Technical Bulletin (TB0405A-RevB)

KMD-5540-005 Modbus Interface

Issued 8 July 2013

Issue

Setup and configuration of the KMD-5540-005 ModBus Interface

Instructions

This version of the KMD-5540 Series CommTalk is designed to work with a KMD series controller, on a KMD Tier 2 network, and interface to a third party device.

NOTE: *ModBus requires Hardware Configuration Manager (HCM) Version 1.08 or greater to configure interface.*

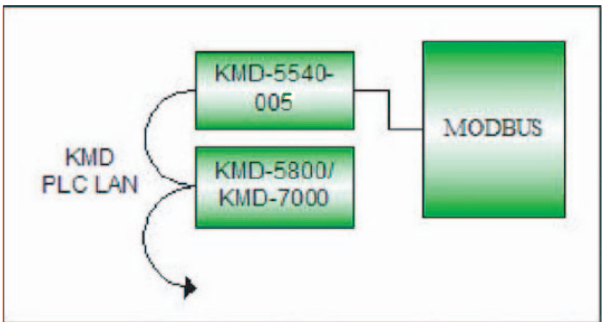


Figure 1- KMD/Modbus Network

Jumper Configuration

Configure for communication on COM1 (left side of board) and COM2 (right side of board). The KMD-5540-005 can communicate on COM2 via either RS-232 (EIA-232) or RS-485 (EIA-485) by changing the internal jumpers to the appropriate settings. Jumpers must be set for RS-232 or RS-485 communication for proper connection to the Modbus device.

