# PREFACE

**kWhCube Modbus Serial Communications**
**Installation and Protocol Guide**
**Revision 1.01**
**April 2002**

This manual represents the Modbus serial communications firmware as manufactured at the time of publication. It assumes standard software. Special versions of software may be fitted, in which case you will be provided with additional details.
Every effort has been made to ensure that the information in this manual is complete and accurate. We revised this manual but cannot be held responsible for errors or omissions.
The hardware has been designed and tested in accordance with EN 61010-1, 'Safety Requirements for Electrical Equipment for Measurement, Control and Laboratory Use'. This operating guide contains information and warnings, which must be followed by the user to ensure safe operation and to maintain the apparatus in a safe condition.
We reserve the right to make changes and improvements to the product without obligation to incorporate these changes and improvements into units previously shipped.

General Editor : Ian Sykes BSc (hons).

## Table Of Contents

# 1. Safety

## 1.1 Warning Symbols

This manual provides details of safe installation and operation of the Modbus serial communications option. Safety may be impaired if the instructions are not followed. Labels on individual devices give details of equipment ratings for safe operation. Take time to examine all labels on the meter and to read this manual before commencing installation.

**CAUTION**
**Refer to User Instructions**

**DANGER**
**Risk of Electric Shock**

**Figure 1-1 Safety Symbols**

> **WARNING**
> The equipment contains no user serviceable parts. Installation and commissioning should be carried out by qualified personnel

## 1.2 Maintenance

The equipment should be maintained in good working order. Damage to the product should be repaired by the manufacturer. The meter may be cleaned by wiping lightly with a soft cloth. No solvents or cleaning agents should be used. All inputs and supplies, to the meter and options module must be isolated before cleaning any part of the equipment.

# 2. Description

The serial communications option uses a high-speed microprocessor to extract information from a kWhCube meter and provides an interface to an industry standard Modbus system.

Use of a dedicated communications processor ensures optimum efficiency, allowing fast access to data on systems with multiple meters. The use of Modbus protocol ensures compatibility with existing systems and/or a number of readily available software packages.

The RS485 interface communicates over distances up to 1200 metres with data rates of 4800, 9600 or 19200 to suit system requirements.

The kWhCube may be supplied with the Modbus serial communications firmware added within the standard instrument enclosure. This factory fitted option provides a compact, cost effective solution for RS485 communications.

Connection to the Modbus system is made at the rear of the kWhCube (terminals 25-28) as shown below.

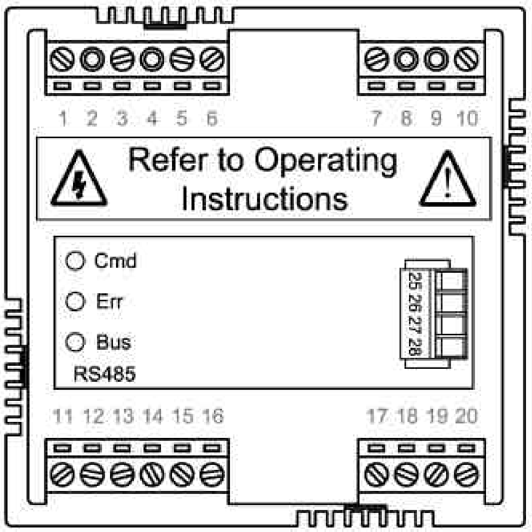


**Figure 2-1 Internal Modbus Option**

# 3. Programming

***NOTE:*** Refer to the kWhCube Operating Instruction manual for full details on programming mode.

The kWhCube programming menu will automatically expand to include Baud Rate and Address settings when valid Modbus Options hardware is detected.

### 3.1.1 Setting Baud Rate

Remote serial communications speeds of 4800, 9600 or 19200 may be selected to suit external system requirements. Higher speeds will provide faster data access while a slower speed may be required in electrically noisy environments.
Enter programming mode on the kWhCube as described in the Meter Operating Instructions. Scroll through program settings using '***PROG***' until the meter displays:


**Figure 3-1 Setting The Baud Rate**

Press △ to select the next highest available baud rate setting.
Press ▽ to select the next lowest available baud rate setting.
Press '***PROG***' and hold for 2 seconds when done.

### 3.1.2 Setting Meter Address

Each outstation (Meter) on a multi-drop Modbus system is identified to the master by a unique address. The kWhCube may be addressed anywhere in the full Modbus range of 1-247.

Enter programming mode on the kWhCube as described in the Meter Operating Instructions. Scroll through program settings using '***PROG***' until the meter displays:


**Figure 3-2 Setting The Modbus Address**

Press △ to increment the Modbus Address.
Press ▽ to decrement the Modbus Address.
Press '***PROG***' and hold for 2 seconds when done.

