==========
========== EKM ModBus ==========
==========

The EKM Omnimeter Pulse v.4 and Omnimeter Pulse v.4 UL both have the ability to be enabled by EKM to communicate using the ModBus protocol (described below) rather than the default EKM protocol which is described here: https://documents.ekmmetering.com/api-docs/rs-485-communications.html#rs-485-communications

We feel that there are many advantages to using the EKM protocol over this ModBus protocol. One big advantage is that the EKM protocol meters are what we know best and what we have in stock. Also, the EKM protocol is what is required for the EKM Push data collection system. The ModBus meters have a maximum of only 254 possible meter addresses whereas the EKM protocol meters have 999,999,999,998 possible meter addresses which makes for a much more scalable system.

Prototyping Software to read your EKM Omnimeter Pulse v.4 which has been ModBus enabled: https://documents.ekmmetering.com/EKM_Modbus_App_For_Customers.zip

Omnimeter Pulse v.4 meters on which ModBus is enabled will be shipped to you preset to ModBus address 5, unless a different address is requested.

Note:
Byte counts are shown in brackets.
ModBus words are two bytes.

ModBus messages (both Request and Response) have the form:
DeviceAddress[1], FunctionCode[1], Parameters[variable], CRC[2]

Three FunctionCodes are currently supported:

Func 0x03 = ReadHoldingReg to read "Read-Write registers"
Func 0x04 = ReadInputReg to read "Read-only registers"
Request = DevAddr[1], FunctionCode[1], 1stAddr[2], wordCount[2], CRC[2]
Response = DevAddr[1], FunctionCode[1], byteCnt[1], data[byteCnt], CRC[2]

Func 0x10 = WriteMultiReg to write "Read-Write registers"
Request = DevAddr[1], FunctionCode[1], 1stAddr[2], wordCnt[2], dataByteCnt[1], data[2*wordCnt], CRC[2]
Response = DevAddr[1], FunctionCode[1], 1stAddr[2], wordCnt[2], CRC[2]

Failure response for all functions:
Response = DevAddr[1], (FunctionCode | 0x80)[1], ErrorCode[1], CRC[2]

Possible ModBus error codes:
MBERR1_ILLEGAL_FUNCTION 1
MBERR2_ILLEGAL_ADDR_VALUE 2
MBERR3_ILLEGAL_DATA_VALUE 3
MBERR5_ACKNOWLEDGE 5

Register Address list for Values matching WattNode.

RegAddr	R/W	WordCnt	Decimal Pl.	Units	Name
1009	R	2	0	watts	Total Watts
1011	R	2	0	watts	Watts Line 1
1013	R	2	0	watts	Watts Line 2
1015	R	2	0	watts	Watts Line 3
1033	R	1	2	Hz	Frequency (Hz * 100)
1147	R	2	0	VAR	VAR Total (Volts Amps Reactive)
1149	R	2	0	VAR	VAR Line 1
1151	R	2	0	VAR	VAR Line 2
1153	R	2	0	VAR	VAR Line 3
1163	R	2	1	amps	Current Line 1 (Amps RMS)
1165	R	2	1	amps	Current Line 2 (Amps RMS)
1167	R	2	1	amps	Current Line 3 (Amps RMS)
1173	R/W	2	0	watts	Max Demand (RMS Watts * 10)
1201	R/W	2	0,1,2	kWh	Resettable Total kWh (kWh * 100)
1205	R	2	0,1,2	kWh	Total kWh (kWh * 100)
1214	R	1	1	volts	Voltage L1 (Volts RMS * 10)
1215	R	1	1	volts	Voltage L2 (Volts RMS * 10)
1216	R	1	1	volts	Voltage L3 (Volts RMS * 10)
1301	R	2	0,1,2	kWh	Total L1 kWh (kWh * 100)
1303	R	2	0,1,2	kWh	Total L2 kWh (kWh * 100)
1305	R	2	0,1,2	kWh	Total L3 kWh (kWh * 100)
1313	R/W	2	0,1,2	kWh	Resettable Total Reverse kWh (kWh * 100)
1315	R	2	0,1,2	kWh	Total Reverse kWh (kWh * 100)
1317	R	2	0,1,2	kWh	L1 Reverse kWh (kWh * 100)
1319	R	2	0,1,2	kWh	L2 Reverse kWh (kWh * 100)
1321	R	2	0,1,2	kWh	L3 Reverse kWh (kWh * 100)
1323	R	2	0,1,2	kWh	Total Reactive kWh (kWh * 100)
1340	R	1	2	cosθ	Cosine Power Factor L1
1341	R	1	2	cosθ	Cosine Power Factor L2
1342	R	1	2	cosθ	Cosine Power Factor L3

Register Address list for EKM custom values (not like WattNode)