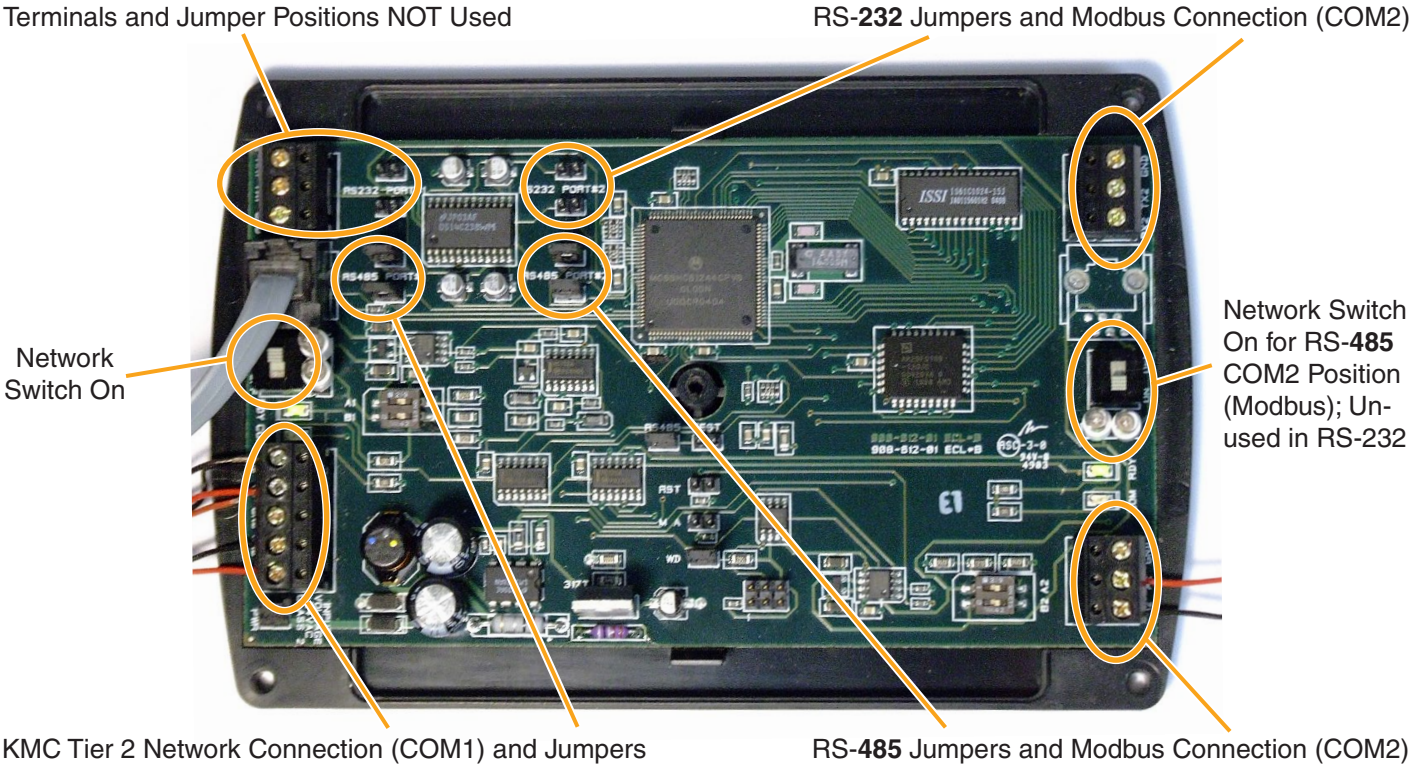


Figure 2 - Modbus Interface with Cover Removed

NOTE: *A slow, intermediate flash (approximately five second intervals) on the Com LED indicates requests by the KMD-5540 are not being answered. A rapid, constant flash indicates normal data exchange.*

Wiring

- When using RS-485 communications, **(+)** at the MODBUS device connects to the **B2** terminal and **(–)** connects to the **A2** terminal at the RS-485 termination on the KMD-5540-005.
- With RS-232 communications, **(RX)** at the MODBUS device connects to the **TX2** terminal and **(TX)** connects to the **RX2** terminal at the RS-232 termination on the KMD-5540-005.

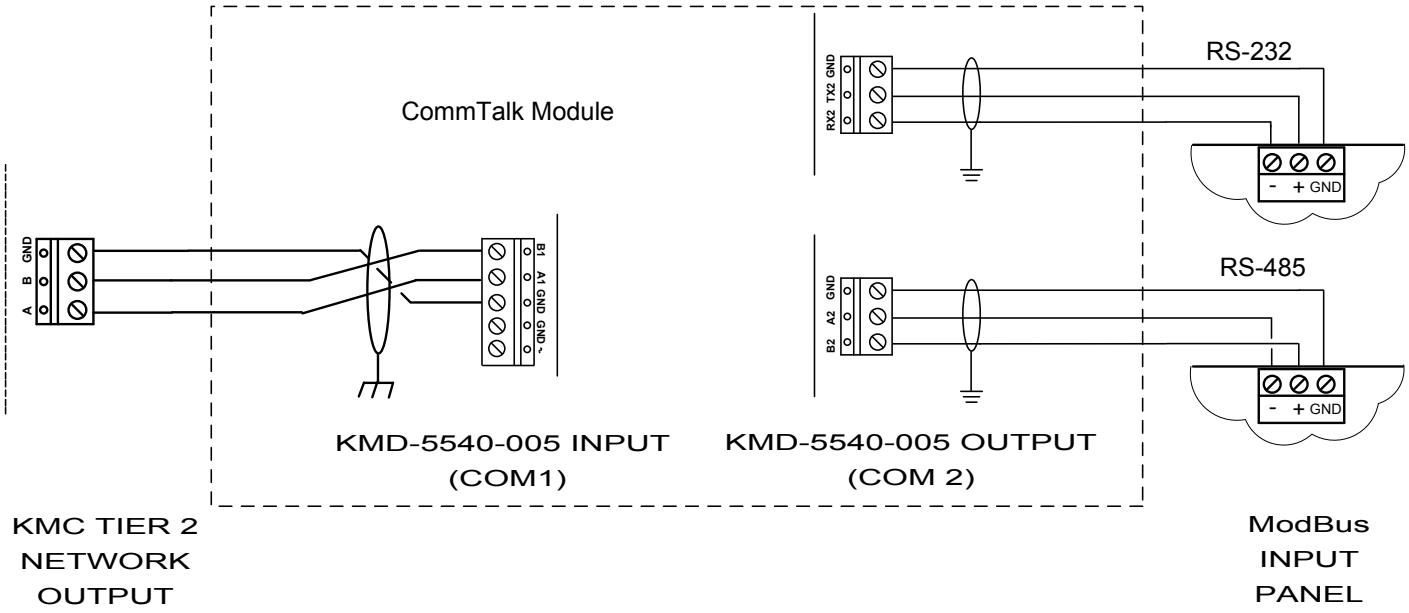


Figure 3 - Interface Wiring Diagram

Hardware Configuration Manager (HCM) Setup

NOTE: *Before you begin the next steps, you need to know the controller address, the baud rate, and the registers for your application.*

Address and Configure the Interface with HCM.