# 4. Modbus Communication

## 4.1 Description

The kWhCube Modbus Option provides a serial communications interface to external systems. This allows remote reading and programming of the meter by a host computer (e.g. PC). The communication protocol used is a subset of Modicon's Modbus enabling use of standard off the shelf software packages and connection to standard controllers.

### 4.1.1 Communication Address

Each meter on a Modbus serial communication network must be assigned a unique address between 1 and 247. This is carried out in programming mode as described in Section 3. If two or more meters, connected in a multi-drop network have the same address, data on the network will be corrupted and communication will fail.

### 4.1.2 Data Format

The device uses a fixed data format for serial communications:

| | | |
|---|---|---|
| 1 Start Bit | 8 Data Bits | 1 Stop Bit |

The 8 data bits are always transmitted least significant bit first. This data byte is binary coded.
The baud rate is programmable as **4800, 9600, or 19200 baud.** This is carried out in programming mode as described in Section 3.

## 4.2 RS485

### 4.2.1 Description

RS485 communication on the kWhCube enables connection of up to 128 meters on a single pair of wires (247 with repeaters). This pair is used for transmission and reception with each meter (and the host) automatically switching data direction. The host should be fitted with an RS485 driver (or converter) capable of operation in two-wire mode (half duplex).

PC operation in RS485 two-wire mode usually requires software control of the data direction. This controls the line drivers connected to the bus at the host serial port. This direction control requires high-speed operation and may be problematic under certain multi-tasking operating systems such as Windows. It is advisable to check with the software vendor before selecting software direction control as the mode of operation. If software data direction control is not suitable, RS232-RS485 converters are available for standard PCs, which carry out automatic hardware direction control. For more information on these contact your distributor.

Each Modbus serial transaction is preceded by a device address allowing the host to temporarily communicate with a specific meter on the bus. Certain commands allow the host to communicate with all meters simultaneously. These commands are known as ***broadcasts*** and use address 0.

### 4.2.2 Cable Selection

A dedicated, screened twisted pair cable is required to provide a basic RS485 connection. A second twisted pair may be used for 0V connection if required. The cable should be chosen to suit the data rate and maximum length to be installed. The EIA RS-485-A standard provides curves that relate cable length to data rate for 24 AWG screened, twisted pair, telephone cable with a shunt capacitance of 50pf/m. For baud rates up to 19,200 the standard suggests a maximum length of 1200m for this type of cable. If other types of cable are to be used it is recommended that the cable supplier is consulted as to the suitability for use with RS485 to 19,200 baud.

### 4.2.3 Signal 0V and Cable Shield

A signal 0V termination is provided on each meter. Although RS485 does not strictly require a signal 0V, it is recommended this be connected as shown in the diagram below. This creates a known reference for the isolated RS485 system thereby reducing potential common-mode errors in the meter's driver circuit.

A cable shield is used to attenuate noise picked up from external sources. This should be continuous, and cover as much of the signal pairs as possible. It is recommended that the shield should be connected to ground at the host only.

### 4.2.4 Terminating Resistors

In order to minimise signal errors due to noise over long cable lengths, terminating resistors may be fitted. These match the RS485 device impedance to that of the cable. Two 120-ohm resistors, one at the host port terminals and the other at the most remote meter terminals are recommended for this purpose.

### 4.2.5 Connection To Meters

The bus should be taken to meters at each location for termination, using the meter terminals as a loop in-out connection. The use of spurs should be avoided wherever possible.