RegAddr	R/W	WordCnt	Name
1500	R	1	In1/In2/In3 Pin State Bit-map
1501	R/W	1	In1 Pulse Ratio
1502	R/W	1	In2 Pulse Ratio
1503	R/W	1	In3 Pulse Ratio
1504	R	2	In1 Pulse Count
1506	R	2	In2 Pulse Count
1508	R	2	In3 Pulse Count
1510	R/W	2	Resettable In1 Raw Pulse Count
1512	R/W	2	Resettable In2 Raw Pulse Count
1514	R/W	2	Resettable In3 Raw Pulse Count
(on the next two you Read 1 word and Write 2 words 1st word is the State, 2nd word is the hold seconds)			
1516	R/W	1/2	I/O Pin 1 State(read/write) hold seconds(write)
1518	R/W	1/2	I/O Pin 2 State(read/write) hold seconds(write)
1520	W	2	Apply Password (apply it; allows a following Write)
1522	W	2	Write New Password (changes it)
1526	W	1	BAUD rate
1527	R/W	1	ModBus Reply Delay
1528	R/W	1	ModBus Password Disable
1530	R	3	Serial Number (all 12 decimal digits)
1540	R/W	2	UserData (2 words)
1542	-	2	reserve for extra UserData later
1544	R/W	-	ModBus_userMap (Write with 16, Read with 03, Apply with 04) variable length; consists of [pairCnt(2), regAddr/wordCnt pairs(4 per pair)]
1546	R/W	-	LcdItemTable (Write with 16, Read with 03) variable length; consists of bytes in LcdItemTable (which are LCD item numbers)
1548	R	1	OneWire Port_1 (temperature as Celsius*10 if DS18B20 is found)
1549	R	1	OneWire Port_2 (temperature as Celsius*10 if DS18B20 is found)
1550	R/W	2	Resettable L1 Energy - Hi Resolution (kWh * 10000)
1552	R/W	2	Resettable L2 Energy - Hi Resolution (kWh * 10000)
1554	R/W	2	Resettable L3 Energy - Hi Resolution (kWh * 10000)
1556	R/W	4	8-byte Date/Time (RTC)
1708	R	1	Firmware Version
1800	R	125	ReadABC (All Registers)

==========

The following examples use a DevID of 05 (as set by the command to enable ModBus). To change device's ModBus Address see register 1652(0x0674) below.

========== Read-only Regs ==========
==========
// Read
// ID[1],func[1],1stAddr[2],regCount[2],chkSum[2]

Dummy Register - Reg addr 0001(0x0001)
Read 4 bytes of zeros (to indicate device presence):
0x01 0x04 0x00 0x01 0x00 0x02 0x20 0x0B addr=1
0x05 0x04 0x00 0x01 0x00 0x02 0x21 0x8F addr=5
- Use device ID 254(0xFE) to get a response from all devices on the bus:
0xFE 0x04 0x00 0x01 0x00 0x02 0x34 0x04 addr=0xFE
Devices will respond with their true dev ID in the first byte.
RESPONSE:
DevId[1], FunctionCode[1], ByteCount[1], data[ByteCount], CRC[2]
DevId[1], 0x04, 4, 0, 0, 0, 0, CRC[2]

Total Watts - Reg addr 1009(0x3F1)
Note: Reg addr 1009 is the decimal version of the register address and 0x3F1 is the hex version of the register address (1009 decimal converted to hex = 0x3F1)
Read 4 bytes Watts:
0x05 0x04 **0x03 0xF1** 0x00 0x02 0x21 0xF8

Watts Line 1 - Reg addr 1011(0x3F3)
Read 4 bytes Watts:
0x05 0x04 0x03 0xF3 0x00 0x02 0x80 0x38

Watts Line 2 - Reg addr 1013(0x3F5)
Read 4 bytes Watts:
0x05 0x04 0x03 0xF5 0x00 0x02 0x60 0x39

Watts Line 3 - Reg addr 1015(0x3F7)
Read 4 bytes Watts:
0x05 0x04 0x03 0xF7 0x00 0x02 0xC1 0xF9

Frequency - Reg addr 1033(0x409)
Read 2 bytes Hz * 100:
0x05 0x04 0x04 0x09 0x00 0x01 0xE1 0x7C

VAR Total - Reg addr 1147(0x47B)
Read 4 bytes VA:
0x05 0x04 0x04 0x7B 0x00 0x02 0x01 0x66

VAR Line 1 - Reg addr 1149(0x47D)
Read 4 bytes VA:
0x05 0x04 0x04 0x7D 0x00 0x02 0xE1 0x67

VAR Line 2 - Reg addr 1151(0x47F)
Read 4 bytes VA:

0x05 0x04 0x04 0x7F 0x00 0x02 0x40 0xA7

VAR Line 3 - Reg addr 1153(0x481)
Read 4 bytes VA:
0x05 0x04 0x04 0x81 0x00 0x02 0x21 0x57

Current Line 1 - Reg addr 1163(0x48B)
Read 4 bytes Amps RMS:
0x05 0x04 0x04 0x8B 0x00 0x02 0x01 0x55

Current Line 2 - Reg addr 1165(0x48D)
Read 4 bytes Amps RMS:
0x05 0x04 0x04 0x8D 0x00 0x02 0xE1 0x54

Current Line 3 - Reg addr 1167(0x48F)
Read 4 bytes Amps RMS:
0x05 0x04 0x04 0x8F 0x00 0x02 0x40 0x94

Total kWh - Reg addr 1205(0x4B5)
Read 4 bytes kWh * 100:
0x05 0x04 0x04 0xB5 0x00 0x02 0x60 0x99

Voltage Line 1 - Reg addr 1214(0x4BE)
Read 2 bytes Volts RMS * 10:
0x05 0x04 0x04 0xBE 0x00 0x01 0x51 0x5A (Volts L1)
Read 4 bytes with two Volts RMS * 10:
0x05 0x04 0x04 0xBE 0x00 0x02 0x11 0x5B (volts L1 and L2)
Read 6 bytes with three Volts RMS * 10:
0x05 0x04 0x04 0xBE 0x00 0x03 0xD0 0x9B (volts L1, L2 and L3)