1) Direct connect to the KMD-5540-005 with HCM. Set Address of device, use last panel (if necessary), and set baud to network speed.
2) Click on the MODBUS button to set the MODBUS communication baud rate (usually 9600 or 19,200), RTU or ASCII and the data parity settings.
3) Click **OK** to send the changes.

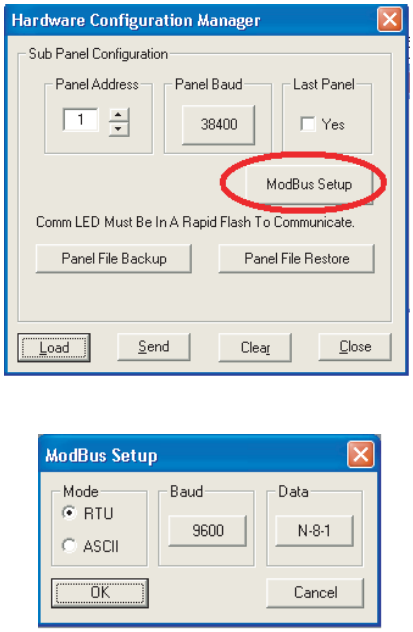


Figure 4 - HCM Setup

Programming with WinControl

Finish configuring the KMD-5540-005 MODBUS Interface by entering **up to 32 registers** in the "Description" column for each variable (see sample below).

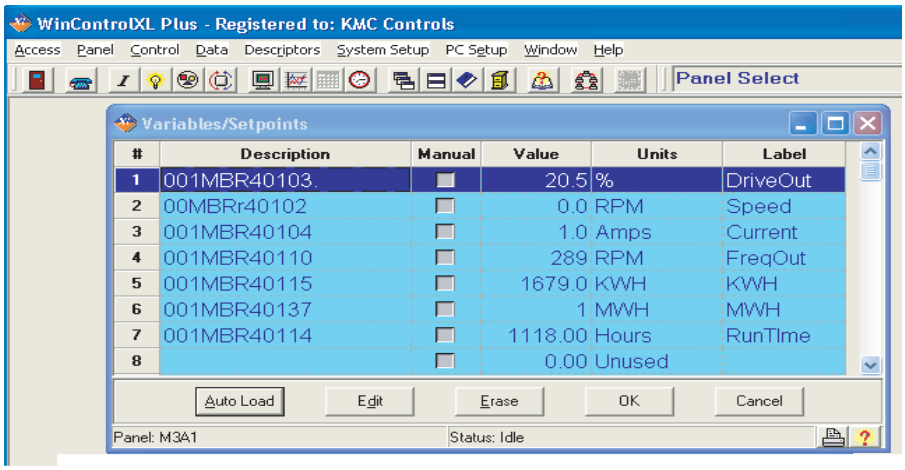


Figure 5 - WinControl Variables/Setpoints screen

Register Definitions in WinControl Variables

The Description of the Registers on the Variables/Setpoints window are defined as follows:

a. The first 3 digits (###) are the device address (decimal), where 001 is address #1.

NOTE: *Device addresses are provided by your third party vendor.*

b. 'MBR' (read) and 'MBW' (write) denote the action of the Register.

- ###**MBR**$#### Read $#### references
- ###**MBW**$#### Write $### references

c. The integer following the read/write command determines the Modbus function code.

NOTE: *This command is used to reference the correct Modbus 'function code'. The samples below are for example only. Obtain the correct integer command from the third party device manufacturer.*

- ###MBR**0**#### (function=1)
- ###MBW**0**#### (function=5)
- ###MBR**1**#### (function=2)
- ###MBR**3**####% (function=4)
- ###MBR**4**####% (function=3)
- ###MBW**4**####% (function=6 / if 2 byte signal)
  (function=16 / if 4 byte signal)

CAUTION: *Note that the various integers and function numbers do not match. (e.g., integer 3=function 4; integer 4=function 3).*

d. The last 4 digits (####) are the register address (decimal).

NOTE: Register addresses are provided by your third party vendor.

e. The final character (%) specifies the data type.

- 'L' is a 4 byte (2 registers) unsigned integer (LSW and MSB of each word is send first)
- 'F' is a 4 byte (2 registers) IEEE float (MSW and MSB of each word is sent first)
- 'S' is a 2 byte (1 register) signed integer (MSB sent first).
- If no data type is specified, then a 2 byte (1 register) unsigned integer is assumed (MSB sent first).

