

EZRead Active-X Meter Communication & Control Module

The EZCom class of the EZRead.exe module can be used in conjunction with the EZMeter.mdb database to read and set the EZ Meter Plus series of power meters. The five properties in the table below are the only ones needed to implement a simple meter system within a property management system. In such a case, the user would use the EZReadIt Gold program to maintain the Access database and set the meters. Software developers wanting to implement other features of the EZReadIT program within their own program may use any of the other exposed properties and methods

Name	Type	Parameter sent
PathName	String	Send path to EZMeter.mdb database. If database is in same folder, us "" as path. Otherwise enter drive and path ending with a \.
Location	String	Send location name as ended in the EZMeter.mdb database.
Connected	Long	Facility number to connect to. This will usually be a 1, but may have a different value if different ports are used to connect to different meters. The value is the ID number of the facility. Set to 0 to disconnect and close serial port. Closing instance of EZCom object will also close serial port.
KWH	Double	Gets and returns current reading of the meter at Location
ErrMsg	String	Returns "OK" if the last attempted meter read was successful. Otherwise, says what the problem was.

Visual Basic code example:

```
Public ez as EZReader.EZCom    \' Declare global variable

Sub Form_Load()              \' Set up the EZCom server

    Set ez = New EZReader.EZCom \' Create an instance of the com server
    ez.PathName = ""          \' set path to EZMeter.mdb
    ez.Connected = 1          \' Facility ID number from Facilities table. This
                              \' will open comport and connect with modem if
                              \' necessary.

End Sub

Private Function ReadMeter()
    ez.Location = vLocationName \' vLocationName is your variable that matches
                              \' the Location name in the EZMeter.mdb database

    ReadMeter = ez.KWH         \' Returns the value of the meter in kilowatt
                              \' hours

End Function
```

Items in the next table are properties that you can set in the EZCom object. Where appropriate, setting the property in the EZCom object will cause it to set that value in the meter. Use the ErrMsg property to verify the value was set properly.

Name	Type	Parameter to send
SerialNo	Long	Serial number of meter to read, set, etc.
Display1	Double	Value to set in meter for DISP1 counter
Display2	Double	Value to set in meter for DISP2 or DISPL counter
Phase3	Double	Value to be set in the phase 3 register
Pulse1	Double	Value to set in Pulse 1 register
Pulse2	Double	Value to set in Pulse 2 register
Pulse3	Double	Value to set in Pulse 3 register
Log	Byte	0= none,1=DISP1, 2-DISP2, 3=DISPL, 4=BOTH; Determines what is written to log file.
IOSet	String	Set IO Aux status (message type 0BH)
OptionBytes	String	String to set in Option Byte register
Retries	Long	Times to send message without reply before timeout
Wheels	Long	number of digits(wheels) in display counter
Decimals	Long	number of digits to right of decimal point in display
Mult(a,b)	Long, Long	1, 2 or 3 for IO1, IO2 or IO3 respectively; Number of pulses from other meter for each billable unit.
Rollover(a,b)	Long, Long	1, 2 or 3 for IO1, IO2 or IO3 respectively; Number of digits (left of decimal point if any) in pulse meter connected to this port.
TimeOut	Long	Milliseconds to wait after sending message to wait before resending request.

If you want to log all three phases on a three phase meter, you must have a dual counter meter such as the 4320 (Olympic). Phase one is on DISP1 while DISP2 contains the sum of phases two and three. Phase two is DISP2 minus Phase3. One single counter meters, all three phases are on DISPL and there is no way to separate phase one from the rest.

This table shows the properties that allow you to get data from the meters or the EZCom object

Name	Type	Reply
Location	String	Location name
Log	Byte	Log code. See above.
Connected	Long	Facility number connected to. 0 if not connected
SerialNo	Long	Serial number of active meter
Display1	Double	Value of DISP1
Display2	Double	Value of DISP2 or DISPL
Both	Double	Total of DISP1 and DISP2
Phase3	Double	Value in Phase 3 register
Pulse1	Double	Value in Pulse1 register
Pulse2	Double	Value in Pulse2 register
Pulse3	Double	Value in Pulse3 register
ModeByte	Long	Value of Mode Byte (contains IO port status, etc)
EModeByte	Long	Value of EMode Byte
MetVariant	String	Meter Variant (Reply from a message type 09H)
OptionBytes	String	Returns the meter model number
AuxIO	Long	For future use
ErrMsg	String	Message returned after message to meter
Retries	Long	How many times service will attempt to contact meter
Wheels	Long	Replies with number of digits in counter.
Decimals	Long	Number of digits to right of decimal point in counter
Mult(a,b)	(a) Long, (b) Long	(a) 1, 2 or 3 for IO1, IO2 or IO3 respectively; (b) Number of pulses from other meter for each billable unit.
Rollover(a,b)	(a) Long, (b) Long	(a) 1, 2 or 3 for IO1, IO2 or IO3 respectively; (b) Number of digits (left of decimal point if any) in pulse meter connected to this port.
TimeOut	Long	milliseconds without answer before request is resent
Tries	Long	number of times last message was sent
Request	String	String of decimal values of bytes sent and received in last message exchange
Answer	String	

The following are methods exposed in the EZCom communication server.

Method	Purpose
GetPulse() (subroutine)	Causes com server to read pulse values from the selected meter. You must call this sub before getting any of the Pulse1, Pulse2 or Pulse3 properties.
ResetMeter (subroutine)	Sends type 01H message to reset meter and resets any error flags within meter.
WhoRU (subroutine)	Sends a Who Are You message (type 055H). Get answer with SerNo. Works only if only one meter is jumpered in Test Mode.
KillMe (subroutine)	Closes serial port