Voltage Line 2 - Reg addr 1215(0x4BF)
Read 2 bytes Volts RMS * 10:
0x05 0x04 0x04 0xBF 0x00 0x01 0x00 0x9A

Voltage Line 3 - Reg addr 1216(0x4C0)
Read 2 bytes Volts RMS * 10:
0x05 0x04 0x04 0xC0 0x00 0x01 0x31 0x42

Total Line 1 kWh - Reg addr 1301(0x515)
Read 4 bytes kWh * 100:
0x05 0x04 0x05 0x15 0x00 0x02 0x61 0x47 (kWh L1)
Read 8 bytes with two kWh * 100:
0x05 0x04 0x05 0x15 0x00 0x04 0xE1 0x45 (kWh L1 and L2)
Read 12 bytes with three kWh * 100:
0x05 0x04 0x05 0x15 0x00 0x06 0x60 0x84 (kWh L1, L2 and L3)

Total Line 2 kWh - Reg addr 1303(0x517)
Read 4 bytes kWh * 100:
0x05 0x04 0x05 0x17 0x00 0x02 0xC0 0x87

Total Line 3 kWh - Reg addr 1305(0x519)
Read 4 bytes kWh * 100:
0x05 0x04 0x05 0x19 0x00 0x02 0xA1 0x44

Total Reverse kWh - Reg addr 1315(0x523)
Read 4 bytes kWh * 100:
0x05 0x04 0x05 0x23 0x00 0x02 0x81 0x49

Reverse kWh Line 1- Reg addr 1317(0x525)
Read 4 bytes kWh * 100:
0x05 0x04 0x05 0x25 0x00 0x02 0x61 0x48

Reverse kWh Line 2- Reg addr 1319(0x527)
Read 4 bytes kWh * 100:
0x05 0x04 0x05 0x27 0x00 0x02 0xC0 0x88

Reverse kWh Line 3- Reg addr 1321(0x529)
Read 4 bytes kWh * 100:
0x05 0x04 0x05 0x29 0x00 0x02 0xA1 0x4B

Total Reactive kWh - Reg addr 1323(0x52B)
Read 4 bytes kWh * 100:
0x05 0x04 0x05 0x2B 0x00 0x02 0x00 0x8B

(The next three each return a signed 16-bit integer giving cosine*100.
0x0000 to 0x0064 (0 to 100) represent a cosine of 0.00 to 1.00
0xFFFF to 0xFF9C (-1 to -99) represent a cosine of -0.01 to -0.99
A cosine of -1.00 is always returned as 1.00.
)
Power Factor Line 1 - Reg addr 1340(0x53C)
Read 2 bytes signed Cosine * 100:
0x05 0x04 0x05 0x3C 0x00 0x01 0xF0 0x8E

Power Factor Line 2 - Reg addr 1341(0x53E)
Read 2 bytes signed Cosine * 100:
0x05 0x04 0x05 0x3D 0x00 0x01 0xA1 0x4E

Power Factor Line 3 - Reg addr 1342(0x53E)
Read 2 bytes signed Cosine * 100:
0x05 0x04 0x05 0x3E 0x00 0x01 0x51 0x4E

In1/In2/In3 Pin State Bit-map - Reg addr 1500(0x5DC)
bits 15:3 reserved,
bit 2 In1 where 0==open, 1==closed/grounded
bit 1 In2 where 0==open, 1==closed/grounded
bit 0 In3 where 0==open, 1==closed/grounded
Read 2 byte bitmap:
0x05 0x04 0x05 0xDC 0x00 0x01 0xF1 0x78

In1 Pulse Count - Reg addr 1504(0x5E0)
Read 4 byte pulse count value:
0x05 0x04 0x05 0xE0 0x00 0x02 0x71 0x75 (In1 pulse count)
Read 12 bytes with three count values:
0x05 0x04 0x05 0xE0 0x00 0x06 0x70 0xB6 (3 4-byte counts In1/In2/In3)

In2 Pulse Count - Reg addr 1506(0x5E2)
Read 4 byte pulse count value:
0x05 0x04 0x05 0xE2 0x00 0x02 0xD0 0xB5

In3 Pulse Count - Reg addr 1508(0x5E4)
Read 4 byte pulse count value:
0x05 0x04 0x05 0xE4 0x00 0x02 0x30 0xB4

Serial Number - Reg addr 1530(0x5FA)
Read 6 bytes, the meter serial number (you convert to 12 decimal digits):
0x05 0x04 0x05 0xFA 0x00 0x03 0x91 0x72

OneWire Port_1 Temperature - Reg addr 1548(0x60C)
Read 2 byte signed Celsius * 10:
0x05 0x04 0x06 0x0C 0x00 0x01 0xF0 0xC5 Port_1
Read 4 bytes with two signed Celsius * 10:
0x05 0x04 0x06 0x0C 0x00 0x02 0xB0 0xC4 Port_1 and Port_2

OneWire Port_2 Temperature - Reg addr 1549(0x60D)
Read 2 byte signed Celsius * 10:
0x05 0x04 0x06 0x0D 0x00 0x01 0xA1 0x05

FIRMWARE_VERSION - Reg addr 1708(0x6AC)
Read 2 byte FIRMWARE_VERSION, Major:Minor
0x01 0x04 0x06 0xAC 0x00 0x01 0xF1 0x63 addr=1
0x05 0x04 0x06 0xAC 0x00 0x01 0xF0 0xE7 addr=5