This following illustration is an example of the *Variables/Setpoints* window in WinControl from an actual KMD-5540-005. Please note that alphabetical portion of the Modbus registers in the Description column **MUST** be entered in uppercase. Data held in the interface module are the last known values.

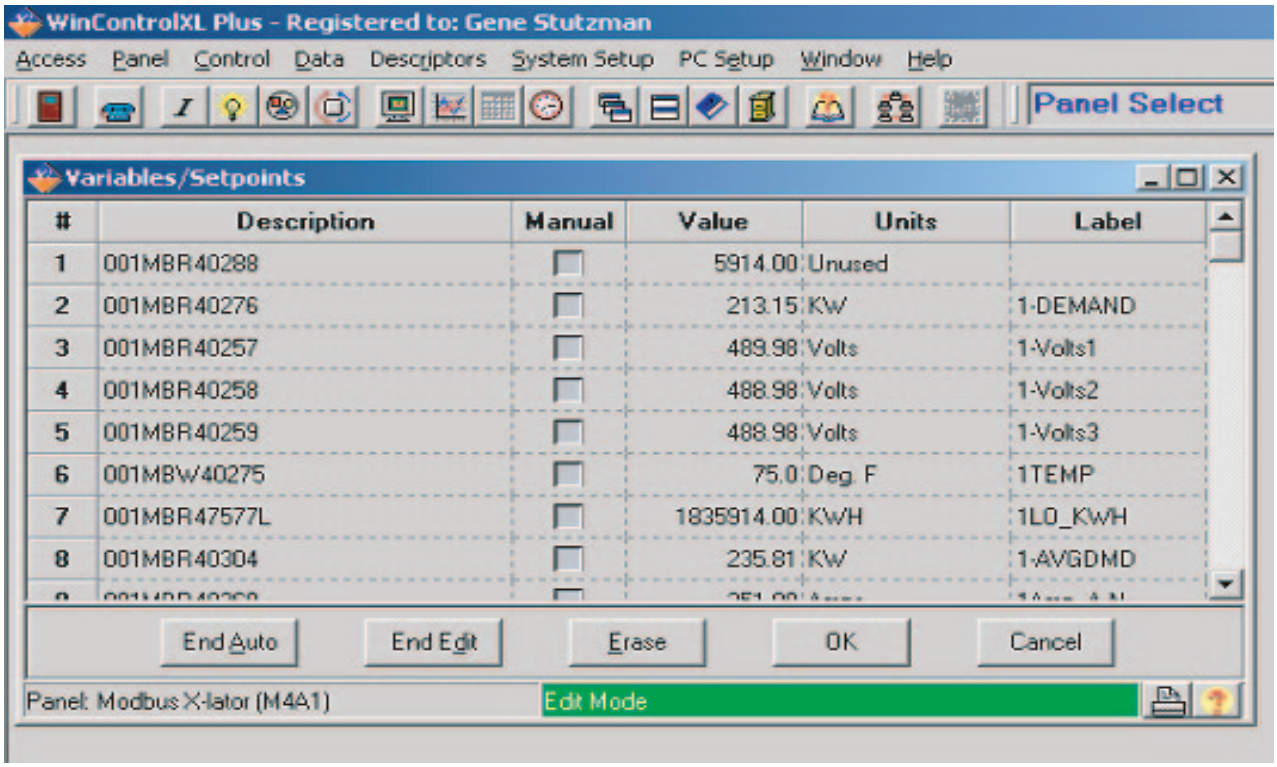


Figure 6 - Modbus Variables (Up to 32)

OPTIONAL: You may choose to 'lock' a value by clicking the check-box in the *Manual* column. When this box is marked with an '**X**', the data in the *Value* column is prevented from being changed by an automatic input.

You may also choose to label your *Variables* by clicking in the *Label* column of the *Variable* row and entering up to 8 printable characters.

After you have entered your *Variables*, enter the *Units* in which the *Values* are measured.

1) Click the *Units* column in the row of your *Variable*. The *Configure Variables* window will appear.
2) Click the Type (Digital or Analog), highlight the appropriate unit, and click **OK**.