#### 4.2.5.1 Internal Modbus Option (RS485)

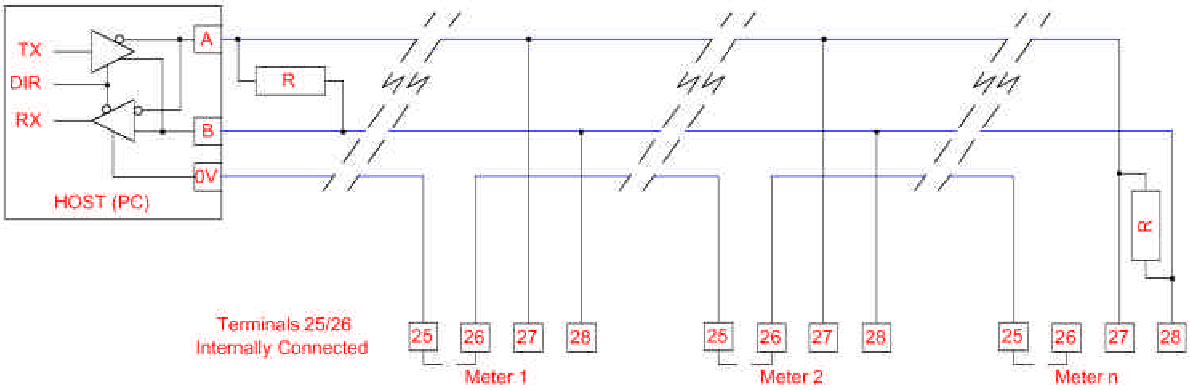


**Figure 4-1 Basic RS485 To Internal Modbus Option**

#### 4.2.5.2 Biasing the RS485 System

When an RS485 system is not communicating, all outstations are in high impedance listen mode. In this state no active drivers are present and the bus floats to an unknown state. The logic levels at the output stage of each meter's RS485 circuit will remain at the level of the last bit received. Recommended practice for RS485 networks suggests biasing of the two wires to ensure a known idle state for the networks receivers. Although biasing is not essential it can often provide a solution to a problematic system.

Biasing normally consists of a pull up (usually to an isolated 5V dc supply) and a pull down resistor. The kWhCube has no internal biasing and so connections should be made externally at a single convenient point in the network. A 5V dc external supply with two 470Ω resistors is adequate.

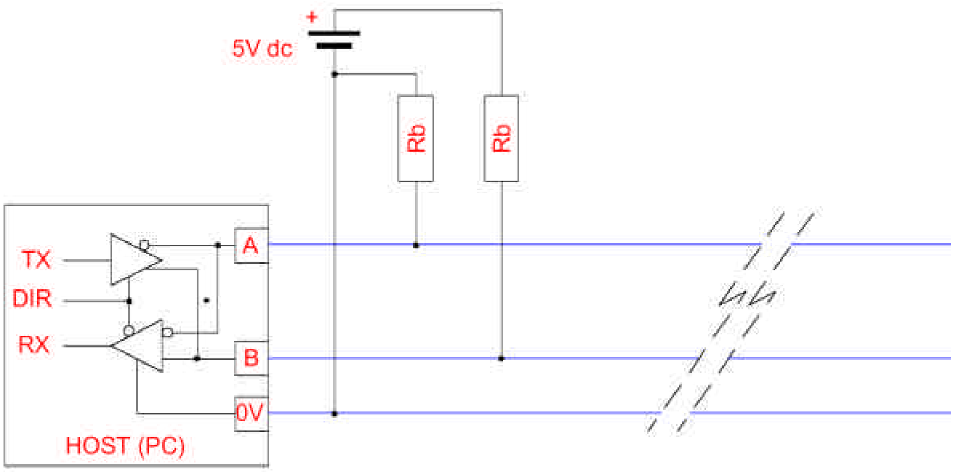


**Figure 4-2 RS485 Biasing**

## 4.3 Modbus Protocol

### 4.3.1 An Introduction To Modbus

A communication protocol defines a set of commands and data formats which will be recognised by all compatible equipment connected on a system. The protocol effectively forms a communication language.

The kWhCube Options Units utilise a subset of Modicon's 'Modbus' standard protocol. This protocol was originally developed for use by programmable logic controllers (PLCs). It defines a set of commands for reading and/or writing data to devices connected on the bus.

Modbus is a master-slave protocol with all transactions initiated by a single host (e.g. a PC). A single transaction commences with the host transmission of a command packet followed by a slave (Options Unit) reply after a short delay for processing the command.

Command packets consist of an address, a command identifier, data and a checksum for error detection. Each slave device continually monitors the bus looking for activity. Command packets are detected by all slaves but may be acted upon only by the device whose address matches that transmitted.

The host may transmit a ***broadcast command***, which uses address 0 to contact all devices on the network. In this instance all slaves act on the command but none of them may reply. This type of command may be useful, for example, in synchronising energy register reset on all meters.

The full Modbus protocol consists of many commands and modes of operation to suit a variety of controllers and applications. The kWhCube Options unit utilises only a few commands and a single transmission mode to perform many functions relevant to metering.

### 4.3.2 RTU. Transmission Mode

The RTU (Remote Terminal Unit) mode is utilised by the kWhCube Options Unit because it provides the most efficient throughput of data at any particular baud rate.

In RTU mode, the start and end of each message is marked by a silent period of at least 3.5 character periods (Approx. 3.5ms @ 9600 baud). This is shown in the RTU message frame in Figure 4-3 below.

| START | ADDRESS | FUNCTION | DATA | CRC | END |
|---|---|---|---|---|---|
| SILENT PERIOD | 8 BITS | 8 BITS | n x 8 BITS | 16 BITS | SILENT PERIOD |

**Figure 4-3 RTU Framing**

The host (PC) initiates all transactions. Slave devices continuously monitor the network, looking for messages framed by silent periods. The first character detected, after a silent period, is assumed to be an address byte and is compared to the meters internal address (zero for broadcasts). An addressed slave reads the remainder of the message and acts upon it as required.