========== Write Only Registers ==========
====================================
Apply Password - Reg addr 1520(0x5F0)
// (This Write does not alter the password;
// it sets an internal flag to enable a following Write command.
// The flag is cleared after each command or after 5 seconds.)
// The response to Apply Password always indicates success.
// Preceded all other Writes with this "Apply Password" command.
e.g. Apply Password for Dev Addr 1
0x01 0x10 0x05 0xF0 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0xC3 0xBB (00000000)
e.g. Apply Password for Dev Addr 5
0x05 0x10 0x05 0xF0 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0xD6 0x8B (00000000)
0x05 0x10 0x05 0xF0 0x00 0x02 0x04 0x00 0xBC 0x61 0x4E 0xBE 0xCB (12345678)
e.g. Apply Password for Dev Addr 189
0xBD 0x10 0x05 0xF0 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0xAA 0x29 (00000000)
e.g. Apply Password for Dev Addr 247
0xF7 0x10 0x05 0xF0 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0xDE 0x30 (00000000)

Set Password - Reg addr 1522(0x5F2)
Write 4 bytes, new password; 4-byte hex value, 0-99999999 decimal
0x05 0x10 0x05 0xF2 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0x57 0x52 (00000000)
0x05 0x10 0x05 0xF2 0x00 0x02 0x04 0x00 0xBC 0x61 0x4E 0x3F 0x12 (12345678)

Set BAUD - Reg addr 1526(0x5F6)
(give a 0-7 index for 300 to 38400)
There is a 3 second delay after the response before the change happens;
0x05 0x10 0x05 0xF6 0x00 0x01 0x02 0x00 0x00 0xD5 0x06 (300)
0x05 0x10 0x05 0xF6 0x00 0x01 0x02 0x00 0x01 0x14 0xC6 (600)
0x05 0x10 0x05 0xF6 0x00 0x01 0x02 0x00 0x02 0x54 0xC7 (1200)
0x05 0x10 0x05 0xF6 0x00 0x01 0x02 0x00 0x03 0x95 0x07 (2400)
0x05 0x10 0x05 0xF6 0x00 0x01 0x02 0x00 0x04 0xD4 0xC5 (4800)
0x05 0x10 0x05 0xF6 0x00 0x01 0x02 0x00 0x05 0x15 0x05 (9600)
0x05 0x10 0x05 0xF6 0x00 0x01 0x02 0x00 0x06 0x55 0x04 (19200)
0x05 0x10 0x05 0xF6 0x00 0x01 0x02 0x00 0x07 0x94 0xC4 (38400)

========== Read/Write Registers ==========
====================================
Format:
// Read
// ID[1],func[1],1stAddr[2],regCount[2],chkSum[2]
// Write
// ID[1],func[1],1stAddr[2],regCount[2],byteCountOfValues[1],regValues[2*regCount],chkSum[2]

// Precede any Writes with the "Apply Password" command which is
// a Write to register 1520(0x5F0) as described under "Custom Commands" below.

Resettable L1 Energy - Reg addr 1550(0x60E)
 Read 4 bytes kWh * 10 (or kWh * 10000 in HiRes mode):
0x01 0x03 0x06 0x0E 0x00 0x02 0xA5 0x40 addr=1
0x05 0x03 0x06 0x0E 0x00 0x02 0xA4 0xC4 addr=5
 Write 4 bytes, only 0x00000000 is accepted:
0x01 0x10 0x06 0x0E 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0x59 0x83
0x05 0x10 0x06 0x0E 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0x4C 0xB3

Resettable L2 Energy - Reg addr 1552(0x610)
 Read 4 bytes kWh * 10 (or kWh * 10000 in HiRes mode):
0x05 0x03 0x06 0x10 0x00 0x02 0xC4 0xC2 addr=5
 Write 4 bytes, only 0x00000000 is accepted:
0x05 0x10 0x06 0x10 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0xCC 0x33

Resettable L3 Energy - Reg addr 1554(0x612)
 Read 4 bytes kWh * 10 (or kWh * 10000 in HiRes mode):
0x05 0x03 0x06 0x12 0x00 0x02 0x65 0x02 addr=5
 Write 4 bytes, only 0x00000000 is accepted:
0x05 0x10 0x06 0x12 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0x4D 0xEA

Date/Time (RTC) - Reg addr 1556(0x614)
 Read 8 bytes of date and time:
0x05 0x03 0x06 0x14 0x00 0x04 0x05 0x01 addr=5 (8 bytes of date/time)
8 bytes returned - Zero, year, month, day, weekday, hour, minute, second
year=0-99, month=1-12, day=1-31, weekday=1-7, hour=0-23, minute=0-59, second=0-59
 Write 8 bytes, same format:
0x05 0x10 0x06 0x14 0x00 0x04 0x08 0x00 0x00 0x04 0x05 0x06 0x00 0x00 0x00 0x46 0x83

0x05 0x10 0x06 0x14 0x00 0x04 0x08 0x00 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x03 0x84

Resettable Max Demand - Reg addr 1173(0x495)
Read 4 bytes RMS Watts * 10:
0x05 0x03 0x04 0x95 0x00 0x02 0xD4 0x93
Write 4 bytes, only 0x00000000 is accepted:
0x05 0x10 0x04 0x95 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0x1D 0x0C

Resettable Total kWh - Reg addr 1201(0x4B1)
Read 4 bytes kWh * 10:
0x05 0x03 0x04 0xB1 0x00 0x02 0x94 0x98
Write 4 bytes, only 0x00000000 is accepted:
0x05 0x10 0x04 0xB1 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0x1E 0xE7