NOTE: *The Units are determined by referring to the table of Modbus Descriptions provided by your third party manufacturer.*

NOTE: *It is advised that you enter both the Description and the Units for each Register at the same time. This prevents backtracking and saves time during configuration.*

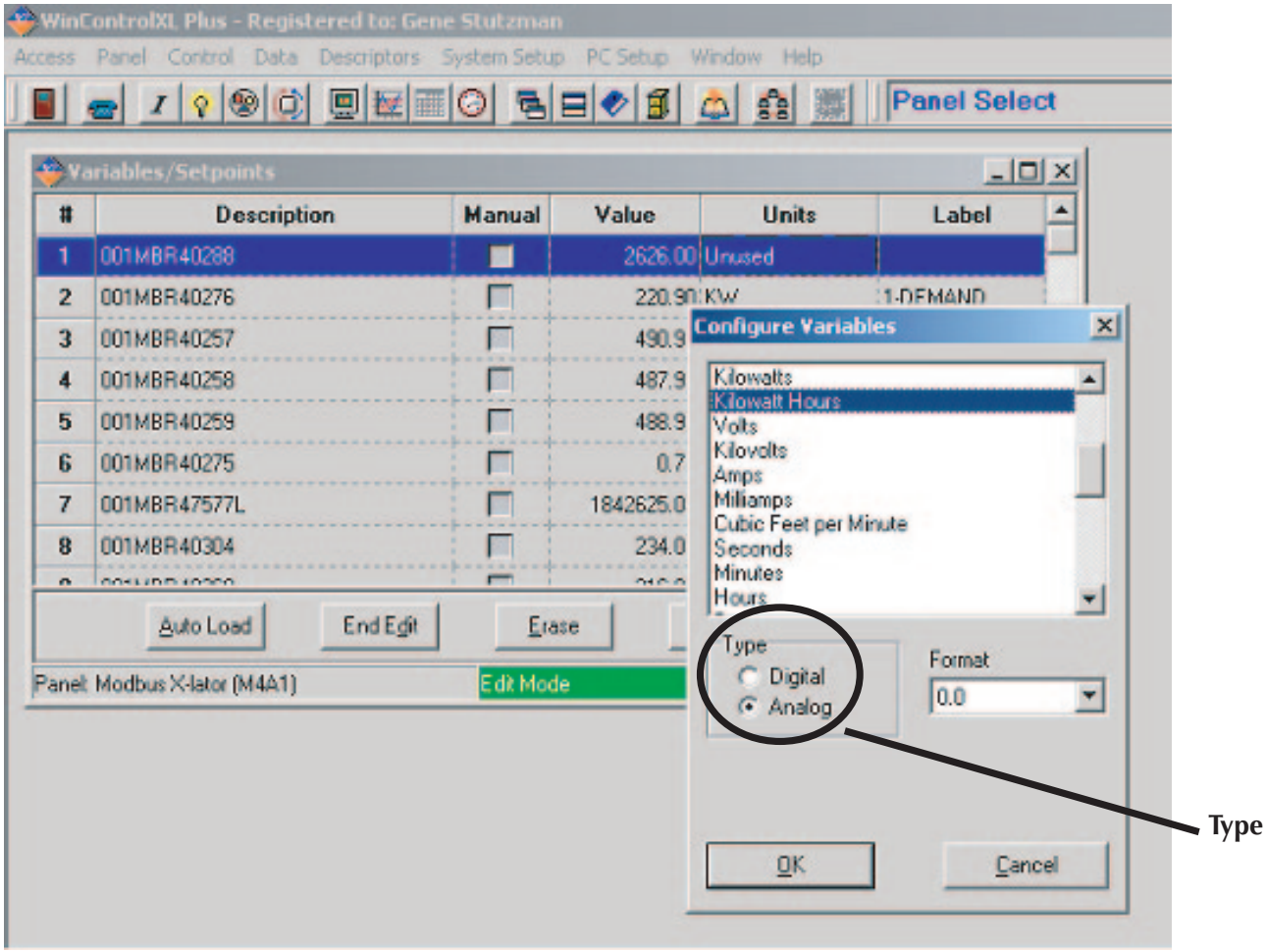


Figure 7 - Units Column / Configure Variables

NOTE: *If the value is digital but a digital unit has not been chosen, the Units will default to a digital range of 'OFF/ON'. If the value is analog but an analog unit has not been chosen, the Units will default to an analog range of "Unused".*

Converting Variable Values

The tables in WinControl are the preferred method for mapping ModBus registers to WinControl variables.