A slave tests the message to determine it's validity and uses the transmitted checksum (CRC) to detect communication errors. A slave will only act on valid messages, received without error, specifically addressed to it.

**ADDRESS**

Valid Modbus addresses are in the range 0-247. Individual devices may be assigned addresses in the range 1-247. Address 0 is retained for broadcast commands which are handled by all slaves. When a slave responds to a command it places its own address in the reply message.

**FUNCTION**

The function code is a single byte telling the device what type of operation to perform. Valid Modbus codes are in the range 1-255 decimal but the kWhCube Options Unit handles only a small subset of these as summarised below.

| Function code | Operation | Broadcast |
|---|---|---|
| 03 | Read Multiple Registers | No |
| 04 | Read Multiple Registers | No |
| 06 | Preset A Single Register | Yes |
| 08 | Loop Back Diagnostic | No |
| 16 | Preset Multiple Registers | Yes |

**Figure 4-4 Function Code Summary**

**DATA FIELD**

Data from the host contains additional information for the remote device specific to the command. For example the data field may specify which meter readings are required or new values for energy registers.

Data from a slave may contain meter readings or other information requested by the host.

The size of the data field varies depending on command type and usage. The data format may also vary from one command to another to suit the application. Instantaneous readings for example are transmitted as 2-byte Integers, whereas energy readings are formatted as 4-byte Long Integers. Data is always transmitted with the most significant byte first. Data formatting is described in more detail in the following sections.

### 4.3.3 CRC ERROR CHECKING

A 16 bit CRC (***C**yclic* ***R**edundancy* ***C**heck*) field is tagged on to the end of all messages. This field is the result of a CRC calculation performed on the message contents. The CRC field is used by the host and receiving devices alike to determine the validity of the entire message string. A receiving device recalculates the CRC and compares it to the value contained in the message. A slave device ignores a message if the two values do not match.

> ***Note***
>
> Use of the CRC is essential when communicating in noisy environments to reduce the effects of erroneous bit errors. The meter will not reply to commands with a CRC in error and the host should re-transmit the command after a pre-determined time-out period. If the host receives a string with a CRC in error the transaction should be re-initiated.

The CRC is calculated on all bytes of a message from the address to the last data byte inclusively. Each bit of the message is processed through the CRC calculation starting with the first bit of the address. The Modbus standard method of CRC calculation requires reversal of the data bytes as they are fed serially through the bit processing routines. A simpler method involves swapping the low and high order bytes of the CRC integer at the end of the calculation. This is shown in the following routine.

The calculation is performed as follows:

1. Load a 16 Bit register (“CRC Register”) with FFFF Hex. (All 1’s).
2. Exclusive-OR the first 8 Bits of the message with the low-order byte of the CRC register. Put the result in the CRC register.
3. Shift the CRC register one bit to the right (divide by 2), filling the MSB with a zero.
4. If the bit shifted out in 3 is a 1, Exclusive-OR the CRC register with the value A001 Hex.
5. Repeat steps 3 and 4 until 8 shifts have been performed and the bits tested. A single byte has thus been processed.
6. Repeat steps 2 to 5 using the next 8 bit byte of the message until all bytes have been processed.
7. The final contents of the CRC register is tagged on to the end of the message with the most significant byte first.
8. Swap the low and high order bytes of the integer result

An implementation of the CRC calculation in C code is shown below :

```
unsigned int check_sum(unsigned char *buff, char start, char bytes)
{
    char byte_cnt,bit_cnt;          /*  loop counters  */
    unsigned int crc_reg;           /*  Result register  */
   unsigned int CRCHi, CRCLO; /*Low and high order bytes of the crc*/

    crc_reg = 0xFFFF;                       /* Set the CRC register to all 1's  */

    /*  Repeat for each byte of sub string  */
    for(byte_cnt=start; byte_cnt<(bytes+start); byte_cnt++)
      {
      crc_reg = crc_reg ^ (unsigned int)buff[byte_cnt]; /*EXOR CRC & Next Byte*/

       /*  Test each bit of the CRC  */
       for(bit_cnt=0; bit_cnt<8; bit_cnt++)
         {
          if(crc_reg & 0x0001)
               {
                crc_reg = crc_reg >> 1;   /* IF LSB=1 EXOR CRC with A001H */
                crc_reg = crc_reg ^ 0xA001;         /*  Then shift CRC toward LSB  */
               }
          else crc_reg = crc_reg >> 1;     /*  ELSE Shift CRC towards LSB  */
         }
      }
CRCLo=crc_reg>>8;  /*Swap the low and high order bytes of the crc result*/
CRCHi=crc_reg<<8;
crc_reg = CRCLo+CRCHi;
 return crc_reg;                                  /*  Final CRC register Result  */
}
```