Resettable Reverse Total kWh - Reg addr 1313(0x521)
Read 4 bytes kWh * 10 (or kWh * 10000 in HiRes mode):
0x05 0x03 0x05 0x21 0x00 0x02 0x95 0x49
Write 4 bytes, only 0x00000000 is accepted:
0x05 0x10 0x05 0x21 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0x1A 0x1B

CT ratio - Reg addr 1603(0x643)
Read 2 byte CT ratio:
0x01 0x03 0x06 0x43 0x00 0x01 0x75 0x56 addr=1
0x05 0x03 0x06 0x43 0x00 0x01 0x74 0xD2 addr=5
Write 2 byte CT ratio:
0x05 0x10 0x06 0x43 0x00 0x01 0x02 0x00 0x64 0xFD 0x88 (set to 100)
0x05 0x10 0x06 0x43 0x00 0x01 0x02 0x00 0xC8 0xFD 0xF5 (set to 200)
0x05 0x10 0x06 0x43 0x00 0x01 0x02 0x03 0x20 0xFD 0x4B (set to 800)
0x05 0x10 0x06 0x43 0x00 0x01 0x02 0x03 0xE8 0xFC 0xDD (set to 1000)

Max Demand Time Selector - Reg addr 1610(0x64A)
Read 2 byte demand period setting (1=15_min 2=30_min 4=60_min):
0x05 0x03 0x06 0x4A 0x00 0x01 0xA4 0xD0
Write 2 byte period setting:
0x05 0x10 0x06 0x4A 0x00 0x01 0x02 0x00 0x01 0x3D 0x3A (set to 1=15 minutes)
0x05 0x10 0x06 0x4A 0x00 0x01 0x02 0x00 0x02 0x7D 0x3B (set to 2=30 minutes)
0x05 0x10 0x06 0x4A 0x00 0x01 0x02 0x00 0x04 0xFD 0x39 (set to 4=60 minutes)
//0x05 0x10 0x06 0x4A 0x00 0x01 0x02 0x00 0x00 0xFC 0xFA (0 should fail)
//0x05 0x10 0x06 0x4A 0x00 0x01 0x02 0x00 0x03 0xBC 0xFB (3 should fail)

ModBus Device Address - Reg addr 1652(0x674)
Write 2 byte ModBus device address (only 1 to 247 per the ModBus spec)
(also fails if you try to set the address to the current address)
0x05 0x10 0x06 0x74 0x00 0x01 0x02 0x00 0x0C 0xF8 0x21 (change to 12==0x0C)
0x05 0x10 0x06 0x74 0x00 0x01 0x02 0x00 0xF7 0xB9 0xA2 (change to 247==0xF7)
0xF7 0x10 0x06 0x74 0x00 0x01 0x02 0x00 0x05 0x25 0x43 (change to 5)
0x05 0x10 0x06 0x74 0x00 0x01 0x02 0x00 0xF8 0xF9 0xA6 (248==0xF8, should fail)
0x05 0x10 0x06 0x74 0x00 0x01 0x02 0x00 0x05 0x38 0x27 (change to 5, should fail)

In1 Pulse Ratio - Reg addr 1501(0x5DD)
Read 2 byte pulse ratio:
0x05 0x03 0x05 0xDD 0x00 0x01 0x15 0x78
Write 2 byte pulse ratio (e.g. 5 to 1):
0x05 0x10 0x05 0xDD 0x00 0x01 0x02 0x00 0x01 0x12 0xDD
0x05 0x10 0x05 0xDD 0x00 0x01 0x02 0x00 0x05 0x13 0x1E
0x05 0x10 0x05 0xDD 0x00 0x01 0x02 0x02 0x03 0x92 0x7C

In2 Pulse Ratio - Reg addr 1502(0x5DE)
Read 2 byte pulse ratio:
0x05 0x03 0x05 0xDE 0x00 0x01 0xE5 0x78
Write 2 byte pulse ratio (e.g. 7 to 1):
0x05 0x10 0x05 0xDE 0x00 0x01 0x02 0x00 0x07 0x92 0xEC

In3 Pulse Ratio - Reg addr 1503(0x5DF)
Read 2 byte pulse ratio:
0x05 0x03 0x05 0xDF 0x00 0x01 0xB4 0xB8
Write 2 byte pulse ratio (e.g. 17 to 1):
0x05 0x10 0x05 0xDF 0x00 0x01 0x02 0x00 0x11 0x12 0xF3

Resettable In1 Raw Pulse Count - Reg addr 1510(0x5E6)
Read 4 byte pulse count (before ratio is applied):
0x05 0x03 0x05 0xE6 0x00 0x02 0x24 0xB4
Write 4 bytes, only 0x00000000 is accepted:
0x05 0x10 0x05 0xE6 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0x57 0xAD

Resettable In2 Raw Pulse Count - Reg addr 1512(0x5E8)
Read 4 byte pulse count (before ratio is applied):
0x05 0x03 0x05 0xE8 0x00 0x02 0x45 0x77
Write 4 bytes, only 0x00000000 is accepted:
0x05 0x10 0x05 0xE8 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0xD6 0x21

Resettable In3 Raw Pulse Count - Reg addr 1514(0x5EA)
Read 4 byte pulse count (before ratio is applied):
0x05 0x03 0x05 0xEA 0x00 0x02 0xE4 0xB7
Write 4 bytes, only 0x00000000 is accepted:
0x05 0x10 0x05 0xEA 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0x57 0xF8