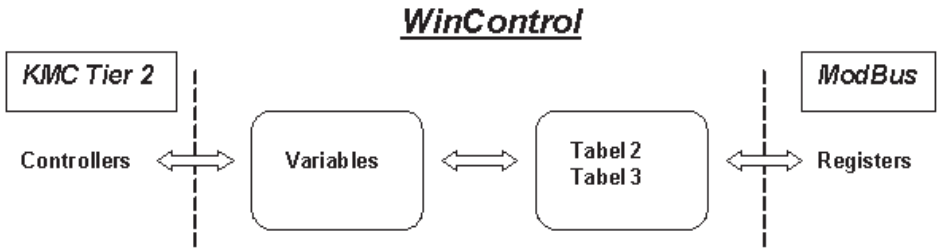
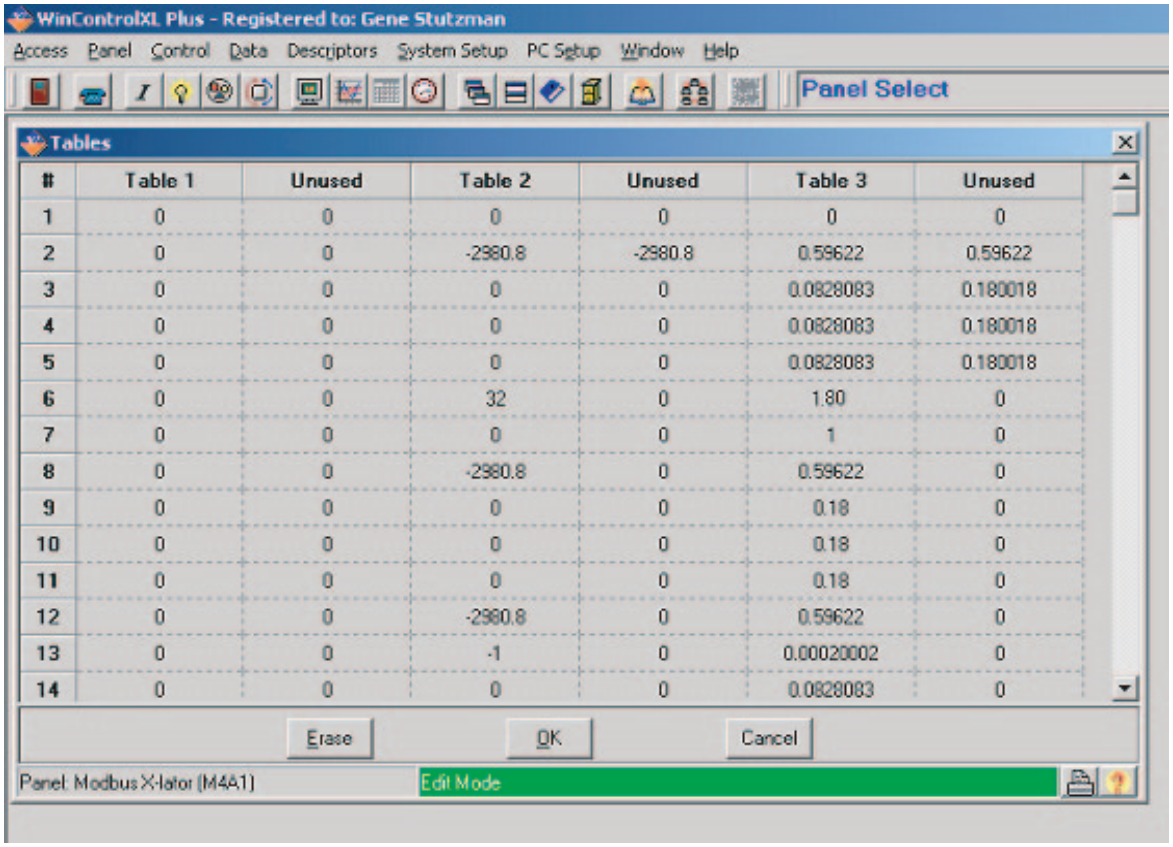


Figure 8 - KMC / Modbus Data Transfer



#	Table 1	Unused	Table 2	Unused	Table 3	Unused
1	0	0	0	0	0	0
2	0	0	-2980.8	-2980.8	0.59622	0.59622
3	0	0	0	0	0.0828083	0.180018
4	0	0	0	0	0.0828083	0.180018
5	0	0	0	0	0.0828083	0.180018
6	0	0	32	0	1.80	0
7	0	0	0	0	1	0
8	0	0	-2980.8	0	0.59622	0
9	0	0	0	0	0.18	0
10	0	0	0	0	0.18	0
11	0	0	0	0	0.18	0
12	0	0	-2980.8	0	0.59622	0
13	0	0	-1	0	0.00020002	0
14	0	0	0	0	0.0828083	0