## 4.4 Options Unit Data Tables

Data in the kWhCube Options Unit is arranged in several tables for convenience. Individual tables contain like information. Table data may be read only (e.g. instantaneous readings) or read/write access (e.g. CT primary). Data in each table is pointed to in a Modbus command by two consecutive data address bytes. The first byte defines the table number and the second byte the offset of the data in the table. For example, 'address 2 , 1' would access Table 2, Entry 1 (3-Phase kWh ). The Modbus standard defines data addresses using a 16-bit integer. In the case of the kWhCube Options Unit the high byte of this integer is represents the table number and the low byte the offset. A Modbus integer address may be calculated as:

**Modbus Data Address = (256 x Table No) + Table Offset**

**SIGNED INTEGER**

Signed Integers are 16 bit values transmitted as two 8 bit bytes. The most significant byte is always transmitted first. These values vary in the range -32767 to +32767 although some registers have a limited range of acceptable values. The most significant bit defines the sign, zero indicating positives.

**UNSIGNED INTEGER**

Unsigned Integers are 16 bit values transmitted as two 8 bit bytes. The most significant byte is always transmitted first. These values vary in the range 0 to 65535 although some registers have a limited range of acceptable values.

**UNSIGNED LONG INTEGERS (Unsigned Long)**

Unsigned long integers are 32 bit values transmitted as four 8-bit bytes. The most significant byte is always transmitted first. These values vary in the range 0 to 4294967295 although energy registers in the KWhCube have a limited range, 0-99999999.

### 4.4.1 Table 2 Energy Registers

| Offset | Address | Contents | Format | Bytes | Words | Access |
|---|---|---|---|---|---|---|
| 0 | 512 | Energy DP Hi | Integer | 2 | 1 | Read Only |
| 1 | 513 | Energy DP Lo | Integer | 2 | 1 | Read Only |
| 2 | 514 | kWh Hi | Integer | 2 | 1 | Read/Write |
| 3 | 515 | kWh Lo | Integer | 2 | 1 | Read/Write |
| 4 | 516 | KWh Count Hi | Integer | 2 | 1 | Read/Write |
| 5 | 517 | KWh Count Lo | Integer | 2 | 1 | Read/Write |

#### 4.4.1.1 Energy Registers

The non-resettable and resettable (count) kWh registers of the kWhCube are stored in Modbus Table 2 as unsigned long integers.

#### 4.4.1.2 Writing to Energy Registers

Function 6 or 16 may be used to write to the energy registers in Table 2.
Function 6 allows access to the upper and lower integers of the 4-byte long individually.
Function 16 may be used to access a number of long integers using a single command. This is most useful for setting both registers to 0 simultaneously. Valid commands must send an even number of integers (2 integers per long) starting at an even address in Table 2 (Start of a register). Failure to follow these basic rules will result in an exception response.

#### 4.4.1.3 Energy Scaling

Energy readings from the kWhCube are stored as unsigned long integer values with no decimal point or legend (e.g. kWh or MWh). An Energy DP variable is provided to enable conversion of the raw data to real numbers in basic Wh. The Energy DP is a constant value calculated in the kWhCube as a function of CT Primary and Scaling Factor programming. To convert raw data to real numbers:

$$E = L \times 10^{(K-3)}$$

**Where:** **L** **=** Long Integer number from Table 2
**K** = Energy DP Constant from Table 2
**E** **=** Scaled Energy Result

**Example:**
If the meter displays 99999.9 kWh the Energy DP Constant would be transmitted as K=5 and the kWh register as 999999.
The host would calculate the scaled energy reading as:

$999999 \times 10^{(5-3)} = 999999 \times 100 = 99999900$Wh

The host programmer could take two approaches to interpreting the data from the kWhCube:

- ✓ Enter a fixed scaling factor (x100 for Wh or x0.1 for kWh in above example). This would be set for each meter in the system based on its display after commissioning.
- ✓ Use the transmitted Energy DP Constant, as shown above, to position the DP in the interpreted result.

### 4.4.2 Table 11 Instantaneous Meter Readings

| Offset | Address | Contents | Format | Bytes | Words | Access |
|---|---|---|---|---|---|---|
| 0 | 2816 | ±kW | Integer | 2 | 1 | Read Only |
| 1 | 2817 | Power Scale | Integer | 2 | 1 | Read Only |

#### 4.4.2.1 Scaling Instantaneous Values

Instantaneous kW readings from the kWhCube are provided as signed integer values with no decimal point or legend (e.g. kW or MW). A Scaling factor is provided to enable conversion of the raw data to real numbers in Watts. This scaling factor is a constant value calculated in the kWhCube as a function of CT Primary and Scaling Factor programming.