I/O Pin 1 State - Reg addr 1516(0x5EC)
Read 2 bytes giving just the level:
0x05 0x03 0x05 0xEC 0x00 0x01 0x44 0xB7

Write 4 bytes, set the level and time (state=0/1, seconds=1-9999, 0 forever):
0x05 0x10 0x05 0xEC 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0xD7 0xD2 (0==closed)
0x05 0x10 0x05 0xEC 0x00 0x02 0x04 0x00 0x00 0x00 0x07 0x96 0x10 (0==closed 7 sec)
0x05 0x10 0x05 0xEC 0x00 0x02 0x04 0x00 0x01 0x00 0x00 0x86 0x12 (1==open)

I/O Pin 2 State - Reg addr 1518(0x5EE)
Read 2 bytes giving just the level:
0x05 0x03 0x05 0xEE 0x00 0x01 0xE5 0x77

Write 4 bytes, set the level and time (state=0/1, seconds=1-9999, 0 forever):
0x05 0x10 0x05 0xEE 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0x56 0x0B (0==closed)
0x05 0x10 0x05 0xEE 0x00 0x02 0x04 0x00 0x00 0x00 0x03 0x16 0x0A (0==closed 3 sec)
0x05 0x10 0x05 0xEE 0x00 0x02 0x04 0x00 0x01 0x00 0x00 0x07 0xCB (1==open)

ModBus Reply Delay - Reg addr 1527(0x5F7)
Read 2 bytes, milliseconds of extra delay:
0x05 0x03 0x05 0xF7 0x00 0x01 0x34 0xB0
Write 2 bytes to set additional milliseconds of delay before replies begin:
0x05 0x10 0x05 0xF7 0x00 0x01 0x02 0x00 0x00 0xD4 0xD7 (0)
0x05 0x10 0x05 0xF7 0x00 0x01 0x02 0x03 0xE8 0xD4 0x69 (1000 == 1 second)

ModBus Password Disable - Reg addr 1528(0x5F8)
Read 2 byte flag (0=password check, 1==no password check):
0x05 0x03 0x05 0xF8 0x00 0x01 0x04 0xB3
Write 2 bytes to enable/disable (0=password check, 1==no password check):
0x05 0x10 0x05 0xF8 0x00 0x01 0x02 0x00 0x00 0xD4 0x28 (0)
0x05 0x10 0x05 0xF8 0x00 0x01 0x02 0x00 0x01 0x15 0xE8 (1)

UserData (2 words) - Reg addr 1540(0x604)
Read 4 bytes of priviously written data:
0x05 0x03 0x06 0x04 0x00 0x02 0x84 0xC6
Write 4 bytes of your choice:
0x05 0x10 0x06 0x04 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0xCC 0xCC (zeros)
0x05 0x10 0x06 0x04 0x00 0x02 0x04 0x12 0x34 0x56 0x78 0xB7 0xF8
0x05 0x10 0x06 0x04 0x00 0x02 0x04 0x44 0x33 0x22 0x11 0xF0 0x9F
0x05 0x10 0x06 0x04 0x00 0x02 0x04 0x00 0xBC 0x61 0x4E 0xA4 0x8C (decimal 12345678)

LcdItemTable - Reg addr 1546(0x60A)
Check which items are in the LCD display rotation (the contents of LcdItemTable[]).
Read the contents of LcdItemTable[]; give the space you have available):
0x05 0x03 0x06 0x0A 0x00 0x7D 0xA4 0xE5 (125 word max; send this since table size is unknown)
Select which items are in the LCD display rotation (sets the contents of LcdItemTable[])
Write bytes with item numbers, must pad to word boundary:
0x05 0x10 0x06 0x0A 0x00 0x02 0x04 0x01 0x04 0x07 0x00 0x0F 0x4D (items 1, 4, 7 + pad to word boundary)
0x05 0x10 0x06 0x0A 0x00 0x02 0x04 0x01 0x04 0x07 0x09 0xCF 0x4B (items 1, 4, 7, 9)
0x05 0x10 0x06 0x0A 0x00 0x01 0x02 0x01 0x2F 0xB2 0x76 (items 1, 47(0x2F) to make sure max can be set)
0x05 0x10 0x06 0x0A 0x00 0x02 0x04 0x01 0x30 0x31 0x32 0xD8 0xF6 (items 1, 48 49 50 to make sure max can be set)
0x05 0x10 0x06 0x0A 0x00 0x03 0x06 0x01 0x27 0x30 0x31 0x32 0x00 0xA8 0xD5 (items 1, 39, 48 49 50)
(Set to items 1-47 + pad to word bounadry)
0x05 0x10 0x06 0x0A 0x00 0x18 0x30 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 00 0x5B 0xE2

========== Custom Commands ==========

====================================

ModBus_userMap - Reg addr 1544(0x608)

