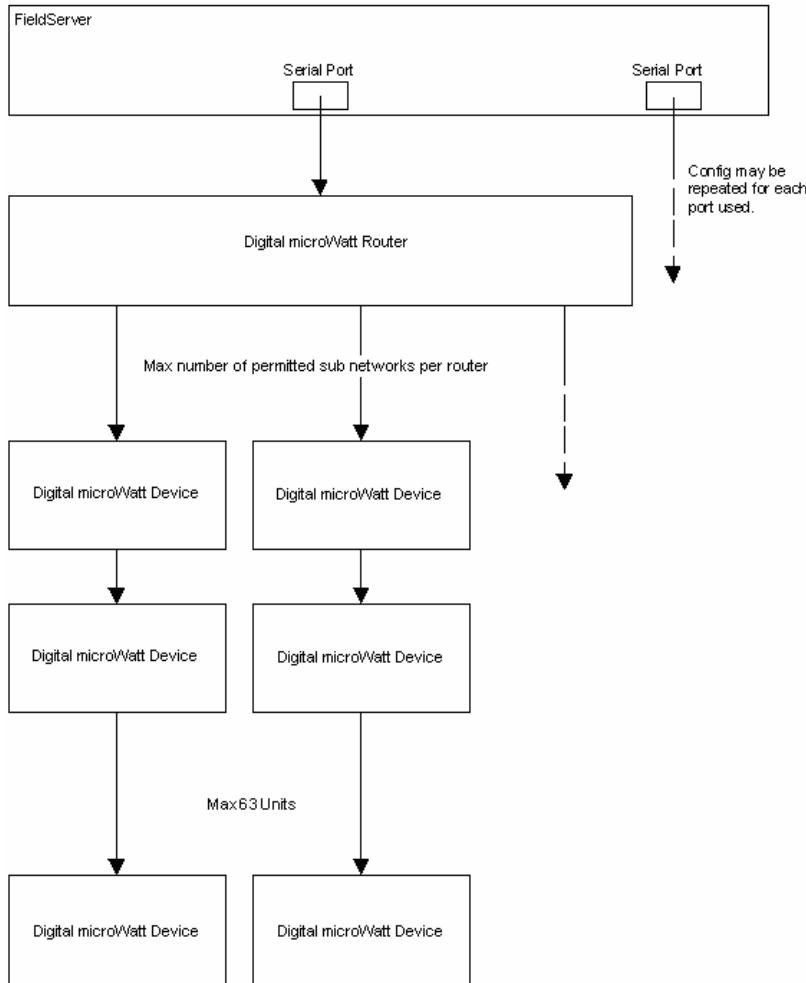


Connection Style 2 – Use multiple serial port on the FieldServer to poll a max of 63 DMWPL devices connected in series on each port. Uses router address zero.





Connection Style 3 – Use (multiple) serial port(s) on the FieldServer to connect to DMW Routers. Each router may be connected to a maximum of 3 sub networks with a max of 63 DMWPL devices.



**Connection Notes**

Cable, distance and baud rate choices and settings to comply with manufacturers requirements for connecting DMW devices and routers. Refer to *Digital microWATT Plus™ Hardware Installers Guide* and <http://www.lutron.com>

**Determining Router Number:**

Router Number = ((Router DIP switch address) \* 4) + Router Link Number  
Thus: a router with DIP switch address 1 and link number 2 would be = (1\*4) +2 = 6



If no router is being used (the DMW controllers are connected directly to the FieldServer), the Router number should be set to 0.