To convert raw data to real numbers:

$$R = I \times 10^{(K-3)}$$

**Where:** **I** **=** Integer number from Table 11
**K** = Power Scale from Table 11
**R** **=** Real 3 Phase Power in Watts

**Example:**

If the meter display shows 36.00kW the Power Scale value would be transmitted as K=4 and the kW Value (I) as 3600.

3-Ph Power would be calculated as:

$R = 3600 \times 10^{(4-3)} = 3600 \times 10 = 36000W$

### 4.4.3 Table 14 Meter Set-up

| Offset | Address | Contents | Format | Bytes | Words | Access |
|---|---|---|---|---|---|---|
| 0 | 3584 | CT Primary /10 | unsigned Int | 2 | 1 | Read/Write |
| 1 | 3585 | Scal | unsigned Int | 2 | 1 | Read/Write |
| 2 | 3586 | Pulse Rate | unsigned Int | 2 | 1 | Read/Write |
| 3 | 3587 | Pulse On Time | unsigned Int | 2 | 1 | Read/Write |
| 4 | 3588 | Baud Rate | unsigned Int | 2 | 1 | Read/Write |
| 5 | 3589 | Modbus ID | unsigned Int | 2 | 1 | Read/Write |
| 6 | 3590 | Meter Model | unsigned Int | 2 | 1 | Read Only |
| 7 | 3591 | Meter Type | unsigned Int | 2 | 1 | Read Only |
| 8 | 3592 | Meter Software | unsigned Int | 2 | 1 | Read Only |

#### 4.4.3.1 Meter Set-up Values

Information about the MultiCube's configuration is available in Table 14 as unsigned integers.

- **CT Primary/10**. CT Primary. Scaled as real CT/10 (eg 20 => 200A)
- **Scal**. Scaling Factor setting (0=0.1, 1=1.0, 2=10.0, 3=100.0, 4=1000.0)
- **Pulse Rate**. Pulse Rate ($10^n$ Counts Of Register Per Pulse Output. n=0-3)
- **Pulse On Time.** Output Pulse On Time /10 (1-255 => 100ms - 25.5 sec).
- **Baud Rate.** (48, 96 or 192) RS485/422 baud rates of 4800, 9600 or 19200.
- **Modbus ID** (1 – 247). Modbus Meter Address.
- **Meter Model** A constant identifying the product range (kWhCube=35).
- **Meter Type** Basic kWh Meter always transmitted as 1.
- **Meter Software** kWhCube software version (e.g. 0x0016 = Version 1.06).

## 4.5 RTU Commands

### 4.5.1 Function 04 (or 03) Read Multiple Registers

**Description**

This function allows a number of registers from a meter table to be read in a single operation. This command is commonly used to obtain instantaneous, energy or set-up data from the meter. This command is not available as a *broadcast* command as it requires a return data packet from the meter.

**Host Request**

| | BYTE | EXAMPLE |
|---|---|---|
| METER ADDRESS | 1 | 19H |
| FUNCTION | 2 | 04H |
| TABLE NUMBER (Address High Byte) | 3 | 0BH |
| TABLE OFFSET (Address Low Byte) | 4 | 00H |
| No. OF WORDS (N) (High Byte) | 5 | 00H |
| No. OF WORDS (N) (Low Byte) | 6 | 02H |
| CHECKSUM (High Byte) | 7 | 70H |
| CHECKSUM (Low Byte) | 8 | 37H |

The example above shows a read of kW and kW Scaling Factor from the Instantaneous Data Table 11(0BH), offset 0. The meter accessed has a Modbus ID of 25 (19H).

**Meter Response**

| | BYTE | EXAMPLE |
|---|---|---|
| METER ADDRESS | 1 | 19H |
| FUNCTION | 2 | 04H |
| NUMBER OF BYTES (2N) | 3 | 04H |
| DATA REGISTER 1 (High Byte) | 4 | 38H |
| DATA REGISTER 1 (Low Byte) | 5 | 61H |
| DATA REGISTER 2 (High Byte) | 6 | 00H |
| DATA REGISTER 2 (Low Byte) | 7 | 05H |
| CHECKSUM (High Byte) | 8 | FFH |
| CHECKSUM (Low Byte) | 8 | 38H |

The example shows a reply of 4 bytes (2 Integers) as:

3-Ph kW = 14433 (3861 Hex)
Power Scale = 5 (0005 Hex)

### 4.5.2 Function 06 Preset a Single Register

**Description**

This function allows a single integer register in a meter table to be changed by the host. This command is commonly used to program meter parameters or to reset energy registers to zero. When broadcast (address=0) all meters on the network are addressed together but none reply.