The UserMap feature is a configurable list of registers you can Read with a single request.
The list consists of multiple pairs of a RegAddr and a WordCount,
Each pair indicates a starting register and how many words to read.
The starting RegAddrs can be in any order.
The count can be greater than the length of the starting register to read a group.
This table shows the valid starting RegAddrs and maximum wordCount for a group:

```
START  MinCnt  MaxCnt   //RegAddr\Name
0x001    2      2       // 0001 DummyReg
0x3F1    2      8       // 1009 Total Watts
0x3F3    2      6       // 1011 Watts L1
0x3F5    2      4       // 1013 Watts L2
0x3F7    2      2       // 1015 Watts L3
0x409    1      1       // 1033 Frequency
0x47B    2      8       // 1147 VAR L1/2/3
0x47D    2      6       // 1149 VAR L1
0x47F    2      4       // 1151 VAR L2
0x481    2      2       // 1153 VAR L3
0x48B    2      6       // 1163 Amps L1
0x48D    2      4       // 1165 Amps L2
0x48F    2      2       // 1167 Amps L3
0x4B5    2      2       // 1205 Total kWh
0x4BE    1      3       // 1214 Volts L1
0x4BF    1      2       // 1215 Volts L2
0x4C0    1      1       // 1216 Volts L3
0x515    2      6       // 1301 L1 kWh
0x517    2      4       // 1303 L2 kWh
0x519    2      2       // 1305 L3 kWh
0x523    2      10      // 1315 Reverse kWh
0x525    2      8       // 1317 L1 Reverse kWh
0x527    2      6       // 1319 L2 Reverse kWh
0x529    2      4       // 1321 L3 Reverse kWh
0x52B    2      2       // 1323 Total Reactive kWh
0x53C    1      3       // 1340 Power Factor (Cosine) L1
0x53D    1      2       // 1341 Power Factor (Cosine) L2
0x53E    1      1       // 1342 Power Factor (Cosine) L3
0x5DC    1      1       // 1500 In1/In2/In3 Pins State Bit-map
0x5E0    2      6       // 1504 Pulse Count 1
0x5E2    2      4       // 1506 Pulse Count 2
0x5E4    2      2       // 1508 Pulse Count 3
0x5FA    3      3       // 1530 Serial Number
0x60C    1      2       // 1548 OneWire Port_1 Temperature probe
0x60D    1      1       // 1549 OneWire Port_2 Temperature probe
0x6AC    1      1       // 1708 Firmware Version
```

//

Write with function code 16 to setup the UserMap:

```
//Devaddr[1], 16[1], 1544[2], wordCnt[2], dataByteCount[1], dataWords[2*wordCnt], chkSum[2]
//  DataWords starting at offset 7 look like this:
//    pairCnt[2], pairWords[pairCnt]  (where pairCnt == dataByteCount-2)
//  PairWords starting at offset 9 look like this:
//      addr1[2], cnt1[2], addr2[2], cnt2[2], addr3[2], cnt3[2], ...
```

```
//  Map is cleared if pairCnt is zero.
//  Success returns 0.
//  Failure returns a ModBus error code.
//
//adr func  mapAddress totalWords bytes pairCount  regAddr1  wordCount1  regAddr2
wordCount2
  map == 1 pair (the Serial Number)          - 1530(0x5FA)  3 regs
0x05 0x10 0x06 0x08  0x00 0x03  0x06  0x00 0x01  0x05 0xFA 0x00 0x03  0x2A 0x5A
  map == 1 pair                              - 1301(0x515)  2 regs
0x05 0x10 0x06 0x08  0x00 0x03  0x06  0x00 0x01  0x05 0x15 0x00 0x02  0xDA 0x6F
  map == 2 pair                              - 1301(0x515)  4 regs, 1303(0x517)  2 regs
0x05 0x10 0x06 0x08  0x00 0x05  0x0A  0x00 0x02  0x05 0x15 0x00 0x04  0x05 0x17 0x00
0x02 0x91 0xA4
  map == 3 pair                              - 1301(0x515)  4 regs, 1303(0x517)  2 regs, 1214(0x4BE)  3
regs
0x05 0x10 0x06 0x08  0x00 0x07  0x0E  0x00 0x03  0x05 0x15 0x00 0x04  0x05 0x17 0x00
0x02  0x04 0xBE 0x00 0x03 0xCB 0xE9
  map == 6 pair                              - 1301(0x515)  4 regs, 1303(0x517)  2 regs, 1214(0x4BE)  3
regs, 1009(0x3F1)  4 regs, 1147(0x47B)  4 regs, 1340(0x53C)  3 regs
0x05 0x10 0x06 0x08  0x00 0x0D  0x1A  0x00 0x06  0x05 0x15 0x00 0x04  0x05 0x17 0x00
0x02  0x04 0xBE 0x00 0x03  0x03 0xF1 0x00 0x04  0x04 0x7B 0x00 0x04  0x05 0x3C 0x00
0x03 0x06 0xF3
  map == 7 pair
0x05 0x10 0x06 0x08  0x00 0x0F  0x1E  0x00 0x07  0x05 0x15 0x00 0x04  0x05 0x17 0x00
0x02  0x04 0xBE 0x00 0x03  0x03 0xF1 0x00 0x04  0x04 0x7B 0x00 0x04  0x05 0x3C 0x00
0x03  0x05 0x15 0x00 0x02 0xE0 0xA9
  clear the map
0x05 0x10 0x06 0x08  0x00 0x01  0x02  0x00 0x00  0xF3 0xD8

 ModBus_userMap - Reg addr 1544(0x608)
Read with function code 03 to Read back the map; give the word space available:
0x05 0x03 0x06 0x08  0x00 0x02  0x44 0xC5  (2 word max, allows 1 pair)
0x05 0x03 0x06 0x08  0x00 0x03  0x85 0x05  (3 word max, allows 1 pair)
0x05 0x03 0x06 0x08  0x00 0x04  0xC4 0xC7  (4 word max, allows 2 pair)
0x05 0x03 0x06 0x08  0x00 0x0C  0xC5 0x01  (12 word max, allows 6 pair)
0x05 0x03 0x06 0x08  0x00 0x0E  0x44 0xC0  (14 word max, allows 7 pair)

 ModBus_userMap - Reg addr 1544(0x608)
Read with function code 04 to Apply the map and get data pointed to by the map;
 give the word space available:
0x05 0x04 0x06 0x08  0x00 0x00  0x70 0xC4  (0 word max, gives error unless map is empty)
0x05 0x04 0x06 0x08  0x00 0x01  0xB1 0x04  (1 word max)
0x05 0x04 0x06 0x08  0x00 0x02  0xF1 0x05  (2 word max)
0x05 0x04 0x06 0x08  0x00 0x03  0x30 0xC5  (3 word max)
0x05 0x04 0x06 0x08  0x00 0x04  0x71 0x07  (4 word max)
0x05 0x04 0x06 0x08  0x00 0x0A  0xF0 0xC3  (10 word max)
0x05 0x04 0x06 0x08  0x00 0x13  0x31 0x09  (19 word max)
0x05 0x04 0x06 0x08  0x00 0x14  0x70 0xCB  (20 word max)
0x05 0x04 0x06 0x08  0x00 0x7D  0xB0 0xE5  (125 word max; probably always just send this)
```