**Communications functions - Supported functions at a glance:**

Command Description	Function	Client can send	Server Response	Server can Parse
Afterhours mode Activate	Write	Yes		Yes
Afterhours mode Cancel	Write	Yes		Yes
Emergency mode Activate	Write	Yes		Yes
Emergency mode Cancel	Write	Yes		Yes
Burn-In mode Activate	Write	Yes		Yes
Burn-In mode Cancel	Write	Yes		Yes
Normal mode Activate	Write	Yes		Yes
Set Max Light Limit	Write	Yes		Yes
Cancel Max Light Limit	Write	Yes		Yes
Set Light Level	Write	Yes		Yes
Set Light Level to Off	Write	Yes		Yes
Set Load Shed	Write	Yes		Yes
Retrieve STATUS from unit	Read	Yes		Yes
Retrieve POWER from unit	Read	Yes		Yes
Wall Control Enable	Write	Yes		Yes
Wall Control Disable	Write	Yes		Yes
Occupant Sensor Enable	Write	Yes		Yes
Occupant Sensor Disable	Write	Yes		Yes
Photo Sensor Enable	Write	Yes		Yes
Photo Sensor Disable	Write	Yes		Yes
Reset Energy Counter	Write	Yes		Yes
Reset Lamp Hours Counter	Write	Yes		Yes
Ack / Nak	Response		Yes	N/A
Status	Response		Yes	N/A
Power	Response		Yes	N/A

**Data Locations**

When the driver is configured as a client, it can request and store data into different array locations that are fixed in size. Currently the storage options are "flat", and "collated".

Flat storage copies incoming STATUS and POWER messages from the DMWPL device byte-for-byte including the message terminators. The STATUS data message is 68 bytes long and the POWER data message is 30 bytes long.

The collated storage method is semi-object oriented. It interprets the STATUS and POWER messages and records specific information within two arrays: the present value (PV) array, and



the missing-in-action array (MIA). There are 20 PV's that this method records, and a corresponding number of MIA status members. If a read request from the FieldServer results in an MIA response, then all 20 members of the MIA array will be set to node offline or missing. Other logic specific the 20 PV elements may also affect specific MIA states.

As a client, the driver can also change DMWPL settings and status through write map descriptors which issue write commands.

A separate Map Descriptor is required for each router (including router zero). This map descriptor will define the base storage location to be used to store the data for unit 1 on that router; all other units will be stored relative to this base offset.

On the server side all messages are stored as "flat" arrays. An additional array is used to summarize STATE information that is manipulated by incoming write commands. The following table identifies the parameters stored in the server side STATE array. The flat arrays store data into the POWER and STATUS arrays in the same structure as response messages. See DMWPL communication protocol for details on this message structure.

### Server Side "State" Array

Function	Accepted values	Storage offset	Comments
Afterhours Mode	0/1	0	
Emergency Mode	0/1	1	
Burn-in Mode	0/1	2	
Normal Mode	0/1	3	
Set High End	0-99	4	function disabled
Set Low End	0-99	5	function disabled
Max Light Limit	0-99	6	
Set Light Level	0-99	7	
Set Load Shed	0-99	8	
Wall Control	0/1	9	
Occupant Sensor Enable	0/1	10	
Photo Sensor Enable	0/1	11	



For collated data storage, each device polled will have the following data stored. The offsets are relative to the base offset for the device. The base offset is based on the unit address.

	Data Array <sup>2</sup>	1	2
	Data Array Elements Used	1	1
		Data Array Offset	
Obj Id.	Description	Present_Value	Out_of_Service (Missing in Action)
1	Relay State	0	0
2	Light Control	1	1
3	Operating State	2	2
4	Load Shed	3	3
5	Max Light Limit	4	4
6	Wall Control	5	5
7	Wall Control En/Disable	6	6
8	Occupancy	7	7
9	Occupant Sensor En/Disable	8	8
10	Photosensor Cutback	9	9
11	Photosensor Cutback En/Disable	10	10
12	Power	11	11
13	Energy	12	12
14	Burn-in Hours	13	13
15	Lamp Hours	14	14
16	Timing State	15	15
17	Time Remaining	16	16
18	Current Light Level	17	17
19	Timing State (Analog)	18	18
20	Operating State (Analog)	19	19

**Functions Described**

The user will be able to configure Write tasks by creating Map Descriptors which define the command function to be sent to the DMWPL units.

**Unsupported Devices or Protocol Options**

Device	Details
All	The command functions <i>Set High End</i> and <i>Set Low End</i> are not implemented.

**Driver Limitations and Exclusions**

When MIA becomes true, the present value will hold the last valid present value.

<sup>2</sup> Data is stored in two Data Arrays



## Protocol Driver



A Sierra Monitor Company

The driver server side provides a limited emulation of a DMWPL device. The driver does not emulate the response of a router.



## Protocol Driver



A Sierra Monitor Company

THIS PAGE INTENTIONALLY LEFT BLANK