**Host Request**

| | BYTE | EXAMPLE |
|---|---|---|
| METER ADDRESS | 1 | 19 H |
| FUNCTION | 2 | 06 H |
| TABLE NUMBER (Address High Byte) | 3 | 0E H |
| TABLE OFFSET (Address Low Byte) | 4 | 00 H |
| DATA VALUE (High Byte) | 5 | 00 H |
| DATA VALUE (Low Byte) | 6 | C8 H |
| CHECKSUM (High Byte) | 7 | 89 H |
| CHECKSUM (Low Byte) | 8 | 6C H |

The example above shows a value of 200 (00H C8H), corresponding to 2000A, written to the CT Primary register (Data Table 14, offset 0). The meter accessed has a Modbus ID of 25 (19H).

**Meter Response**

| | BYTE | EXAMPLE |
|---|---|---|
| METER ADDRESS | 1 | 19 H |
| FUNCTION | 2 | 06 H |
| TABLE NUMBER (Address High Byte) | 3 | 0E H |
| TABLE OFFSET (Address Low Byte) | 4 | 00 H |
| DATA VALUE (High Byte) | 5 | 00 H |
| DATA VALUE (Low Byte) | 6 | C8 H |
| CHECKSUM (High Byte) | 7 | 89 H |
| CHECKSUM (Low Byte) | 8 | 6C H |

The reply format is a copy of the command confirming its validity:

### 4.5.3 Function 08 Loop Back Diagnostic

**Description**

This function provides a simple means of testing the communication network and detecting if a particular meter is present. This command is not available as a *broadcast* command as it requires a return data packet from the meter.

**Host Request**

| | BYTE | EXAMPLE |
|---|---|---|
| METER ADDRESS | 1 | 19 H |
| FUNCTION | 2 | 08 H |
| DIAGNOSTIC CODE (High Byte) | 3 | 00 H |
| DIAGNOSTIC CODE (Low Byte) | 4 | 00 H |
| DIAGNOSTIC DATA (High Byte) | 5 | 03 H |
| DIAGNOSTIC DATA (Low Byte) | 6 | E8 H |
| CHECKSUM (High Byte) | 7 | E3 H |
| CHECKSUM (Low Byte) | 8 | 6D H |

The example above shows a command with a Loop Back Code of 0 and Diagnostic Data of 1000 (03H E8H). The meter accessed has a Modbus ID of 25 (19H).

**Meter Response**

| | BYTE | EXAMPLE |
|---|---|---|
| METER ADDRESS | 1 | 19 H |
| FUNCTION | 2 | 08 H |
| DIAGNOSTIC CODE (High Byte) | 3 | 00 H |
| DIAGNOSTIC CODE (Low Byte) | 4 | 00 H |
| DIAGNOSTIC DATA (High Byte) | 5 | 03 H |
| DIAGNOSTIC DATA (Low Byte) | 6 | E8 H |
| CHECKSUM (High Byte) | 7 | E3 H |
| CHECKSUM (Low Byte) | 8 | 6D H |

The reply format is a copy of the command confirming its validity:

### 4.5.4 Function 16 Preset Multiple Registers

**Description**

This function allows a number of registers in a meter table to be set, by the host, in a single operation. When broadcast (address=0) all meters on the network are addressed together but none reply.

**Host Request**

| | BYTE | EXAMPLE |
|---|---|---|
| METER ADDRESS | 1 | 19 H |
| FUNCTION | 2 | 10 H |
| TABLE NUMBER (Address High Byte) | 3 | 02 H |
| TABLE OFFSET (Address Low Byte) | 4 | 02 H |
| NUMBER OF DATA WORDS (N) (High Byte) | 5 | 00 H |
| NUMBER OF DATA WORDS (N) (Low Byte) | 6 | 04 H |
| NUMBER OF DATA BYTES (2N) | 7 | 08 H |
| DATA BYTE 1 | 8 | 00 H |
| DATA BYTE 2 | 9 | 00 H |
| DATA BYTE 3 | 10 | 00 H |
| DATA BYTE 4 | 11 | 00 H |
| DATA BYTE 5 | 12 | 00 H |
| DATA BYTE 6 | 13 | 00 H |
| DATA BYTE 7 | 14 | 00 H |
| DATA BYTE 8 | 15 | 00 H |
| CHECKSUM (High Byte) | 16 | 50H |
| CHECKSUM (Low Byte) | 17 | F5H |

The example above simultaneously writes 00 to all both energy registers (8 Bytes in total). The meter accessed has a Modbus ID of 25 (19H).