ReadABC - Reg addr 1800(0x708)
Request = DevAddr[1], FunctionCode[1], 1stAddr[2], wordCount[2], CRC[2]
Read 125 words (250 bytes) of meter data in EKM format,
0x01 0x04 0x07 0x08 0x00 0x7D 0xB0 0x9D addr=1
0x05 0x04 0x07 0x08 0x00 0x7D 0xB1 0x19 addr=5
0xF7 0x04 0x07 0x08 0x00 0x7D 0xA4 0x0B addr=247
Response = DevAddr[1], FunctionCode[1], byteCnt[1], data[byteCnt], CRC[2]
Returns 255 bytes of response total:
  data[0] = 0x05 (Dev ID)
  data[1] = 0x04 (function code)
  data[2] = 0xFA (data byte count)
  data[3] = METER_TYPE:major
  data[4] = METER_TYPE:minor
  data[5] = FIRMWARE_VERSION
  data[6] = SerialNumber (5 bytes)
  data[11] = CT_ratio   (2 bytes)
  data[13] = Total kWh
  data[17] = Total Reactive kWh
  data[21] = Total Reverse kWh
  data[25] = L1 Total kWh
  data[29] = L2 Total kWh
  data[33] = L3 Total kWh
  data[37] = L1 Total Reverse kWh
  data[41] = L2 Total Reverse kWh
  data[45] = L3 Total Reverse kWh
  data[49] = Resettable Total kWh (HiRes dependant)
  data[53] = Resettable Reverse kWh (HiRes dependant)
  data[57] = Rate 1 Total kWh)
  data[61] = Rate 2 Total kWh)
  data[65] = Rate 3 Total kWh)
  data[69] = Rate 4 Total kWh)
  data[73] = Rate 1 Total Reverse kWh
  data[77] = Rate 2 Total Reverse kWh
  data[81] = Rate 3 Total Reverse kWh
  data[85] = Rate 4 Total Reverse kWh
  data[89] = Max demand
  data[93] = Max demand period selection
  data[94] = L1 Voltage
  data[96] = L2 Voltage
  data[98] = L3 Voltage
  data[100] = L1 current
  data[104] = L2 current
  data[108] = L3 current
  data[112] = L1 power
  data[116] = L2 power
  data[120] = L3 power
  data[124] = Total power
  data[128] = L1 Reactive Power
  data[132] = L2 Reactive Power
  data[136] = L3 Reactive Power
  data[140] = Total Reactive Power
  data[144] = Frequency
  data[146] = L1 PF indicator ('L'=inductive 'C'=capacitive ' '=1.00)
  data[147] = L1 Power Factor Cosine * 100

```
    data[149] = L2 PF indicator ('L'=inductive 'C'=capacitive ' '=1.00)
    data[150] = L2 Power Factor Cosine * 100
    data[152] = L3 PF indicator ('L'=inductive 'C'=capacitive ' '=1.00)
    data[153] = L3 Power Factor Cosine * 100
    data[155] = In1 converted pulse count
    data[159] = In2 converted pulse count
    data[163] = In3 converted pulse count
    data[167] = In1 conversion ratio
    data[169] = In2 conversion ratio
    data[171] = In3 conversion ratio
    data[173] = In1 raw pulse count
    data[177] = In2 raw pulse count
    data[181] = In3 raw pulse count
    data[185] = In1/In2/In3 state bitmap (bit3=In1 bit2=In2 bit0=In3)
    data[186] = Max Demand auto-reset period selection
    data[187] = Variable Imp/kWh pulse rate selection
    data[189] = Current direction bitmap (bits 3,2,1=L3,L2,L1 set=rev)

    data[190] = Relay settings indicator
    data[191] = Decimal place indicator
    data[192] = Year    (0-99)
    data[193] = Month   (1-12)
    data[194] = Day     (1-31)
    data[195] = Weekday (1-7)
    data[196] = Hour    (0-23)
    data[197] = Minute  (0-59)
    data[198] = Second  (0-59)

    // Unused bytes are zero filled.

    data[253] = CRC_lo
    data[254] = CRC_hi
//End of ReadABC
```