Figure 9 - Modbus Offsets and Multipliers Table

1) The 1st column (“#”) is the Variable Number.
2) The 2nd and 3rd columns, (*“Table 1”)* and *(“Unused”)* respectively, are not used.
3) The 4th and 5th columns, (*“Table 2”*) and *(“Unused”)* respectively, are reserved for Offsets.
    a. *“Table 2”* contains the offsets for Var1 through Var16. The value in Row 15 is the offset for both Var15 and Var16.
    b. *“Unused”* contains the offsets for Var17 through Var32. The value in Row 15 is the offset for both Var31 and Var32.
4) The 6th and 7th columns, (*“Table 3”*) and *(“Unused”)* respectively, are reserved for Multipliers.
    a. The 6th column (*“Table 3”*) contains the multipliers for VAR1 through VAR16. The value in Row 15 Column 1 is the multiplier for both VAR15 and VAR16.
    b. The following column (*“Unused”*) contains the multiplier for VAR17 through VAR32. The value in Row 15 is the multiplier for both VAR31 and VAR32.

The tables can be used for the following conversions:

- **To execute a single Offset (Table 2).**
    - The Offset is subtracted from 'Write' (###MRW####) Variables.
    - The Offset is added to 'Read' (###MBR####) Variables.
- **To execute a single Multiplier (Table 3).**
    - The Multiplier divides a 'Write' (###MBW####) Variables.
    - The Multiplier multiplies a 'Read' (###MBR####) Variables.

    NOTE: For ‘Multipliers’, a value of ‘0’ in any cell is assumed to be a ‘1’.
- **To execute both an Offset and a Multiplier.**
    - For 'Read' (###MBR####) Varibles, the Multiplier is first applied to the variable, then the Offset is added, and the value is sent to the Tier 2 controller.
    - For 'Write' (###MRW####) Variables, the Offset is first subtracted from the variable, then the variable is divided by the Multiplier, and the value is sent to Modbus.

NOTE: *Rounding ‘errors’ are inherent when going from floating points to integers. Modbus stores data as integers. Unless you use floating points, decimal values are lost.*

Example of a 'Read' Variable

VAR2 (see Figure 6) has description of "001MBR40276".

1) The KMD-5540 reads a value of '5357.0 ' from the Modbus device.
2) *Table 3* - Row 2 - 6th column : '0.59622' (see Figure 9).
   [5357.0 * 0.59622 = 3193.95]
3) *Table 2* - Row 2 - 4th column): '-2980.8' (see Figure9).
   [3193.95 + (-2980.8) = 213.15]
4) A final value of '213.15' KW is placed into the value of VAR2.

Example of a 'Write' Variable

VAR6 (see Figure 6) has description of '001MBW40275'

1) VAR6 has a value of '75.0' ° F .
2) *Table 2* - Row 2 - 4th column: '32' (see Figure 9).
   [75.0 - 32 = 43.0]
3) *Table 3* - Row 2 - 6th column: '1.80' (see Figure 9).
   [43/ 1.8 = 23.8888888]
4) KMD-5540 will send '23.8888888' ° C to the Modbus device.

Example of a rounding 'error'

VAR6 (see Figure 6) has description of '001MBW40275' (Write).

1) VAR6 has a value of '75.0' ° F.
2) *Table 2* - Row 2 - 4th column: '32' (see Figure 9).
3) *Table 3* - Row 2 - 6th column: '1.80' (see Figure 9).
4) The KMD-5540 will **write** '23' to the Modbus device (see NOTE above:).
   [(75.0 - 32) / 1.80 = 23.888888]

When **reading** back the value, it will show '73.4' ° F.
[23 * 1.8] + (32) = 73.4

Additional Information

Veris™ meters have 2 byte unsigned integers (which requires *Table 3* to get the proper values) or IEEE 4 byte float values that are correct without the tables, as long as 'F' (see Register Definitions) is specified as the data type in of the Descriptor name.