**Meter Response**

| | BYTE | EXAMPLE |
|---|---|---|
| METER ADDRESS | 1 | 19 H |
| FUNCTION | 2 | 10 H |
| TABLE NUMBER (Address High Byte) | 3 | 02 H |
| TABLE OFFSET (Address Low Byte) | 4 | 02 H |
| NUMBER OF DATA WORDS (High Byte) | 5 | 00 H |
| NUMBER OF DATA WORDS (Low Byte) | 6 | 04 H |
| CHECKSUM (High Byte) | 7 | 62H |
| CHECKSUM (Low Byte) | 8 | 6AH |

The reply confirms the data address and amount of data received.

### 4.5.5 Exception Responses

When a host sends a query to an individual meter on the network it expects a normal response. In fact one of four possible events may occur as a result of the query:

◊ If the Options meter receives the message with no communication errors, and can handle the query it will reply with a normal response.
◊ If the meter does not receive the message due to a communication failure, no response will be returned and the host will eventually time-out.
◊ If the meter receives the message but detects a communication error via its CRC, no response will be returned and the host will time-out.
◊ If the meter receives the query with no communication errors but cannot handle the query (out of range data or address) the response will be an ***Exception Response*** informing the host of the nature of the error.

An Exception Response differs from a normal response in its Function Code and Data Fields.

**Exception Response**

| | BYTE | EXAMPLE |
|---|---|---|
| METER ADDRESS | 1 | 19 H |
| FUNCTION | 2 | 84 H |
| EXCEPTION CODE | 3 | 02 H |
| CHECKSUM (High Byte) | 4 | 42 H |
| CHECKSUM (Low Byte) | 5 | C6H |

**EXCEPTION FUNCTION CODE**
All normal function types have a most significant bit of 0 (< 80 Hex). In an Exception Response the meter sets the MSB to 1 (adds 80H to the received Function Type). The Function can therefore be used by the host to detect an Exception Response.

**DATA FIELD**
In an Exception Response the data field is used only to return the type of error that occurred (***Exception Code***).

The MultiCube Options Unit utilises the following Exception Codes:

| Code | Meaning |
|---|---|
| 1 | Data out of range |
| 2 | Table and/or offset out of range for this function |
| 3 | Data Over Range |
| 4 | Data Under Range |
| 5 | Data Error |

## 4.6 Modbus Diagnostics LEDs

The kWhCube Option units use 3 LEDs to aid commissioning into a Modbus system. These indicators are also useful to check continued operation of the device.

The LEDs are intelligent indicators, which are illuminated under specific conditions as follows:

***VALID COMMAND*** ‘***Cmd***’ - Illuminated when a host command is received, surrounded by RTU frame breaks (ref 4.3.2) with the correct Modbus address.

***ERROR/EXCEPTION*** ‘***Err***’ - Illuminated when a Valid Command is received as above but a CRC Error (ref 4.3.2) or Exception Condition (ref 4.5.5) is detected.

***BUS ACTIVITY*** ‘***Bus***’ - Illuminated whenever changes occur on the receive input pins of the module.

# 5. Specification

| Modbus Serial Comms | |
|---|---|
| **Bus Type**<br>**RX Loading**<br>**TX Drive** | RS485 4/2 Wires + 0V. Half Duplex<br>1/4 Unit Load Per Options Unit<br>32 Unit Loads |
| **Protocol** | Modbus RTU with 16 bit CRC. (JBUS compatible) |
| **Baud Rate** | 4800, 9600 or 19200 user programmable |
| **Address** | User Programmable 1-247 |
| **Speed**<br>**Reply Time**<br>**Max Data Packet**<br>**Command Rate** | <br>100ms maximum from command end to reply start<br>Any complete Table (Energy, Instantaneous, set-up etc)<br>New command within 5ms of previous one |

| Mechanical (Retro-Fit Options Module) | |
|---|---|
| **Enclosure** | Custom Options Enclosure. Material Mablex UL94-V-0 |
| **Dimensions**<br>**Options Unit Unfitted**<br>**MultiCube + Options** | <br>W=87mm x H=59mm x L=75mm<br>W=96mm x H=96mm x L= 138mm (130mm behind panel) |
| **Weight** | Approx. 200g |
| **Terminals** | Rising Cage. 0.2 - 4.0mm$^2$ Conductors |

| General | |
|---|---|
| **Temperature**<br>**Operating**<br>**Storage** | <br>-10 deg C to +65 deg C<br>-25 deg C to +70 deg C |
| **Environment** | IP40 (Retro-Fit Options Unit) |
| **Humidity** | <75% non-condensing |

Northern Design (Electronics) Ltd,
228 Bolton Road, Bradford,
West Yorkshire, BD3 0QW, England.
Telephone: +44 (0) 1274 729533
Fax: +44 (0) 1274 721074
Email: sales@ndmeter.co.uk
Or: support@ndmeter.co.uk