NOTE: *This is the preferred method of reading data from the Veris Hawkeye meters. Use only the float registers for the upper 16-bits, as the lower 16-bit values are not needed.*

Square D Power Logic ™ meters require 'S' as part of the *Descriptor* name. This meter uses signed integers and table multiplier values are not required. It also uses the 4X registers, although this information is not actually provided in the meters.

Control Basic Conversions

Some registers may require conversion of raw data, received via the interface, that cannot be supported by the tables (e.g., a 'Read' Variable that must be Offset first and then Multiplied to display the value properly). In these cases, the *Control BASIC Editor* can be used to convert the variables to their proper value. Refer to the Modbus documents that came with the unit or contact the technical department where you purchased the device to determine the proper conversions and their sequence.

The example below illustrates how to execute this task in *Control BASIC Editor*. VAR1 displays the correct value which was converted from the raw data brought in by VAR8.

NOTE: *If more than one conversion is required, you will have to link multiple variables. Each additional link will reduce the number of total variables available for mapping.*

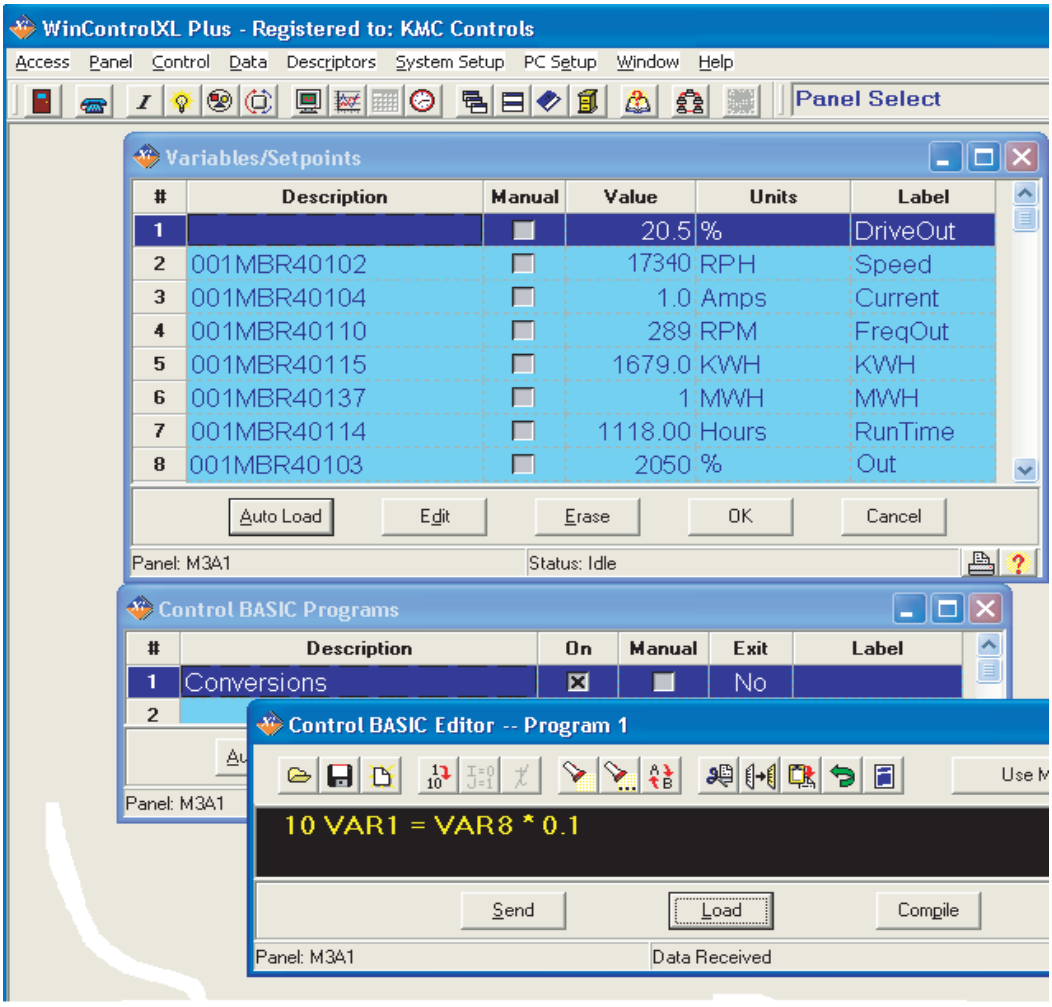


***Figure 10** - Control Basic Editor, WinControl XL connected to KMD-5540-005 and ModBus*